Citrix NetScaler MAS NITRO Developer's Guide for REST API

Citrix® NetScaler® 11.1

Last Updated: June 2016

Document code: June 24 2016

Contents

Chapter 1

NITRO API

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).

Obtaining the Latest NITRO Package

The latest NITRO package is available as a tar file on the **Downloads** page of the NetScaler MAS server's configuration utility. 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 classpath 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.

Prerequisites

To use the NITRO protocol, the client application needs only the following:

- Access to a NetScaler MAS, release 11.1.
- 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.
- For Java clients, you must have a system where Java Development Kit (JDK) 1.5 or later is available. The JDK

can be downloaded from http://www.oracle.com/technetwork/java/javase/downloads/index.html.

- 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.

Note: You must have a basic understanding of the NetScaler MAS server before using the NITRO protocol.

Chapter 2

Execution Flow

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:

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

Chapter 3

API Categorization and Usage

Topics:

The NITRO functionality can be grouped into the following:

Note: All NITRO requests are synchronous. After sending a request, the client application blocks until it receives a response from the NITRO web service.

System Management

This section describes the system level operations that can be performed on the NetScaler MAS.

- Connect to the NetScaler MAS on page 12
- Disconnect from the NetScaler MAS on page 12

The following table describes the system level operations. For a more detailed description of the operations, see the API reference available in the `<NITRO_SDK_HOME>/doc` folder.

Connect to the NetScaler MAS

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. The session ID that is created must be specified in the cookie field of all requests in the session.
 	
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 MAS with IP address 10.102.31.16 by using the HTTP protocol:
	
	• **URL.** http://10.102.31.16/nitro/v1/config/login
	• **HTTP Method.** POST
	• **Request Payload.**
	`object=`
	`{`
	`    "login":`
	`    {`
	`        "username":"nsroot",`
	`"password":"verysecret"`
	`    }`
	`}`
	• **Response Payload.**
	`{`
	`    "errorcode": 0,`
	`    "message":"Done",`
	`"sessionid":"##78C060..."`
	`}`

Disconnect from the NetScaler MAS

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

Description	Sample
To disconnect from the appliance, use the DELETE HTTP method.	To disconnect from a NetScaler MAS with IP address 10.102.31.16: ◆ **URL.** http://10.102.31.16/nitro/v1/config/login ◆ **HTTP Method.** DELETE ◆ **Cookie.** SESSID=##78C060...

Configurations

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

- **For Java and .NET.** 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 resource has the `com.citrix.mas.nitro.resource.config.ns` package or namespace.

- **For REST.** Each NetScaler MAS resource has an unique URL associated with it, depending on the type of operation to be performed. URLs for configuration operations have the format `http://<MAS-IP>/nitro/v1/config/<resource_type>`.

  For example, to configure a NetScaler resource, the URL is `http://10.102.31.16/nitro/v1/config/ns`.

This section describes the configurations that can be performed on the NetScaler MAS

The following tables describe the configurations operations that can be performed. For a more detailed description of the operations, see the API reference available in the `<NITRO_SDK_HOME>/doc` folder.

Create Resource

Description	Sample

To create a new resource (for example, a NetScaler instance) on the NetScaler MAS, specify the resource name and other related arguments in the specific resource object.	To create a NetScaler instance named vpx1:
	◆ **URL.** http://10.102.31.16/nitro/v1/config/ns
	◆ **HTTP Method.** POST
	◆ **Cookie.** SESSID=##78C060...
	◆ **Request Payload.**
	`object=`
	`{`
	`    "ns":`
	`    {`
	`        "name":"vpx1",`
	`"ip_address":"192.168.100.2",`
	`"netmask":"255.255.255.0",`
	`"gateway":"192.168.100.1",`
	`"image_name":"nsvpx-9.3-45_nc.xva",`
	`        "vm_memory_total":2048,`
	`        "throughput":1000,`
	`        "pps":1000000,`
	`"license":"Standard",`
	`"profile_name":"ns_nsroot_profile",`
	`        "username":"admin",`
	`        "password":"admin",`
	`        "if_10_1":"true"`
	`    }`
	`}`

Retrieve Resource Details

Description	Sample
NetScaler MAS resource details can be retrieved as follows:	To retrieve the configuration information for the NetScaler instance with ID 123456a:

◆ To retrieve details of a specific resource on the NetScaler MAS, specify the id of the resource in the URL. ◆ To retrieve the properties of resources on the basis of some filter, specify the filter conditions in the URL. The URL has the form: `http://<MAS-IP>/nitro/v1/config/<resource_type>?filter=<property1>:<value>,<property2>:<value>`. ◆ If your request is likely to result in a large number of resources returned from the appliance, you can retrieve these results in chunks by dividing them into "pages" and retrieving them page by page. For example, assume that you want to retrieve all NetScaler instances on a NetScaler MAS that has 53 of them. Instead of retrieving all 53 in one big response, you can configure the results to be divided into pages of 10 NetScaler instances each (6 pages total), and retrieve them from the server page by page. You specify the page count with the `pagesize` query string parameter and use the `pageno` query string parameter to specify the page number that you want to retrieve. The URL has the form: `http://<MAS-IP>/nitro/v1/config/<resource_type>?pageno=<value>&pagesize=<value>`. You do not have to retrieve all the pages, or retrieve the pages in order. Each request is independent, and you can even change the pagesize setting between requests. **Note:** If you want to have an idea of the number of resources that are	◆ **URL.** http://10.102.31.16/nitro/v1/config/ns/123456a ◆ **HTTP Method.** GET ◆ **Cookie.** SESSID=##78C060... ◆ **Response Payload.** `{ "errorcode":0, "message":"Done", "ns": [ { "name":"vpx1", "id":"123456a", "ip_address":"192.168.100.2", "gateway":"192.168.100.1", "netmask":"255.255.255.255", "vm_state":"DOWN", "vm_memory_total":2048, ... } ] }`

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. To get the number of NetScaler instances available, the URL would be `http://<MAS_IP>/nitro/v1/config/<resource_type>?count=yes`.	

Retrieve Resource Statistics

Description	Sample
To retrieve the statistics of a resource, use the GET HTTP method. You will receive a response payload that includes the statistics information.	To retrieve the configuration information for the NetScaler instance with ID 123456a: ◆ **URL.** http://10.102.31.16/nitro/v1/config/ns/123456a ◆ **HTTP Method.** GET ◆ **Cookie.** SESSID=##78C060... ◆ **Response Payload.** (see below)

```
{
    "errorcode":0,
    "message":"Done",
    "ns":
    [
        {
            "name":"vpx1",
            "id":"123456a",

"ip_address":"192.168.100.2"
,

"gateway":"192.168.100.1",

"netmask":"255.255.255.255",

"vm_state":"DOWN",
           "vm_memory_total":
2048,
            ...
        }
    ]
}
```

Note: Not all NetScaler MAS resources have statistic attributes associated with them.

Update Resource

Description	Sample

To update an existing NetScaler MAS resource, use the PUT HTTP method. In the HTTP request payload, specify the name and the other arguments that have to be changed.	To change the name of NetScaler instance with ID 123456a to vpx2:
	◆ **URL.** http://10.102.31.16/nitro/v1/config/ns
	◆ **HTTP Method.** PUT
	◆ **Cookie.** SESSID=##78C060...
	◆ **Request Payload.**
	`object=`
	`{`
	`    "ns":`
	`    {`
	`        "name":"vpx2",`
	`        "id":"123456a"`
	`    }`
	`}`

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

Description	Sample
To delete an existing resource, specify the name of the resource to be deleted in the URL.	To delete a NetScaler instance with ID 123456a: ◆ **URL.** http://10.102.31.16/nitro/v1/config/ns/123456a ◆ **HTTP Method.** DELETE ◆ **Cookie.** SESSID=##78C060...

Bulk Operations

You can query or change the configuration of resources simultaneously by, for example adding multiple NetScalers in the same operation. This minimizes the network traffic.

NITRO supports the following behavior 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 2 NetScaler resources in one operation:
	
	- **URL.** http://10.102.29.60/nitro/v1/config/ns
	- **HTTP Method.** POST
	- **Cookie.** sessionid="##78C060..."
	- **Request Payload.**
	
	`object=`
	`{`
	`    "ns":`
	`    [`
	`        {`
	
	`"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"`
	`        }`
	`    ]`
	`}`
	
	To add multiple resources (two NetScalers and two MPS users) in one operation:
	
	- **URL.** http://10.102.29.60/nitro/v1/config/

- **HTTP Method.** POST
- **Cookie.** sessionid="##78C060..."
- **Request Payload.**

```
object=
{
    "ns":
    [
        {

"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",

"permission":"superuser"
        },
        {
            "name":"admin",

"password":"admin",

"permission":"superuser"
        }
    ]
}
```

Exception 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 appliance.

The response payload of all operations, specifies the error code and error message. For example, to get the status of a operation:

- **URL.** http://10.102.31.16/nitro/v1/config/ping
- **HTTP Method.** GET
- **Cookie.** sessionid="##78C060..."
- **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.