About NX-API

On Cisco Nexus devices, command-line interfaces (CLIs) are run only on the device. NX-API improves the accessibility of these CLIs by making them available outside of the switch by using HTTP/HTTPS. You can use this extension to the existing Cisco Nexus CLI system on the Cisco Nexus 9000 Series devices. NX-API supports `show` commands, configurations, and Linux Bash.

NX-API supports JSON-RPC.

Transport

NX-API uses HTTP/HTTPS as its transport. CLIs are encoded into the HTTP/HTTPS POST body.

The NX-API backend uses the Nginx HTTP server. The Nginx process, and all of its children processes, are under Linux cgroup protection where the CPU and memory usage is capped. If the Nginx memory usage exceeds the cgroup limitations, the Nginx process is restarted and restored.

Message Format

NX-API is an enhancement to the Cisco Nexus 9000 Series CLI system, which supports XML output. NX-API also supports JSON output format for specific commands.

Note

- NX-API XML output presents information in a user-friendly format.
- NX-API XML does not map directly to the Cisco NX-OS NETCONF implementation.
- NX-API XML output can be converted into JSON.
Security

NX-API supports HTTPS. All communication to the device is encrypted when you use HTTPS.

NX-API is integrated into the authentication system on the device. Users must have appropriate accounts to access the device through NX-API. NX-API uses HTTP basic authentication. All requests must contain the username and password in the HTTP header.

Note

You should consider using HTTPS to secure your user's login credentials.

You can enable NX-API by using the feature manager CLI command. NX-API is disabled by default.

NX-API provides a session-based cookie, nxapi_auth when users first successfully authenticate. With the session cookie, the username and password are included in all subsequent NX-API requests that are sent to the device. The username and password are used with the session cookie to bypass performing the full authentication process again. If the session cookie is not included with subsequent requests, another session cookie is required and is provided by the authentication process. Avoiding unnecessary use of the authentication process helps to reduce the workload on the device.

Note

A nxapi_auth cookie expires in 600 seconds (10 minutes). This value is a fixed and cannot be adjusted.

Note

NX-API performs authentication through a programmable authentication module (PAM) on the switch. Use cookies to reduce the number of PAM authentications, which reduces the load on the PAM.

Using NX-API

The commands, command type, and output type for the Cisco Nexus 9000 Series devices are entered using NX-API by encoding the CLIs into the body of a HTTP/HTTPS POST. The response to the request is returned in XML or JSON output format.

Note

For more details about NX-API response codes, see Table of NX-API Response Codes.

You must enable NX-API with the feature manager CLI command on the device. By default, NX-API is disabled.

The following example shows how to configure and launch the NX-API Sandbox:

- Enable the management interface.

```
switch# config
switch(config)# interface mgmt 0
switch(config)# ip address 198.51.100.1/24
switch(config)# vrf context management
switch(config)# ip route 203.0.113.1/0 1.2.3.1
```
- Enable the NX-API nxapi feature.

```
switch# conf t
switch(config)# feature nxapi
```

The following example shows a request and its response in XML format:

Request:
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ins_api>
  <version>0.1</version>
  <type>cli_show</type>
  <chunk>0</chunk>
  <sid>session1</sid>
  <input>show switchname</input>
  <output_format>xml</output_format>
</ins_api>
```

Response:
```
<?xml version="1.0"?>
<ins_api>
  <type>cli_show</type>
  <version>0.1</version>
  <sid>eoc</sid>
  <outputs>
    <output>
      <body>
        <hostname>switch</hostname>
      </body>
      <input>show switchname</input>
      <msg>Success</msg>
      <code>200</code>
    </output>
  </outputs>
</ins_api>
```

The following example shows a request and its response in JSON format:

Request:
```
{
  "ins_api": {
    "version": "0.1",
    "type": "cli_show",
    "chunk": "0",
    "sid": "session1",
    "input": "show switchname",
    "output_format": "json"
  }
}
```

Response:
```
{
  "ins_api": {
    "type": "cli_show",
    "version": "0.1",
    "sid": "eoc",
    "outputs": {
      "output": {
        "body": {
```
Sample NX-API Scripts

The sample scripts demonstrate how a script is used with NX-API. The scripts are available at [https://github.com/datacenter/nexus9000/tree/master/nx-os/nxapi/check_cable](https://github.com/datacenter/nexus9000/tree/master/nx-os/nxapi/check_cable).

- Cable Checker (check_cable.py)
- Cable Checker Blueprint (connectivity.json)

NX-API Sandbox

The NX-API Sandbox is the web-based user interface that you use to enter the commands, command type, and output type for the Cisco Nexus 9000 Series device using HTTP/HTTPS. After posting the request, the output response is displayed.

By default, NX-API is disabled. Begin enabling NX-API with the `feature` manager CLI command on the switch. Then enable NX-API with the `nxapi sandbox` command.

Use a browser to access the NX-API Sandbox.

When using the NX-API Sandbox, Cisco recommends that you use the Firefox browser, release 24.0 or later.

The following example shows how to configure and launch the NX-API Sandbox:

- Enable the management interface.
  
  ```
  switch# conf t
  switch(config)# interface mgmt 0
  switch(config)# ip address 198.51.100.1/24
  switch(config)# vrf context management
  switch(config)# ip route 203.0.113.1/0 1.2.3.1
  ```

- Enable the NX-API `nxapi` feature.
  
  ```
  switch# conf t
  switch(config)# feature nxapi
  switch(config)# nxapi sandbox
  ```

- Open a browser and enter `http://mgmt-ip` to launch the NX-API Sandbox. The following figure is an example of a request and output response.
In the NX-API Sandbox, you specify the commands, command type, and output type in the top pane. Click the POST Request button above the left pane to post the request. Brief descriptions of the request elements are displayed below the left pane.

After the request is posted, the output response is displayed in the right pane.

The following sections describe the commands to manage NX-API and descriptions of the elements of the request and the output response.

**NX-API Management Commands**

You can enable and manage NX-API with the CLI commands listed in the following table.

**Table 1: NX-API Management Commands**

<table>
<thead>
<tr>
<th>NX-API Management Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>feature nxapi</td>
<td>Enables NX-API.</td>
</tr>
<tr>
<td>no feature nxapi</td>
<td>Disables NX-API.</td>
</tr>
<tr>
<td>nxapi [http/https] port port</td>
<td>Specifies a port.</td>
</tr>
<tr>
<td>no nxapi [http/https]</td>
<td>Disables HTTP/HTTPS.</td>
</tr>
<tr>
<td>show nxapi</td>
<td>Displays port information.</td>
</tr>
</tbody>
</table>
NX-API Management Command | Description
---|---
**nxapi certificate** `{httpsct|httpskey}` | Specifies the upload of the following:
  - HTTPS certificate when `httpscrt` is specified.
  - HTTPS key when `httpskey` is specified.

**nxapi certificate enable** | Enables a certificate.

### NX-API Request Elements

NX-API request elements are sent to the device in XML format or JSON format. The HTTP header of the request must identify the content type of the request.

You use the NX-API elements that are listed in the following table to specify a CLI command:

**Table 2: NX-API Request Elements**

<table>
<thead>
<tr>
<th>NX-API Request Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>Specifies the NX-API version.</td>
</tr>
<tr>
<td>NX-API Request Element</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>type</td>
<td>Specifies the type of command to be executed. The following types of commands are supported:</td>
</tr>
</tbody>
</table>
|                        | • cli_show
|                        | CLI show commands that expect structured output. If the command does not support XML output, an error message is returned. |
|                        | • cli_show_ascii
|                        | CLI show commands that expect ASCII output. This aligns with existing scripts that parse ASCII output. Users are able to use existing scripts with minimal changes. |
|                        | • cli_conf
|                        | CLI configuration commands. |
|                        | • bash
|                        | Bash commands. Most non-interactive Bash commands are supported by NX-API. |

**Note**

- Each command is only executable with the current user's authority.
- The pipe operation is supported in the output when the message type is ASCII. If the output is in XML format, the pipe operation is not supported.
- A maximum of 10 consecutive show commands are supported. If the number of show commands exceeds 10, the 11th and subsequent commands are ignored.
- No interactive commands are supported.
### NX-API Request Element

<table>
<thead>
<tr>
<th>NX-API Request Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>chunk</strong></td>
<td>Some <code>show</code> commands can return a large amount of output. For the NX-API client to start processing the output before the entire command completes, NX-API supports output chunking for <code>show</code> commands. Enable or disable chunk with the following settings:</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Only <code>show</code> commands support chunking. When a series of <code>show</code> commands are entered, only the first command is chunked and returned. The output message format is XML. (XML is the default.) Special characters, such as <code>&lt;</code> or <code>&gt;</code>, are converted to form a valid XML message (<code>&lt;</code> is converted into <code>&amp;lt;</code> and <code>&gt;</code> is converted into <code>&amp;gt</code>). You can use XML SAX to parse the chunked output.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td><strong>sid</strong></td>
<td>The session ID element is valid only when the response message is chunked. To retrieve the next chunk of the message, you must specify a <code>sid</code> to match the <code>sid</code> of the previous response message.</td>
</tr>
<tr>
<td><strong>input</strong></td>
<td>Input can be one command or multiple commands. However, commands that belong to different message types should not be mixed. For example, <code>show</code> commands are <code>cli_show</code> message type and are not supported in <code>cli_conf</code> mode. Except for <code>bash</code>, multiple commands are separated with <code>&quot;;&quot;</code>. (The <code>;</code> must be surrounded with single blank characters.) For <code>bash</code>, multiple commands are separated with <code>&quot;;&quot;</code>. (The <code>;</code> is <strong>not</strong> surrounded with single blank characters.) The following are examples of multiple commands:</td>
</tr>
<tr>
<td></td>
<td><code>cli_show</code></td>
</tr>
<tr>
<td></td>
<td><code>cli_conf</code></td>
</tr>
<tr>
<td></td>
<td><code>bash</code></td>
</tr>
</tbody>
</table>
The available output message formats are the following:

- **xml**: Specifies output in XML format.
- **json**: Specifies output in JSON format.

**Note**
The Cisco Nexus 9000 Series CLI supports XML output, which means that the JSON output is converted from XML. The conversion is processed on the switch.

To manage the computational overhead, the JSON output is determined by the amount of output. If the output exceeds 1 MB, the output is returned in XML format. When the output is chunked, only XML output is supported.

The content-type header in the HTTP/HTTPS headers indicate the type of response format (XML or JSON).

### NX-API Response Elements

The NX-API elements that respond to a CLI command are listed in the following table:

**Table 3: NX-API Response Elements**

<table>
<thead>
<tr>
<th>NX-API Response Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>NX-API version.</td>
</tr>
<tr>
<td>type</td>
<td>Type of command to be executed.</td>
</tr>
<tr>
<td>sid</td>
<td>Session ID of the response. This element is valid only when the response message is chunked.</td>
</tr>
<tr>
<td>outputs</td>
<td>Tag that encloses all command outputs. When multiple commands are in cli_show or cli_show_ascii, each command output is enclosed by a single output tag. When the message type is cli_conf or bash, there is a single output tag for all the commands because cli_conf and bash commands require context.</td>
</tr>
<tr>
<td>output</td>
<td>Tag that encloses the output of a single command output. For cli_conf and bash message types, this element contains the outputs of all the commands.</td>
</tr>
<tr>
<td>input</td>
<td>Tag that encloses a single command that was specified in the request. This element helps associate a request input element with the appropriate response output element.</td>
</tr>
<tr>
<td>body</td>
<td>Body of the command response.</td>
</tr>
<tr>
<td>NX-API Response Element</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>code</td>
<td>Error code returned from the command execution. NX-API uses standard HTTP error codes as described by the Hypertext Transfer Protocol (HTTP) Status Code Registry (<a href="http://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml">http://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml</a>).</td>
</tr>
<tr>
<td>msg</td>
<td>Error message associated with the returned error code.</td>
</tr>
</tbody>
</table>