Citrix® NetScaler® MAS
NITRO API Getting Started Guide

Citrix® NetScaler® MAS 12.1

Copyright and Trademark Notice

Last Updated: May 2018

Contents

Introduction

The Citrix® NetScaler® MAS NITRO protocol allows you to configure and monitor the NetScaler MAS programmatically.

NITRO exposes its functionality through Representational State Transfer (REST) interfaces. This ensures that the NITRO functionality can be accessed by applications developed in any programming language. Additionally, for applications that must be developed in Java or .NET, the NITRO protocol is exposed as Java and .NET libraries that are packaged as separate Software Development Kits (SDKs).

How NITRO Works

The NITRO protocol consists of the client application and the NITRO web service, which runs on the NetScaler MAS. The communication between the client application and the NITRO web service is based on REST architecture using HTTP or HTTPS.
Invocation of a NITRO REST request initiates the following processes:

- The application sends REST request messages to the NITRO web service.
- The NITRO web service processes the requests and returns a corresponding REST response message to the client application.

To minimize traffic on the NetScaler MAS network, you retrieve the whole state of a resource from the server, make modifications to the state of the resource locally, and then upload it back to the server in one network transaction. For example, to update a mpsuser resource, you must retrieve the object, update the properties, and then upload the changed object in a single transaction.

Note: Local operations on a resource (changing its properties) do not affect its state on the server until the state of the object is explicitly uploaded.

REST Web Services

REST (REpresentational State Transfer) is an architectural style based on simple HTTP requests and responses between the client and the server. REST is used to query or change the state of objects on the server side. In REST, the server side is modeled as a set of entities where each entity is identified by a unique URL.

Each resource also has a state on which the following operations can be performed:

- **Create.** Clients can create new server-side resources on a "container" resource. You can think of container resources as folders, and child resources as files or subfolders. The calling client provides the state for the resource to be created. The state can be specified in the request by using XML or JSON format. The client can also specify the unique URL that will identify the new object. Alternatively, the server can choose and return a unique URL identifying the created object. The HTTP method used for create requests is POST.
- **Read.** Clients can retrieve the state of a resource by specifying its URL with the HTTP GET method. The response message contains the resource state, expressed in JSON format.
- **Update.** You can update the state of an existing resource by specifying the URL that identifies that object and its new state in JSON or XML, using the PUT HTTP method.
- **Delete.** You can destroy a resource that exists on the server-side by using the DELETE HTTP method and the URL identifying the resource to be removed.

The general format for NITRO URLs is as follows:

- https://<mas-ip-address>/nitro/v1/config/<resource-type>

For example, for a NetScaler MAS User, <resource-type> can be replaced by mpsuser.
In addition to the **CRUD** operations (Create, Read, Update, and Delete), resources (such as managed device) can support other operations or actions. These operations use the HTTP POST method, with the URL specifying the operation to be performed and the request body specifying the parameters for that operation.

Prerequisites

To use NITRO API, the client application needs the following:

- Access to a NetScaler MAS, release 12.1.
- A system to generate HTTP or HTTPS requests (payload in JSON format) to the NetScaler MAS. You can use any programming language or a tool to generate the requests.
- You must have a basic understanding of NetScaler MAS before using the NITRO API.
- For Java clients, you must have a system where Java Development Kit (JDK) 1.5 or later is available. The JDK be downloaded from: http://www.oracle.com/technetwork/java/javase/downloads/index.htm.
- For .NET clients, you must have a system with .NET framework 3.5 or later installed. The .NET framework can be downloaded from: http://www.microsoft.com/downloads/en/default.aspx.
- For Python clients, you must have a system with Python 2 with version 2.7.13 or above version and the Requests library (available in <NITRO_SDK_HOME>/lib) installed. The Python package can be downloaded from: https://www.python.org/downloads/.

Obtaining the Latest NITRO Package

The latest NITRO package is available as a tar file on the **Downloads** page of the NetScaler MAS GUI. You must download and un-tar the file to a folder on your local system. This folder is referred to as <NITRO_SDK_HOME> in this documentation.

The folder contains the NITRO libraries (JARs for Java and DLLs for .NET) in the lib subfolder. The libraries must be added to the client application's class path to access NITRO functionality. The <NITRO_SDK_HOME> folder also provides samples and documentation that can help you understand the NITRO SDK.

Note: The REST package contains only documentation for using the REST interfaces.

Basic NetScaler MAS Operations

This section describes the basic operations that can be performed on NetScaler MAS. The following table describes the basic NetScaler MAS operations.

Connect to NetScaler MAS

Before you perform any operation, you must authenticate and establish a session with the NetScaler MAS .

Using REST APIs through HTTP

Description	Sample

To connect to the NetScaler MAS, specify the username and password in the login object. The **Nitro Auth Token** that is created must be specified in the cookie field of all requests in the session.
 	
To ensure secure communication, use the HTTPS protocol in NITRO requests. You must have a user account on that NetScaler MAS. The configurations you can perform are limited by the administrative role assigned to your account.
 	
Note: By default, the connection with the NetScaler MAS expires after 15 minutes of inactivity. To change the session timeout period, specify the new timeout period in the login object.
	To connect to a NetScaler MAS with IP address 10.102.31.16 by using the HTTP protocol:
	• **URL.** `http://10.102.31.16/nitro/v1/config/login`
	• **HTTPS Method.** `POST`
	• **Request Headers:**
	`X-NITRO-USER: nsroot`
	`X-NITRO-PASS: verysecret`
	`Content-Type: application/json`
	• **Request Payload.**
	`{ "login": { "username":"nsroot", "password":"verysecret" } }`
	• **Response :**
	`HTTPS Status Code on success - 200 OK`
	`HTTPS Status Code on Failure -`

Description	Sample

	4xx <string> (for general HTTPS errors)
	or
	5xx <string> (for NetScaler-MAS-specific errors). The response payload provides details of the error.
	• **Response Header**
	Set-Cookie: NITRO_AUTH_TOKEN=##88374218…
	• **Response Payload.**
	{
	"errorcode": 0,
	"message":"Done",
	"sessionid":"##78C060..."
	}

Connect to the NetScaler MA Service

Before you perform any operation, you must authenticate and establish a session with the MAS service.

Using REST APIs through HTTP

Description	Sample
To connect to the NetScaler MA Service, specify the ID and Secret in the login object. The **Nitro Auth Token** that is created must be specified in the cookie field of all requests in the session. To ensure secure communication, use the HTTPS protocol in NITRO requests. You must have a user account on that NetScaler MAS. The configurations you	To connect to a NetScaler MAS with IP address 10.102.31.16 by using the HTTP protocol: • **URL.** http://10.102.31.16/nitro/v1/config/login • **HTTPS Method.** POST • **Request Headers:** X-NITRO-USER: nsroot X-NITRO-PASS: verysecret Content-Type: application/json • **Request Payload.**

can perform are limited by the administrative role assigned to your account.
 	
Note: By default, the connection with the NetScaler MAS expires after 15 minutes of inactivity. To change the session timeout period, specify the new timeout period in the login object.
	<pre>{
	"login":
	{
	"ID":"nsroot",
	"Secret":"verysecret"
	}
	}</pre>
	• **Response:**
	HTTPS Status Code on success - 200 OK
	HTTPS Status Code on Failure - 4xx <string> (for general HTTPS errors)
	or
	5xx <string> (for NetScaler-MAS-specific errors). The response payload provides details of the error.
	• **Response Header**
	Set-Cookie: NITRO_AUTH_TOKEN=##88374218…
	
	• **Response Payload.**
	
	<pre>{
	"errorcode": 0,
	"message":"Done",
	"sessionid":"##78C060..."
	}</pre>

Using REST APIs through SDKs

The first step towards using NITRO through SDKs is to establish a session with NetScaler MAS and then authenticate the session by using the NetScaler MAS user's credentials.

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

Note:

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

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

Java - Sample code to establish session

```
//Specify the NetScaler MAS IP address and protocol

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

//Specify the login credentials
client.set_credentials("admin","verysecret",360);

#uncomment the following line for NetScaler MA Service
#client.set_isCloud(true);

#In case of NetScaler MA Service, the credentials will be
treated as ID and Secret.

Client.login();
```

.NET - Sample code to establish session

```
//Specify the NetScaler MAS IP address and protocol

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

//Specify the login credentials

client.set_credentials("admin","verysecret",360);

#client.isCloud(true);  #uncomment this line for NetScaler MA
Service
#In case of NetScaler MA Service, the credentials will be
treated as ID and Secret.
```

```
Client.login();
```

Python - Sample code to establish session

```
#Specify the NetScaler MAS IP address and protocol
client = nitro_service("10.102.29.60","https")

#Specify the login credentials

client.set_credentials("admin","verysecret",360)

#client.isCloud(true)  #uncomment this line for NetScaler MA
Service

#In case of NetScaler MA Service, the credentials will be
treated as ID and Secret.

Client.login()
```

Disconnect from NetScaler MAS

When your application does not need to interact with the NetScaler MAS, logging off from the NetScaler MAS is a good practice.

Using REST APIs through HTTP

Description	Sample
To disconnect from the NetScaler MAS, use the POST HTTPS method with delete action.	To disconnect from a NetScaler MAS with IP address 10.102.31.16: • **URL.** `https://10.102.31.16/nitro/v1/config/logout` • **HTTPS Method.** `POST` • **Request Headers:** `Cookie. NITRO_AUTH_TOKEN=##78C060...` `Content-Type: application/json`

Using REST APIs through SDKs

To logout from the NetScaler MAS, invoke the **logout** method of the **nitro_service** instance. You must use the same **nitro_service** instance that was used to login for logging out.

```
Java - Sample code to logout

Client.logout();

.NET - Sample code to logout

Client.logout();

Python - Sample code to logout

Client.logout()
```

Modify Session Timeout

You can modify the timeout period by specifying a new timeout period (in seconds) in the login object. For example, to modify the timeout period to 60 minutes:

```
{

"login":
     {
     "username":"<username>",
     "password":"<password>",
     "timeout":"3600"
     }

}
```

Some points to note with regards to session timeout for NetScaler MAS <VERSION> and later versions:

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

NetScaler MAS Resource Operations

A NetScaler MAS can support multiple resources. The NITRO protocol can be used to configure these resources.

- **Using REST.** Each NetScaler MAS resource has a unique URL associated with it, depending on the type of operation to be performed. URLs for configuration operations have the format:

  https://<MAS-IP>/nitro/v1/config/ <resource_type>.

  For example, to configure a NetScaler MAS resource, the URL is https://10.102.31.16/nitro/v1/config/mpsuser.

- **Using Java, .NET, and Python SDKs.** The APIs to configure a resource are grouped into packages or namespaces that have the format com.citrix.mas.nitro.resource.config.<resource_type>.

  Each of these packages contain a class named <resource_type> that provides the APIs to configure the resource. For example, the NetScaler MAS resource mpsuser is under the package mps com.citrix.mas.nitro.resource.config.mps package or namespace.

Create Resource

Using REST APIs through HTTP

Description	Sample
To create a new resource (for example, a managed_device instance) on the NetScaler MAS, specify the resource name and other related arguments in the specific resource object	To create an instance of managed_device resource where managed_device is used to manage: • **URL.** `https://10.102.31.16/nitro/v1/config/managed_deivce?action=add_device` • **HTTPS Method.** `POST` • **Request Headers:** `Cookie. NITRO_AUTH_TOKEN=##78C060... Content-Type: application/json` • **Request Payload.**

	{
	"managed_device": w
	{
	“ip_address”:”10.106.101.17”
	"profile_name":"ns_nsroot_profile
	}
	}

Using REST APIs through SDKs

To add a managed device, first create an instance of managed_device class and set the IP address for the device being added. This must be set as it is a mandatory property. You can also set other optional properties by using the corresponding setter methods. Then invoke the static add_device method on the managed_device class by passing this instance.

Java – Sample code to add a managed_device

```
managed_device managed_device_obj = new managed_device();
managed_device_obj.set_ip_address ("10.106.101.17");
managed_device_obj.set_profile_name (
"ns_nsroot_profile");
managed_device.add_device(mas_client,
managed_device_obj);
```

C# – Sample code to add a managed_device

```
managed_device managed_device_obj = new managed_device();
managed_device_obj.ip_address = "10.106.101.17";
managed_device_obj.profile_name = "ns_nsroot_profile";
managed_device.add_device(mas_client,
managed_device_obj);
```

Python – Sample code to add a managed_device

```
managed_device_obj = managed_device()
managed_device_obj.ip_address = "10.106.101.17"
managed_device_obj.profile_name = "ns_nsroot_profile"
managed_device.add_device(mas_client, managed_device_obj)
```

Retrieve Details of NetScaler MAS Resources

Using REST APIs through HTTP

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

Note: A sample format of the request and response is provided below the table.

Retrieving all details of all resources of a specific type	In the URL, specify the type of resource for which you want to retrieve the details. For example, to retrieve all details of mpsuser resources available on a NetScaler MAS: https://<mas-ip-address>/nitro/v1/config/mpsuser
Retrieving all details of a specific resource	In the URL, specify the name of resource for which you want to retrieve the details. For example, to retrieve all details of a mpsuser resource whose id is <id_value>: https://<mas-ip-address>/nitro/v1/config/mpsuser /<id_value>
Retrieving all details of resources that have Multiple unique identifiers	In the URL, specify the type of resource and use the "args" query parameter to specify the unique attributes and the values for those attributes. For example, to get the mpsuser resources that have same tenant name and belonging to the same group: https://<mas-ip-address>/nitro/v1/config/mpsuser?args=tenant_name:Owner,groups:owner
Retrieving specific details of all resources of	In the URL, specify the type of the resource and use the "attrs" query parameter to specify the resource details that you want to retrieve. For example, to retrieve the "name" and "id" of all mpsuser resources:

a specific type	https://<mas-ip-address>/nitro/v1/config/mpsuser?attrs=name,id
Retrieving specific details of a specific resource	In the URL, specify the type and name of the resource and use the "attrs" query parameter to specify the resource details that you want to retrieve. For example, to retrieve the "name" and "id" of a mpsuser resource whose id is <id_value>: https://<mas-ip-address>/nitro/v1/config/mpsuser/<id_value>?attrs=name,id
Filtering the retrieved resources	In the URL, specify the type of resource and use the "filter" query parameter to specify the attribute(s) and the value(s) of the attributes. The resources fetched will be filtered based on the filter criteria. **Note**: The filter query parameter supports the use of PCRE regular expressions. For example, to filter the mpuser resources based on the group named “owner”: https://<mas-ip-address>/nitro/v1/config/mpsuer?filter=groups:owner
Retrieving resources in paginated manner	If the request is likely to result in a large number of resources, you can divide the results into pages and retrieve them page by page (paginated). For example, if you are retrieving, say 40, managed_device resources of a NetScaler MAS, instead of retrieving all 40 in one response, you can configure the results to be divided into 4 pages each having 10 results. In the URL, specify the name of the resource and use the following query parameters: • "pageno" - The page number to be displayed. • "pagesize" - The number of resources to be displayed in each page. For example, to retrieve the manage_device in a paginated form,

	first get a count (using the "count" query parameter shown in below row) of the managed devices. Then, accordingly specify the number of results for each page and then specify the page number to be displayed. https://<mas-ip-address>/nitro/v1/config/managed_device?pagesize=10&pageno=3

The following are some examples of retrieving NetScaler MAS resource operations:

Retrieve List of All Managed Devices from NetScaler MAS

Using REST APIs through SDKs

To get the list of all managed devices from MAS, just invoke the static get_filtered method by passing an instance of nitro_service and empty filter value.

Java – Sample code to retrieve list of all managed_devices

```
managed_device[] simplelist =
managed_device.get_filtered(mas_client, "");
```

C# – Sample code to retrieve list of all managed_devices

```
managed_device[] simplelist =
managed_device.get_filtered(mas_client, "");
```

Python – Sample code to retrieve list of all managed_devices

```
simplelist = managed_device.get_filtered(mas_client, "")
```

Retrieve Specific Details of All Managed Device Resources of a NetScaler MAS

Using REST APIs through HTTP

Description	Sample

To retrieve the specific details of all resource of one type, specify the resource along with ?attrs=prop1,prop2 in the URL.	To get IP address and ID of all managed_device resources:
	• **URL.** https://10.102.31.16/nitro/v1/config/managed_deivce?attrs=ip_address,id
	• **HTTPS Method.** GET
	• **Request Headers:** Cookie. NITRO_AUTH_TOKEN=##78C060... Content-Type: application/json
	• **Response Payload.**
	{
	"errorcode": 0,
	"message": "Done",
	"operation": "get",
	"resourceType": "managed_device",
	"username": "nsroot",
	"tenant_name": "Owner",
	"tenant_id": "64b60021-c549-41d3-814a-8b23ec8a7cf5",
	"resourceName": "",
	"managed_device": [
	{
	"id": "03ba3316-05f9-4dc7-93e5-41d7084a83d2",
	"ip_address": "192.168.117.38"
	},
	{
	"id": "05d59af7-83a9-4e2a-b350-fd3d3597fb21",
	"ip_address": "10.106.101.112-t2"
	},
	…
	]
	}

Get the Count of NetScaler MAS Resources

Using REST APIs through HTTP

If you want to have an idea of the number of resources that are likely to be returned by a request, you can use the count query string parameter to ask for a count of the resources to be returned, rather than the resources themselves.

https://<MAS_IP>/nitro/v1/ config/<resource_type>? count=yes

Description	Sample

To get a count of a specific resource type, in the URL specify the count query parameter as "yes".	To get a count of all managed devices:
	• **URL.** `https://10.102.31.16/nitro/v1/config/managed_deivce?count=yes`
	• **HTTPS Method.** `GET`
	• **Request Headers:**
	`Cookie.`
	`NITRO_AUTH_TOKEN=##78C060...`
	`Content-Type: application/json`
	• **Response Payload.**
	`{`
	`"managed_device":`
	`[`
	`{`
	`"__count": 4`
	`}`
	`]`
	`}`

Retrieve the Details of a Specific Resource

Using REST APIs through HTTP

Description	Sample
To retrieve the details of a specific resource, specify the resource along with ID that identifies the particular resource in the URL	To get details of managed device whose ID is 03ba3316-05f9-4dc7-93e5-41d7084a83: • **URL.** `https://10.102.31.16/nitro/v1/config/managed_deivce/03ba3316-05f9-4dc7-93e5-41d7084a83`

Description	Sample

	• **HTTPS Method.** GET
	• **Request Headers:**
	Cookie.
	NITRO_AUTH_TOKEN=##78C060...
	Content-Type: application/json
	• **Response Payload.**
	{
	"errorcode": 0,
	"message": "Done",
	"operation": "get",
	"resourceType":
	"managed_device",
	"username": "nsroot",
	"tenant_name": "Owner",
	"tenant_id": "64b60021-
	c549-41d3-814a-8b23ec8a7cf5",
	"resourceName": "03ba3316-
	05f9-4dc7-93e5-41d7084a83d2",
	"managed_device": [
	{
	"manufacturedate":
	"",
	"is_grace":
	"false",
	"hostname":
	"b02032rz-rlb-30",
	"std_bw_config":
	"0",
	. …
	}
	]
	}

Using REST APIs through SDKs

To get the specific details of managed device whose id is 03ba3316-05f9-4dc7-93e5-41d7084a83 from MAS, just invoke the static get_filtered method by passing an instance of nitro_service with which session was established and filter value containing this id.

Java – Sample code to retrieve list of all managed_devices

```
String json_filter = "id:03ba3316-05f9-4dc7-93e5-41d7084a83"
```

```
managed_device[] simplelist =
managed_device.get_filtered(mas_client, json_filter);
```

C# – Sample code to retrieve list of all managed_devices

```
string json_filter = "id:03ba3316-05f9-4dc7-93e5-41d7084a83"
managed_device[] simplelist =
managed_device.get_filtered(mas_client, json_filter);
```

Python – Sample code to retrieve list of all managed_devices

```
json_filter = "id:03ba3316-05f9-4dc7-93e5-41d7084a83"
simplelist =
managed_device.get_filtered(mas_client,json_filter)
```

Filter Results

Using REST APIs through SDKs

You can also retrieve resources by specifying a filter on the value of their properties by using the *com.citrix.mas. nitro.util.filtervalue* class or calling get_filtered() method defined for the resource.
For example, you can retrieve all mpsuser resources whose tenant_name or tenant_id is same and belonging to a same user groups

Java - Sample code to get filtered results

```
filtervalue[] filter = new filtervalue[2]; filter[0] = new
filtervalue("tenant_name","Owner");
filter[1] = new filtervalue("groups","owner");
mpsuser [] result =mpsuser.get_filtered(client,filter);
```

.NET - Sample code to get filtered results

```
filtervalue[] filter = new filtervalue[2]; filter[0] = new
filtervalue("tenant_name","Owner");
filter[1] = new filtervalue("groups","owner");
mpsuser [] result =mpsuser.get_filtered(client,filter);
```

Python - Sample code to get filtered results

```
filter_params = []
```

```
filter_params = [ filtervalue() for _ in range(2)]
filter_params[0] = filtervalue("groups","owner")
filter_params[1] =
filtervalue("tenant_name","Owner")
result =mpsuser.get_filtered(client, filter_params)
```

Update Resource

Using REST APIs through HTTP

Description	Sample
To update an existing NetScaler MAS resource, use the PUT HTTPS method. In the HTTPS request payload, specify the name and the other arguments that have to be changed.	To change the name of managed device instance with ID 03ba3316-05f9-4dc7-93e5-41d7084a83d2 to dev2: • **URL.** `https://10.102.31.16/nitro/v1/config/managed_deivce` • **HTTPS Method.** `PUT` • **Request Headers:** `Cookie. NITRO_AUTH_TOKEN=##78C060... Content-Type: application/json` • **Request Payload.** `{ "managed_device": { "name":"dev2", "id":" 03ba3316-05f9-4dc7-93e5-41d7084a83d2 " } }`

Note: Some properties in some NetScaler MAS resources cannot be modified after creation. The username of a NetScaler instance is an example. If the request payload of the upload operations includes such properties, NITRO does not return an error. The values provided for these properties are ignored.

Delete Resource

Using REST APIs through HTTP

Description	Sample
To delete an existing resource, specify the name of the resource to be deleted in the URL.	To delete a managed_device instance with ID 03ba3316-05f9-4dc7-93e5-41d7084a83d2: • **URL.** https://10.102.31.16/nitro/v1/config/managed_deivce/03ba3316-05f9-4dc7-93e5-41d7084a83d2 • **HTTPS Method.** DELETE • **Request Headers:** Cookie. NITRO_AUTH_TOKEN=##78C060... Content-Type: application/json

Using REST APIs through SDKs

To delete a managed device from MAS, create an instance of **managed_device** class and set the device id of the device you want to delete and invoke the static delete method by passing this instance. In the following sample code, **device_id** is the id of the device you want to delete. To know about the specific device id, get the details of the device.

Java – Sample code to delete a managed_device

```
Managed_device device_obj = managed_device();
device_obj.set_id (Object.toString(device_id));
managed_device.delete(mas_client,device_obj);
```

C# – Sample code to delete a managed_device

```
Managed_device device_obj = managed_device();
device_obj.id = Convert.ToString(device_id);
managed_device.delete(mas_client,device_obj);
```

Python – Sample code to delete a managed_device

```
device_obj = managed_device()
```

```
device_obj.id = str(device_id)
managed_device.delete(mas_client,device_obj)
```

Delete a User Session

Using REST APIs through HTTP

Description	Sample
To delete a user session identified by session id, an instance of mpssession has to be deleted by specifying the session id in the URL.	To delete a mpssession instance with ID 85326d1c-53bc-4836-a265-31489408cdec: • **URL.** https://10.102.31.16/nitro/v1/config/ *mpssession/* 85326d1c-53bc-4836-a265-31489408cdec • **HTTPS Method.** DELETE • **Request Headers:** Cookie. NITRO_AUTH_TOKEN=##78C060... Content-Type: application/json

Bulk Operations

You can query or change the configuration of resources simultaneously by, for example adding multiple mpsusers in the same operation. This minimizes the network traffic. NITRO supports the following behaviour in bulk operations:

- **Exit.** When the first error is encountered, the execution stops. The commands that were executed before the error are committed.
- **Continue.** All the commands in the list are executed even if some commands fail.

Description	Sample
To perform a bulk operation, specify the required parameters in the same request payload. You can specify the behaviour of the bulk operation in the request header using the X-NITRO-ONERROR parameter **Note:** You can also add resources of different types in one request.	To add two NetScaler MAS resources in one operation: • **URL.** https://10.102.31.16/nitro/v1/config/mpsuser • **HTTPS Method.** POST • **Request Headers:** Cookie. NITRO_AUTH_TOKEN=##78C060... Content-Type: application/json • **Response Payload.** { "mpsuser": [{ "name":"new_name", "password":"newsceret" }, { "name":"new_name1", "password":"newsecret1" }] } To add multiple resources (two NetScalers and two mpsusers) in one operation: • **URL.** https://10.102.31.16/nitro/v1/config/ • **HTTPS Method.** POST • **Request Headers:** Cookie. NITRO_AUTH_TOKEN=##78C060... Content-Type: application/json • **Response Payload.** { "ns": [{

Description	Sample

	"name":"ns_instance1",
	"ip-address":"10.70.136.5",
	"netmask":"255.255.255.0",
	"gateway":"10.70.136.1"
	},
	{
	"name":"ns_instance2",
	"ip-address":"10.70.136.8",
	"netmask":"255.255.255.0",
	"gateway":"10.70.136.1"
	}
	],
	"mpsuser":
	[
	{
	"name":"admin",
	"password":"admin",
	},
	{
	"name":"admin1",
	"password":"admin1"
	}
	]
	
	}

Perform File Operations

NITRO allows you to perform file operations such as uploading files, retrieving files, retrieving file content, and deleting files of types: txt, cert, req, xml, lic, and key.

Notes:

- Use the **BASE64** value for the file encoding attribute in the request payload. This is the only valid encoding currently supported.
- The file location path must be URL encoded. For example, if the path is /nsconfig/ssl, encode the / and use the file location as %2Fnsconfig%2Fssl.
- When uploading a file, make sure that each directory of the file path has the 755 (read, write, execute) permission.

Upload a File

Description	Sample
To upload a file to the NetScaler MAS, specify a name for the file, the location where the file must be created on the NetScaler MAS, and the content of the file.	To upload SSL certificate, sslcert resource should be specified in URL and file should be sent as multi –part form data: • **URL.** `https://10.102.31.16/nitro/v1/config/sslcert` • **HTTPS Method.** `POST` • **Request Headers:** `Cookie. NITRO_AUTH_TOKEN=##78C060... Content-Type: application/json` • **Request Payload:** `Multi-part form data with File Stream` • **Response Payload.** `{ "errorcode": 0, "message": "Done }`

Retrieve Files

Description	Sample
To retrieve the files from a specific NetScaler MAS directory, specify the directory path in the URL.	To retrieve SSL certificates, ssl_cert resource and file name should be specified in URL: • **URL.** https://10.102.31.16/nitro/v1/config/*download/ssl_cert/file_name_value<String>* • **HTTPS Method.** GET • **Request Headers:** Cookie. NITRO_AUTH_TOKEN=##78C060... Content-Type: application/json • **Response Payload.** Binary Stream

Delete a File

Description	Sample
To delete a file from the NetScaler MAS, specify the filename and the directory path in the URL..	To delete SSL certificate file, ssl_cert resource and file name should be specified in URL: • **URL.** https://10.102.31.16/nitro/v1/config/*ssl_cert/file_name_value<String>* • **HTTPS Method.** DELETE • **Request Headers:** Cookie. NITRO_AUTH_TOKEN=##78C060... Content-Type: application/json • **Response Payload.** { "errorcode": 0, "message": "Done", "severity": <String_value>

Description	Sample
	}

Error Handling

The errorcode field indicates the status of the operation.

- An errorcode of 0 indicates that the operation is successful.
- A non-zero errorcode indicates an error in processing the NITRO request.

The error message field provides a brief explanation and the nature of the failure. By default, NITRO captures only error messages. You can capture warnings by specifying the warning flag while establishing a connection with the NetScaler MAS.

Description	Sample

The response payload of all operations, specifies the error code and error message.	To get the status of an operation.
	• **URL.** https://10.102.31.16/nitro/v1/config/*ping*
	• **HTTPS Method.** GET
	• **Request Headers:** Cookie. NITRO_AUTH_TOKEN=##78C060... Content-Type: application/json
	• **Response Payload.**
	{
	"errorcode":-1,
	"message":"IP address is missing"
	}

For a more detailed description of the error codes, see the API reference available in the <NITRO_SDK_HOME>/doc folder.

Error in a Single Resource Operation

The response of a single erroneous operation is as follows:

```
 HTTPS Status Code: 4xx <string> (for general HTTPS errors) or
5xx <string> (for NetScaler-MAS-specific errors)

 Response Payload
```

```
{

errorcode: <Error code>
message: "<Error message>"
severity: "ERROR"
}
```

Error in a Bulk Operation

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

Response

```
HTTPS Status Code: 207 Multi Status
```

Response Payload when X-NITRO-ONERROR is Set to Continue

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

```
{
"errorcode": 1243,

"message": "Bulk operation failed", "severity": "ERROR",
"response":
[
{
"errorcode": 273,

"message": "Resource already exists", "severity": "ERROR"
},
{

"errorcode": 0, "message": "Done", "severity": "NONE"
}
]
}
```

Response Payload when X-NITRO-ONERROR is Set to Exit

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

```
{
"errorcode": 1243,

"message": "Bulk operation failed", "severity": "ERROR",
"response":
[
         {
"errorcode": 273,

"message": "Resource already exists", "severity": "ERROR"
}
        ]
}
```

NetScaler MAS NITRO API Proxy

Basically, MAS NITRO API proxy is used to perform operations on NetScaler appliance through NetScaler MAS. The NITRO API request with specific header identifying the NetScaler appliance is sent to the MAS which acts as request proxy on identifying this header and forwards the request as it is to the identified NetScaler. On receiving the request, NetScaler processes this request as if the request was directly sent to it and sends the response back to the MAS and MAS then sends this response as it is back to the client.

Any request to NetScaler appliance through MAS API Proxy should have the appropriate headers and payload format adhering to NetScaler REST Documentation. Refer below link for NetScaler REST API Documentation:

https://developer-docs.citrix.com/projects/netscaler-nitro-api/en/latest/

Every MAS API proxy request must contain either one of the following headers:

- _MPS_API_PROXY_MANAGED_INSTANCE_IP: Name of the managed instance
- _MPS_API_PROXY_MANAGED_INSTANCE_ID: IP address of the managed instance
- _MPS_API_PROXY_MANAGED_INSTANCE_NAME: ID of the managed instance

Connect to the NetScaler Appliance through MAS API Proxy

Before you perform any operation, you must authenticate and establish a session with the appliance.

Description	Sample

To connect to the appliance, specify the username and password in the login object. Either one of the following headers must be specified for MAS API proxy:
• _MPS_API_PROXY_MANAGED_INSTANCE_IP: ns_ip_address
• _MPS_API_PROXY_MANAGED_INSTANCE_ID: ns_instance_id
 	
The Nitro Auth Token that is created must be specified in the cookie field of all requests in the session.
 	
To ensure secure communication, use the HTTPS protocol in NITRO requests.
 	
You must have a user account on that appliance. The configurations you can perform are limited by the administrative role assigned to your account.
 	
Note: By default, the connection with the appliance expires after 15 minutes of inactivity. To change the session timeout period, specify the new timeout period in the login object.
	To connect to a NetScaler whose IP address is 10.102.29.60 through MAS with IP address 10.102.31.16 by using the HTTPS protocol:
	
	URL.
	https://10.102.31.16/nitro/v1/config/login
	HTTPS Method. POST
	
	Request Headers:
	X-NITRO-USER: nsroot
	X-NITRO-PASS: verysecret
	_MPS_API_PROXY_MANAGED_INSTANCE_IP:
	10.102.29.60
	Content-Type: application/json
	
	Request Payload.
	{
	"login":
	{
	"username":"nsroot",
	
	"password":"verysecret"
	}
	}
	
	Response
	
	HTTPS Status Code on success - 200 OK
	HTTPS Status Code on Failure - 4xx <string> (for general HTTPS errors)
	or

Description	Sample

	5xx <string> (for NetScaler-MAS-specific errors). The response payload provides details of the error.
	
	Response Header.
	Set-Cookie: NITRO_AUTH_TOKEN=##88374218…
	
	Response Payload.
	{
	"errorcode": 0,
	"message":"Done",
	"sessionid":"##78C060..."
	}

Create a New Resource on NetScaler Appliance through MAS API Proxy

Description	Sample
To create a new resource (for example, lbvserver instance) on the NetScaler through MAS API Proxy, specify the resource name and other related arguments in the specific resource object	To create an instance of lbvserver resource on NetScaler whose id is 746603d3-cd7e-4972-833d-c32163b92c0a through NetScaler MAS with IP address 10.102.31.16 where lbvserver is a load balancing virtual server: • **URL.** https://10.102.31.16/nitro/v1/config/lbvserver • **HTTPS Method.** POST • **Request Headers.** Cookie: NITRO_AUTH_TOKEN=##78C060... Content-Type: application/vnd.com.citrix.NetScaler.lbvserver+json

	Accept-type: application/vnd.com.citrix.NetScaler.lbvserver+json
	
	_MPS_API_PROXY_MANAGED_INSTANCE_ID: 746603d3-cd7e-4972-833d-c32163b92c0a
	• **Request Payload.**
	{
	"lbvserver":{
	"ipv46":"31.4.5.5",
	"servicetype":"SSL",
	"name":"lb_demo",
	"appflowlog":"ENABLED",
	"downstateflush":"ENABLED",
	"port":80,
	"lbmethod":"LEASTCONNECTION",
	"clttimeout":45
	}
	}

Java, .NET, and Python API Usage

This section provides basic information for using the Java, .NET, and Python SDKs that are provided for the NITRO API. The API are categorized on their scope and purpose.

NetScaler MAS resources are organized into a set of packages or namespaces. Each package or namespace corresponds to a MAS feature. For example, all NetScaler MAS resources are available in *com.citrix.mas.nitro.resource.config.ns*. Each NetScaler MAS resource is represented by a class. For example, the class that represents a user is called **mpsuser** (in *com.citrix.mas.nitro.resource.config.mps*). The state of a resource is represented by properties of a class. You can get and set the properties of the class.

Notes:

- For the python SDK, the package path is of the form massrc.com.citrix.mas.....
- The setter and getter properties are always executed locally on the client. They do not involve any network interaction with the NITRO web service. All properties have basic simple types: integer, long, boolean, and string.
- All NITRO operations are logged in the /var/mps/log/mps_service.log file on the NetScaler MAS .
- Executable samples are available in the *<NITRO_SDK_HOME>/sample* directory.

Tutorial: Create Your First NITRO Application

After completing this tutorial, you will understand and be able to use NITRO to log in to the MAS, create resources, retrieve details of a resource, delete a resource, schedule configuration jobs, perform configuration audit, save the configurations, and log out of the MAS.

Notes:

- Before you begin, make sure that you have the latest MAS NITRO SDK and that the client application satisfies the prerequisites for using the MAS NITRO SDK.
- All NITRO exceptions are captured by the *com.citrix.mas.nitro.exception.nitro_exception* class.

Use Java API to Create your First NITRO Application

1. Copy the libraries from <NITRO_SDK_HOME>/lib folder to the project classpath.
2. Create a new class and name it MyFirstNitroApplication.

3. Create an instance of com.citrix.mas.nitro.service.nitro_service class. This instance is used to perform all operations on the NetScaler MAS:

```
nitro_service client = new
nitro_service("10.102.29.170","HTTP");
```

This code establishes a connection with NetScaler MAS that has IP address 10.102.29.170 and uses the HTTP protocol. Replace 10.102.29.170 with the IP address of the NetScaler MAS that you have access to.

Note:

- To use HTTPS connection for incorporating wire level security, use HTTPS protocol by passing https instead of http to constructor of nitro_service as shown below:

```
nitro_service client = new
nitro_service("10.102.29.170", "https");
```

- SDKs take care of SSL Certificates involved in SSL Handshake.
- MPS currently presents a test certificate not issued by any signing authority, so SDK bypasses the credentials check by using empty Trust Manager

4. Use the **nitro_service** instance to log in to the NetScaler MAS using your credentials:

```
client.set_credentials("admin","verysecret");
#uncomment this line for NetScaler MA Service
#client.set_isCloud(true);
#In case of NetScaler MA Service, the credentials will be
treated as ID and Secret.
client.login();
```

This code logs into the NetScaler MAS , with user name as admin and password as verysecret. Replace the credentials with your login credentials.

5. Create an instance of the **com.citrix.mas.nitro.resource.config.mps.mpsuser** class. You will use this instance to perform operations on the mpsuser.

```
mpsuser my_mpsuser = new mpsuser();
```

6. Use the mpsuser instance to create a new mpsuser resource on the NetScaler MAS:

```
my_mpsuser.set_name("nsroot");
my_mpsuser.set_password("mynsroot");
String[] user_groups = new String[1];
user_groups[0] = "owner";
my_mpsuser.set_groups(user_groups);
mpsuser.add(client, my_mpsuser);
```

This code first sets the attributes (name, password, user groups of the user) of the mpsuser locally and then creates the resource by using the corresponding add() method.

7. Retrieve the details of the created mpsuser:

```
mpsuser new_mpsuser = mpsuser.get(client, my_mpsuser);
System.out.println("Name : " +new_mpsuser.get_name()
```

This code first retrieves the details of the mpsuser resource as identified by my_mpsuser as an object from the NetScaler MAS, extracts the required attributes (name) from the object, and displays the results.

8. Delete the created mpsuser:

```
mpsuser.delete(client, my_mpsuser);
```

This code deletes the mpsuser resource as identified by my_mpsuser on the MAS.

9. Log out of the NetScaler MAS using `client.logout()`.

Use .NET API to Create your First NITRO Application

1. Copy the libraries from <NITRO_SDK_HOME>/lib folder to the project classpath.
2. Create a new class and name it MyFirstNitroApplication.
3. Create an instance of com.citrix.mas.nitro.service.nitro_service class. This instance is used to perform all operations on the NetScaler MAS :

```
nitro_service client = new nitro_service("10.102.29.170",
"http");
```

This code establishes a connection with NetScaler MAS that has IP address 10.102.29.170 and uses the HTTP protocol. Replace 10.102.29.170 with the IP address of the NetScaler MAS that you have access to.

Note:

- To use HTTPS connection for incorporating wire level security, use HTTPS protocol by passing https instead of http to constructor of nitro_service as shown below:

  ```
  nitro_service client = new
  nitro_service("10.102.29.170", "https");
  ```

- SDKs take care of SSL Certificates involved in SSL Handshake.
- MPS currently presents a test certificate not issued by any signing authority, so SDK bypasses the credentials check by using empty Trust Manager.

4. Use the nitro_service instance to log in to the NetScaler MAS using your credentials:

   ```
   client.set_credentials("admin","verysecret");
   #client.isCloud = true;  #uncomment this line for
   NetScaler MA Service
   #In case of NetScaler MA Service, the credentials will be
   treated as ID and Secret.
   Client.login();
   ```

   This code logs into the NetScaler MAS , with user name as admin and password as verysecret. Replace the credentials with your login credentials.

5. Create an instance of the com.citrix.mas.nitro.resource.config.mps.mpsuser class. You will use this instance to perform operations on the mpsuser:

   ```
   mpsuser my_mpsuser = new mpsuser();
   ```

6. Use the mpsuser instance to create a new mpsuser:

   ```
   my_mpsuser.name = "nsroot";
   my_mpsuser.password = "mynsroot";
   String[] user_groups = {"owner"};
   my_mpsuser.groups = user_groups;
   mpsuser.add(client, my_mpsuser);
   ```

   This code first sets the attributes (name, password, user groups of the user) of the mpsuser locally and then creates the resource by using the corresponding add() method.

7. Retrieve the details of the mpsuser you have created:

   ```
   mpsuser new_mpsuser = mpsuser.get(client, my_mpsuser);
   System.out.println("Name : " +new_mpsuser.name);
   ```

   This code first retrieves the details of the mpsuser resource as identified by my_mpsuser as an object from the NetScaler MAS, extracts the required attributes (name) from the object, and displays the results.

   Delete the mpsuser you created in the above steps:

   ```
   mpsuser.delete(client, my_mpsuser);
   ```

8. Log out of the MAS using `client.logout()`.

Use Cases

This section covers some examples and use cases. More examples will be added in future updates to this section.

Add a mpsuser Resource to NetScaler MAS

To create a new mpsuser, instantiate the mpsuser class, configure the instance by setting its properties locally, and then upload the new resource instance to the NetScaler MAS. The following sample code creates a mpsuser resource

Java - Sample code to add a NetScaler MAS resource

```
//Create an instance of the mpsuser
mpsuser my_mpsuser = new mpsuser();

my_mpsuser.set_name("nsroot");
my_mpsuser.set_password("mynsroot");
String[] user_groups = new String[1];
user_groups[0] = "owner";
my_mpsuser.set_groups(user_groups);


//Upload the resource to NetScaler MAS
mpsuser.add(client, my_mpsuser);
```

.NET - Sample code to add a NetScaler MAS resource

```
//Create an instance of the mpsuser
mpsuser my_mpsuser = new mpsuser();
my_mpsuser.name = "nsroot";
my_mpsuser.password = "mynsroot");
String[] user_groups = {"owner"};
my_mpsuser.groups = user_groups;

//Upload the resource to NetScaler MAS
mpsuser.add(client, my_mpsuser);
```

Python - Sample code to add a NetScaler MAS resource

```
#Create an instance of the mpsuser
my_mpsuser = mpsuser()
my_mpsuser.name = "nsroot"
```

```
my_mpsuser.password = "mynsroot"
user_groups = []
user_groups.append("owner")
my_mpsuser.groups = user_groups

#Upload the resource to NetScaler MAS
mpsuser.add(client, my_mpsuser)
```

Retrieve Properties of mpsuser resource

To retrieve the properties of a mpsuser resource, you retrieve the resource object from the NetScaler MAS. Once the object is retrieved, you can extract the required properties of the resource locally, without further network traffic. The following sample code retrieves the details of a mpsuser resource.

Java - Sample code to get details of resource

```
//Retrieve the id of the resource object from the NetScaler
MAS
mpsuser my_mpsuser = new mpsuser();
mpsuser[] simplelist = mpsuser.get(client);
for(mpsuser item : simplelist ){
    if (item.get_name().equals("nsroot")) {
         my_mpsuser.set_id(item.get_id());
    }
      }
      my_mpsuser.set_name("nsroot");

//Retreive the resource object identified by the id
     my_mpsuser_retreived = mpsuser.get(client,my_mpsuser);

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

System.out.println(my_mpsuser_retreived.get_name());
System.out.println(my_mpsuser_retreived.get_id());
```

.NET - Sample code to get details of resource

```
//Retrieve the id of the resource object from the NetScaler
MAS
mpsuser my_mpsuser = new mpsuser();
mpsuser[] simplelist = mpsuser.get(client);
```

```
 for(int i=0; i< simplelist.Length; i++){
          if (simplelist[i].name.Equals("nsroot")){
               my_mpsuser.id = simplelist[i].id;
 }
      }
      my_mpsuser.name = "nsroot";

 //Retreive the resource object identified by the id
      my_mpsuser_retreived = mpsuser.get(client,my_mpsuser);

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

 Console.WriteLine(my_mpsuser_retreived.name);
 Console.WriteLine(my_mpsuser_retreived.id);
```

Python - Sample code to get details of resource

```
 #Retrieve the id of the resource object from the NetScaler MAS
 my_mpsuser = mpsuser()
 simplelist = mpsuser.get(client)
 for item in simplelist :
     if item.name == "nsroot" :
          mympsuser.id = item.id;
       mympsuser.name = "nsroot"

 #Retreive the resource object identified by the id
      my_mpsuser_retreived = mpsuser.get(client,mympsuser);

 #Extract the properties of the resource from the object
locally
 print(my_mpsuser_retreived.name)
 print(my_mpsuser_retreived.id)
```

Update a mpsuser Resource

To update the properties of a mpsuer resource, instantiate the mpsuser resource class, specify the id of the resource to be updated, configure the resource by updating its properties locally, and then upload the updated resource instance to the NetScaler MAS.

Note: Some properties in some NetScaler MAS resources are not allowed to be modified after creation. The location or tenant_id of the managed_device resource, are examples of

such properties. Even though the update method appears to succeed, these properties retain their original values on the NetScaler MAS .
The following sample code updates the password of a mpsuser resource.

Java - Sample code to update a NetScaler MAS resource

```
mpuser update_mpsuser = new mpuser();

//Specify the id of the mpuser to be updated
update_mpsuser.set_id("9b304030-4233-45df-be3b-0d73d3cd59a9
");

//Specify the updated password
update_mpsuser.set_password("verysecret");

//Upload the resource to NetScaler MAS
mpuser.update(client,update_mpsuser);
```

.NET - Sample code to update a NetScaler MAS resource

```
mpuser update_mpsuser = new mpuser();

//Specify the id of the mpuser to be updated
update_mpsuser.id = "9b304030-4233-45df-be3b-0d73d3cd59a9 ";

//Specify the updated password
update_mpsuser.password = "verysecret";

//Upload the resource to NetScaler MAS
mpuser.update(client,update_mpsuser);
```

Python - Sample code to update a NetScaler MAS resource

```
update_mpsuser = mpuser()

#Specify the id of the mpuser to be updated
update_mpsuser.id = "9b304030-4233-45df-be3b-0d73d3cd59a9 "

#Specify the updated password
update_mpsuser.password = "verysecret"

#Upload the resource to NetScaler MAS
mpuser.update(client,update_mpsuser)
```

Delete a mpsuser Resource

To delete an existing resource, invoke the static method **delete()** on the resource class, by passing the id of the resource. The following sample code deletes mpsuser resource whose id is 9b304030-4233-45df-be3b-0d73d3cd59a9:

Java - Sample code to delete a NetScaler MAS resource

```
mpsuser remove_mpsuser = new mpsuser();
remove_mpsuser.set_id("9b304030-4233-45df-be3b-0d73d3cd59a9
");
mpsuser.delete(client, remove_mpsuser);
```

.NET - Sample code to delete a NetScaler MAS resource

```
mpsuser remove_mpsuser = new mpsuser();
remove_mpsuser.id = "9b304030-4233-45df-be3b-0d73d3cd59a9 ";
mpsuser.delete(client, remove_mpsuser);
```

Python - Sample code to delete a NetScaler MAS resource

```
remove_mpsuser = mpsuser()

remove_mpsuser.id = "9b304030-4233-45df-be3b-0d73d3cd59a9 "
mpsuser.delete(client, remove_mpsuser)
```

Upload a SSL Certificate to NetScaler MAS

To upload a SSL certificate, first create an instance of ns_ssl_cert class and set the file name and path to the SSL file. Then invoke the static upload method by passing the created instance.

Java – Sample code to upload a SSL certificate

```
ns_ssl_cert certificate_obj = new ns_ssl_cert();
certificate_obj.set_file_location_path("/root/random_scripts/
");
certificate_obj.set_file_name ("mydomain29282.com.pem");
ns_ssl_cert.upload(mas_client, certificate_obj);
```

C# – Sample code to upload a SSL certificate

```
ns_ssl_cert certificate_obj = new ns_ssl_cert();
certificate_obj.file_location_path = "/root/random_scripts/";
certificate_obj.file_name = "mydomain29282.com.pem";
ns_ssl_cert.upload(mas_client, certificate_obj);
```

Python – Sample code to upload a SSL certificate

```
certificate_obj = ns_ssl_cert()
certificate_obj.file_location_path = "/root/random_scripts/"
certificate_obj.file_name = "mydomain29282.com.pem"
ns_ssl_cert.upload(mas_client, certificate_obj)
```

Replace the file name and file path to appropriate values.

Download a SSL Certificate from NetScaler MAS

To download a SSL certificate, first create an instance of ns_ssl_cert class and set the file name and path to the SSL file. Then invoke the static download method by passing the created instance.

Java – Sample code to download a SSL certificate

```
ns_ssl_cert certificate_obj = new ns_ssl_cert();
certificate_obj.set_file_location_path("/root/random_scripts/
");
certificate_obj.set_file_name ("mydomain29282.com.pem");
ns_ssl_cert retrieved_certificate_obj = ns_ssl_cert.download
(mas_client, certificate_obj);
```

C# – Sample code to download a SSL certificate

```
ns_ssl_cert certificate_obj = new ns_ssl_cert();
certificate_obj.file_location_path = "/root/random_scripts/";
certificate_obj.file_name = "mydomain29282.com.pem";
ns_ssl_cert retrieved_certificate_obj =
ns_ssl_cert.download(mas_client, certificate_obj);
```

Python – Sample code to download a SSL certificate

```
certificate_obj = ns_ssl_cert()
certificate_obj.file_location_path = "/root/random_scripts/"
certificate_obj.file_name = "mydomain29282.com.pem"
retrieved_certificate_obj = ns_ssl_cert.download(mas_client,
certificate_obj)
```

Replace the file name and file path to appropriate values.

Delete a SSL Certificate from NetScaler MAS

To delete a SSL certificate, first create an instance of ns_ssl_cert class and set the file name and path to the SSL file. Then invoke the static delete method by passing the created instance.

Java – Sample code to delete a SSL certificate

```
ns_ssl_cert certificate_obj = new ns_ssl_cert();
certificate_obj.set_file_location_path("/root/random_scripts/"
);
certificate_obj.set_file_name ("mydomain29282.com.pem");
ns_ssl_cert.delete (mas_client, certificate_obj);
```

C# – Sample code to delete a SSL certificate

```
ns_ssl_cert certificate_obj = new ns_ssl_cert();
certificate_obj.file_location_path = "/root/random_scripts/";
certificate_obj.file_name = "mydomain29282.com.pem";
ns_ssl_cert.delete(mas_client, certificate_obj);
```

Python – Sample code to delete a SSL certificate

```
certificate_obj = ns_ssl_cert()
certificate_obj.file_location_path = "/root/random_scripts/"
certificate_obj.file_name = "mydomain29282.com.pem"
ns_ssl_cert.delete(mas_client, certificate_obj)
```

Retrieve Details of a SSL Certificate from NetScaler MAS

To retrieve details of a SSL certificate, invoke the static get_filtered method by passing the filename of certificate.

Java – Sample code to retrieve details of a SSL certificate

```
String json_filter = "file_name: mydomain29282.com.pem";
retrieved_cert = ns_ssl_cert.get_filtered(mas_client,
json_filter);
```

C# – Sample code to retrieve details of a SSL certificate

```
String json_filter = "file_name: mydomain29282.com.pem";
retrieved_cert = ns_ssl_cert.get_filtered(mas_client,
json_filter);
```

Python – Sample code to retrieve details of a SSL certificate

```
json_filter = "file_name: mydomain29282.com.pem"
retrieved_cert = ns_ssl_cert.get_filtered(mas_client,
json_filter)
```

Retrieve List of All SSL Certificates from NetScaler MAS

To retrieve list of all SSL certificates, invoke the static get_filtered method by passing empty filter value.

Java – Sample code to retrieve list of all SSL certificates

```
retrieved_cert = ns_ssl_cert.get_filtered(mas_client, "");
```

C# – Sample code to retrieve list of all SSL certificates

```
retrieved_cert = ns_ssl_cert.get_filtered(mas_client, "");
```

Python – Sample code to retrieve list of all SSL certificates

```
retrieved_cert = ns_ssl_cert.get_filtered(mas_client, "")
```

Perform Inventory Check from NetScaler MAS

Performing inventory check involves rediscovering that device that was added to NetScaler MAS. To do this check, create an instance of **inventory** class and set device IP address and the name of the partition in the device. Then invoke static **get** method by passing this instance. In the following sample code, activity ID of the rediscovery process is fetched.

Java – Sample code to retrieve activity id of rediscovery process

```
inventory inventory_obj = new inventory();
```

```
 String device_ip = device_ip + "-" + partition;
 inventory_obj.set_device_ipaddress(device_ip);
 inventory[] simplelist =
inventory.get(mas_client,inventory_obj);
 String act_id =  simplelist[simplelist.length-1].act_id;
```

C# – Sample code to retrieve activity id of rediscovery process

```
 inventory inventory_obj = new inventory();
 String device_ip = device_ip + "-" + partition;
 inventory_obj.device_ipaddress = device_ip;
 inventory[] simplelist =
 inventory.get(mas_client,inventory_obj);
 String act_id =  simplelist.Last().act_id;
```

Python – Sample code to retrieve activity id of rediscovery process

```
 inventory_obj = inventory()
 device_ip = device_ip + "-" + partition
 inventory_obj.device_ipaddress = device_ip
 simplelist = inventory.get(mas_client, inventory_obj)
 act_id =  simplelist[-1].act_id
```

Retrieve List of all Events from NetScaler MAS

To retrieve list of all events, invoke **get** method of the event class. To get the count of all events, create an instance of **event** class and set the count property to **yes** and pass it as argument to the static get method.

Java – Sample code to retrieve list of all events and its count

```
event[]  simplelist = event.get(mas_client);
event event_obj =new event();
event_obj.set_count(yes);
int event_count = event.get(mas_client,event_obj);
```

C# – Sample code to retrieve list of all events and its count

```
event[]  simplelist = event.get(mas_client);
event event_obj =new event();
event_obj.count = yes;
int event_count = event.get(mas_client,event_obj);
```

Python – Sample code to retrieve list of all events and its count

```
simplelist = event.get(mas_client)
event_obj = event()
event_obj.count = yes
event_count = event.get(mas_client,event_obj)
```

Delete an Event from NetScaler MAS

To delete an event from MAS, create an instance of event class and set the id of the event and invoke static delete method by passing this instance.

Java – Sample code to delete an event

```
event event_obj = new event();
event_obj.set_id (event_id);
event.delete(mas_client, event_obj);
```

C# – Sample code to delete an event

```
event event_obj = new event();
event_obj.id = event_id;
event.delete(mas_client, event_obj);
```

Python – Sample code to delete an event

```
event_obj = event()
event_obj.id = event_id
event.delete(mas_client, event_obj)
```

Create a Configuration Job on NetScaler MAS

To create a config job, first create an instance of **config_job** class, then set all the mandatory and relevant properties and invoke the **add** static method by passing this instance to the method.

In the following sample codes, the mandatory parameter is the name of the config job. To demonstrate how a job is created, other parameters are also set.

Java – Sample code to create a config job

```
String[]  device_list=new String[10];
device_list[0] = '10.102.201.86-p1';
```

```
config_command config_commandObj= new config_command();
config_commandObj.set_protocol('SSH');
config_commandObj.set_command('show ns config');
String command_list= new String[10];
command_list[0] = config_commandObj;
configuration_template  config_temp= new
configuration_template();
config_temp.set_commands(command_list);
config_job config_job_obj= new config_job();
config_job_obj.set_name('test2');
config_job_obj.set_devices(device_list);
config_job_obj.set_device_family('ns');
config_job_obj.set_template_info(config_temp);
config_job[] simplelist = config_job.add(client,
config_job_obj);
```

C# – Sample code to create a config job

```
string[]  device_list=new string[10];
device_list[0] = '10.102.201.86-p1';
config_command config_commandObj= new config_command();
config_commandObj.protocol='SSH';
config_commandObj.command='show ns config';
string command_list= new string[10];
command_list[0] = config_commandObj;
configuration_template  config_temp= new
configuration_template();
config_temp.commands=command_list;
config_job config_job_obj= new config_job();
config_job_obj.name='test2';
config_job_obj.devices=device_list;
config_job_obj.device_family='ns';
config_job_obj.template_info=config_temp;
config_job[] simplelist = config_job.add(client,
config_job_obj);
```

Python – Sample code to create a config job

```
device_list=[]
device_list.append('10.102.201.86-p1')
config_commandObj=config_command()
config_commandObj.protocol='SSH'
config_commandObj.command='show ns config'
command_list=[]
command_list.append(config_commandObj)
```

```
config_temp=configuration_template()
config_temp.commands=command_list
config_job_obj= config_job()
config_job_obj.name='test2'
config_job_obj.devices=device_list
config_job_obj.device_family='ns'
config_job_obj.template_info=config_temp
simplelist = config_job.add(client, config_job_obj)
```

Add Datacenter Through NetScaler MAS

To add a new datacenter, create an instance of mps_datacenter class and set latitude, longitude and name of the datacenter and invoke “add” static method by passing this instance. Mandatory parameters are latitude, longitude and name.

Java – Sample code to add a datacentre

```
mps_datacenter my_datacenter = new mps_datacenter();
my_datacenter.set_name("new_dc");
my_datacenter.set_latitude(12.9716);
my_datacenter.set_longitude(77.5946);
mps_datacenter simplelist =
mps_datacenter.add(client,my_datacenter);
```

C# – Sample code to add a datacentre

```
mps_datacenter my_datacenter = new mps_datacenter();
my_datacenter.name = "new_dc";
my_datacenter.latitude = 12.9716;
my_datacenter.longitude = 77.5946;
mps_datacenter simplelist =
mps_datacenter.add(client,my_datacenter);
```

Python – Sample code to add a datacentre

```
my_datacenter = mps_datacenter()
my_datacenter.name = "new_dc"
my_datacenter.latitude = float(12.9716)
my_datacenter.longitude = float(77.5946)
simplelist = mps_datacenter.add(client,my_datacenter)
```

Delete a Datacenter from NetScaler MAS

To delete a datacenter from NetScaler MAS, create an instance of mps_datacenter class and set the id of the datacentre, and invoke static delete method by passing this instance. The id of the datacenter named new_dc is fetched and then the datacenter is deleted using the id.

Java – Sample code to delete a datacentre

```
mps_datacenter my_datacenter = new mps_datacenter();
String filter_value = "name:new_dc";
mps_datacenter simplelist =
mps_datacenter.get_filtered(client,filter_value);
mps_datacenter my_datacenter = new mps_datacenter();
my_datacenter.set_id(simplelist[0].id);
mps_datacenter simplelist =
mps_datacenter.delete(client,my_datacenter);
```

C# – Sample code to delete a datacentre

```
mps_datacenter my_datacenter = new mps_datacenter();
string filter_value = "name:new_dc";
mps_datacenter simplelist =
mps_datacenter.get_filtered(client,filter_value);
mps_datacenter my_datacenter = new mps_datacenter();
my_datacenter.id= simplelist[0].id;
mps_datacenter simplelist =
mps_datacenter.delete(client,my_datacenter);
```

Python – Sample code to delete a datacentre

```
my_datacenter = mps_datacenter()
filter_value = "name:new_dc"
simplelist = mps_datacenter.get_filtered(client,filter_value)
my_datacenter = mps_datacenter()
my_datacenter.id= str(simplelist[0].id)
simplelist = mps_datacenter.delete(client,my_datacenter)
```

Retrieve Details of NetScaler SDX Instances from NetScaler MAS

To retrieve NetScaler SDX instances from MAS, invoke static **get** method:

Java – Sample code to retrieve NetScaler SDX instances

```
nssdx simplelist = nssdx.get(client);
for(nssdx item : simplelist){
print("name of the NetScaler SDX: " + item.get_name());
print("id of the NetScaler SDX: " + item.get_id());
}

C# – Sample code to retrieve NetScaler SDX instances

nssdx simplelist = nssdx.get(client);
foreach (nssdx item in simplelist){
print("name of the NetScaler SDX: " + item.name);
print("id of the NetScaler SDX: " + item.id);
}

Python – Sample code to retrieve NetScaler SDX instances

simplelist = nssdx.get(client)
for item in simplelist:
print("name of the NetScaler SDX: " + item.name)
print("id of the NetScaler SDX: " + item.id)
```

Retrieve Count of Docker Host Instances from NetScaler MAS

To retrieve count of docker host instances from NetScaler MAS, invoke static **count** method:

```
Java – Sample code to retrieve Count of Docker_host instances
int count = docker_host.count(client);
print("Count of Docker hosts: "+ count);

C# – Sample code to retrieve Count of Docker_host instances
int count = docker_host.count(client);
print("Count of Docker hosts: "+ count);

Python – Sample code to retrieve Count of Docker_host
instances
count = docker_host.count(client)
print("Count of Docker hosts: "+ str(count))
```

Retrieve Details of Stylebooks from NetScaler MAS

To retrieve details of stylebooks from NetScaler MAS, invoke static **get** method:

Java – Sample code to retrieve details of stylebooks

```
stylebooks simplelist = stylebooks.get(client) ;
for(stylebooks item : simplelist){
print("Name of the stylebook: "+ item.get_display_name());
print("Source of the stylebook: "+ item.get_source());
}
```

C# – Sample code to retrieve details of stylebooks

```
stylebooks simplelist = stylebooks.get(client) ;
foreach(stylebooks item in simplelist){
print("Name of the stylebook: "+ item.display_name);
print("Source of the stylebook: "+ item.source);
}
```

Python – Sample code to retrieve details of stylebooks

```
simplelist = stylebooks.get(client)
for item in simplelist:
print("Name of the stylebook: "+ item.display_name)
print("Source of the stylebook: "+ str(item.source)
```