

Network Services Hub

REST API Guide

Version: 1.0

Status: Portfolio sample

Owner: Rima Sharma

This sample REST API guide is created for portfolio purposes. All endpoints, server URLs, and payloads represent a fictional network environment.

Network Services Hub REST API

The Network Services Hub REST API provides a programmatic interface for configuring and managing virtual network services. By utilizing standard HTTP methods, administrators and external management systems can automate operational tasks, including security service configuration, network interface management, and system maintenance.

The Network Services Hub REST API follows a RESTful request–response model where clients send requests to specific endpoints and receive data in JSON format. This architecture enables seamless integration of Network Services Hub functionality into orchestration workflows and third-party management platforms.

API Access Information

Use the following connection parameters when accessing the Network Services Hub REST API.

Attribute	Value
Base URL	https://nsh.example.com
Data format	JSON
Authentication	HTTP Basic

All API endpoints described in this guide are relative to the base URL.

Authentication

The Network Services Hub REST API uses HTTP Basic Authentication. Each request must include an `Authorization` header containing base64-encoded credentials, as shown in the following example:

```
Authorization: Basic YWRtaW46cGFzc3dvcmQ=
```

The encoded value represents the credentials `admin:password`.

For demonstration purposes, this guide uses the following credentials:

- Username: admin
- Password: password

Response Codes

Status	Name	Description
200	OK	The request succeeded, and the response contains the requested data.
201	Created	The resource was successfully created.
400	Bad Request	The request was invalid due to missing or incorrect parameters.
401	Unauthorized	Authentication failed or credentials were not provided.
404	Not Found	The requested resource does not exist on the server.

RADIUS

Endpoint Summary

Method	Endpoint
GET	/api/nsh/security/radius/servers
POST	/api/nsh/security/radius/servers
GET	/api/nsh/security/radius/servers/{serverAddress}
DELETE	/api/nsh/security/radius/servers/{serverAddress}
PATCH	/api/nsh/security/radius/servers/settings

Get RADIUS Servers

Method

GET

Endpoint

/api/nsh/security/radius/servers

Description

Retrieves information about all configured RADIUS servers.

Request Example

```
curl -u admin:password -X GET
```

https://nsh.example.com/api/nsh/security/radius/servers

Response Example

HTTP/1.1 200 OK

Content-Type: application/json

```
{
  "servers": [
    {
      "address": "1.1.1.1",
      "secret": {
        "key": "7",
        "sharedSecret": "wawy3"
      },
      "authPort": 1812,
      "acctPort": 1813
    },
    {
      "address": "2.2.2.2",
      "secret": {
        "key": "0",
        "sharedSecret": "plainkey"
      },
      "authPort": 1812,
      "acctPort": 1813
    }
  ]
}
```

Add a RADIUS Server

Method

POST

Endpoint

/api/nsh/security/radius/servers

Description

Registers a new RADIUS server for centralized authentication and accounting services.

Request Parameters

Parameter	Type	Required	Description
address	string	Yes	IPv4 address of the RADIUS server.
key	string	Yes	The encryption mode of the associated secret: 0: Clear text (unencrypted) 7: Encrypted (proprietary format)

sharedSecret	string	Yes	Preshared secret to authenticate communication between the RADIUS server and Network Services Hub. A preshared secret is alphanumeric and case-sensitive, and it can have a maximum of 63 characters.
authPort	integer	No	The UDP port for authentication traffic. The default port is 1812, and the valid range is 0 to 65535.
acctPort	integer	No	The UDP port for accounting and logging traffic. The default port is 1813, and the valid range is 0 to 65535.

Request Examples

Valid Request Example

```
curl -u admin:password -X POST
https://nsh.example.com/api/nsh/security/radius/servers
-H "Content-Type: application/json"
-d '{
  "address": "10.10.1.1",
  "sharedKey": "7",
  "sharedSecret": "myRadiusPassword",
  "authPort": 1812,
  "acctPort": 1813
}'
```

Invalid Request Example

```
curl -u admin:password -X POST
https://nsh.example.com/api/nsh/security/radius/servers
-H "Content-Type: application/json"
-d '{
  "sharedKey": "7",
  "sharedSecret": "myRadiusPassword",
  "authPort": 1812,
  "acctPort": 1813
}'
```

Response Examples

Valid Response Example

```
HTTP/1.1 201 Created
Content-Type: application/json
{
  "message": "RADIUS server successfully created",
  "server": {
    "address": "10.10.1.1",
    "authPort": 1812,
    "acctPort": 1813
  }
}
```

Invalid Response Example

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
{
  "error": "Invalid request",
  "message": "Parameter 'address' is required"
}
```

Get a RADIUS Server

Method

GET

Endpoint

/api/nsh/security/radius/servers/{serverAddress}

Description

Retrieves configuration details of a RADIUS server.

Path Parameters

Parameter	Type	Required	Description
serverAddress	string	Yes	IPv4 address of the RADIUS server

Request Example

```
curl -u admin:password -X GET
https://nsh.example.com/api/nsh/security/radius/servers/3.3.3.3
```

Response Example

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "server": {
    "address": "3.3.3.3",
    "secret": {
      "key": "7",
      "sharedSecret": "wawy3"
    },
    "authPort": 1812,
    "acctPort": 1813
  }
}
```

Delete a RADIUS Server

Method

DELETE

Endpoint

/api/nsh/security/radius/servers/{serverAddress}

Description

Deletes the specified RADIUS server from the system configuration.

Path Parameters

Parameter	Type	Required	Description
serverAddress	string	Yes	IPv4 address of the RADIUS server

Request Example

```
curl -u admin:password -X DELETE  
https://nsh.example.com/api/nsh/security/radius/servers/3.3.3.3
```

Response Example

```
HTTP/1.1 200 OK  
Content-Type: application/json  
{  
  "message": "RADIUS server successfully removed"  
}
```

Configure RADIUS Timeout and Retransmission Settings

Method

PATCH

Endpoint

/api/nsh/security/radius/servers/settings

Description

Configures the duration to wait for a response from the RADIUS server before declaring a timeout failure or the number of retransmits allowed before reverting to local authentication.

Request Parameters

Parameter	Type	Required	Description
timeout	integer	Yes	Time in seconds to wait for a server response
retransmit	integer	Yes	Number of retry attempts before reverting to local authentication

Request Example

```
curl -u admin:password -X PATCH  
https://nsh.example.com/api/nsh/security/radius/servers/settings  
-H "Content-Type: application/json"  
-d '{"timeout": 10, "retransmit": 5}'
```

Response Example

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "message": "RADIUS settings successfully updated",
  "settings": {
    "timeout": 10,
    "retransmit": 5
  }
}
```

LLDP

Endpoint Summary

Method	Endpoint
GET	/api/nsh/network/interfaces/{interfaceName}/lldp
PATCH	/api/nsh/network/interfaces/{interfaceName}/lldp

Get LLDP Configuration

Method

GET

Endpoint

/api/nsh/network/interfaces/{interfaceName}/lldp

Description

Retrieves the current LLDP state for a physical network interface.

Path Parameters

Parameter	Type	Required	Description
interfaceName	string	Yes	Name of the physical network interface

Request Example

```
curl -u admin:password -X GET
https://nsh.example.com/api/nsh/network/interfaces/eth0/lldp
```

Response Example

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "interface": "eth0",
  "lldp": "enabled"
}
```

Enable or Disable LLDP

Method

PATCH

Endpoint

/api/nsh/network/interfaces/{interfaceName}/lldp

Description

Configures the LLDP state for a physical network interface.

Path Parameters

Parameter	Type	Required	Description
interfaceName	string	Yes	Name of the physical network interface

Request Parameters

Parameter	Type	Required	Description
lldp	string	Yes	LLDP state (enabled or disabled)

Request Example

```
curl -u admin:password -X PATCH
https://nsh.example.com/api/nsh/network/interfaces/eth0/lldp
-H "Content-Type: application/json"
-d '{"lldp": "enabled"}
```

Response Example

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "interface": "eth0",
  "lldp": "enabled",
  "message": "LLDP configuration successfully updated"
}
```

Configuration

Endpoint Summary

Method	Endpoint
POST	/api/nsh/configuration/save
POST	/api/nsh/configuration/load

Save the Running Configuration

Method

POST

Endpoint

/api/nsh/configuration/save

Description

Saves the current running configuration to a file in the Network Services Hub repository.

Request Parameters

Parameter	Type	Required	Description
fileName	string	Yes	Name of the configuration file

Request Example

```
curl -u admin:password -X POST
https://nsh.example.com/api/nsh/configuration/save
-H "Content-Type: application/json"
-d '{"fileName": "backup-config.sav"}'
```

Response Example

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "message": "Configuration successfully saved"
}
```

Load a Saved Configuration

Method

POST

Endpoint

/api/nsh/configuration/load

Description

Loads a saved configuration file from the repository and applies it to the Network Services Hub.

Request Parameters

Parameter	Type	Required	Description
fileName	string	Yes	Name of the configuration file

Request Example

```
curl -u admin:password -X POST
https://nsh.example.com/api/nsh/configuration/load
-H "Content-Type: application/json"
-d '{"fileName": "backup-config.sav}"'
```

Response Example

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "message": "Configuration successfully loaded"
}
```