



# NX-API

---

- [About NX-API, page 1](#)
- [Using NX-API, page 2](#)

## 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 resource 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# 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
```

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": {
          "hostname": "switch"
        },
        "input": "show switchname",
        "msg": "Success",
        "code": "200"
      }
    }
  }
}
```

}

## 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.

**Note**

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

You must enable NX-API with the **feature** manager CLI command on the switch and use a browser to use the NX-API Sandbox. By default, NX-API is disabled.

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

- Enable the management interface.

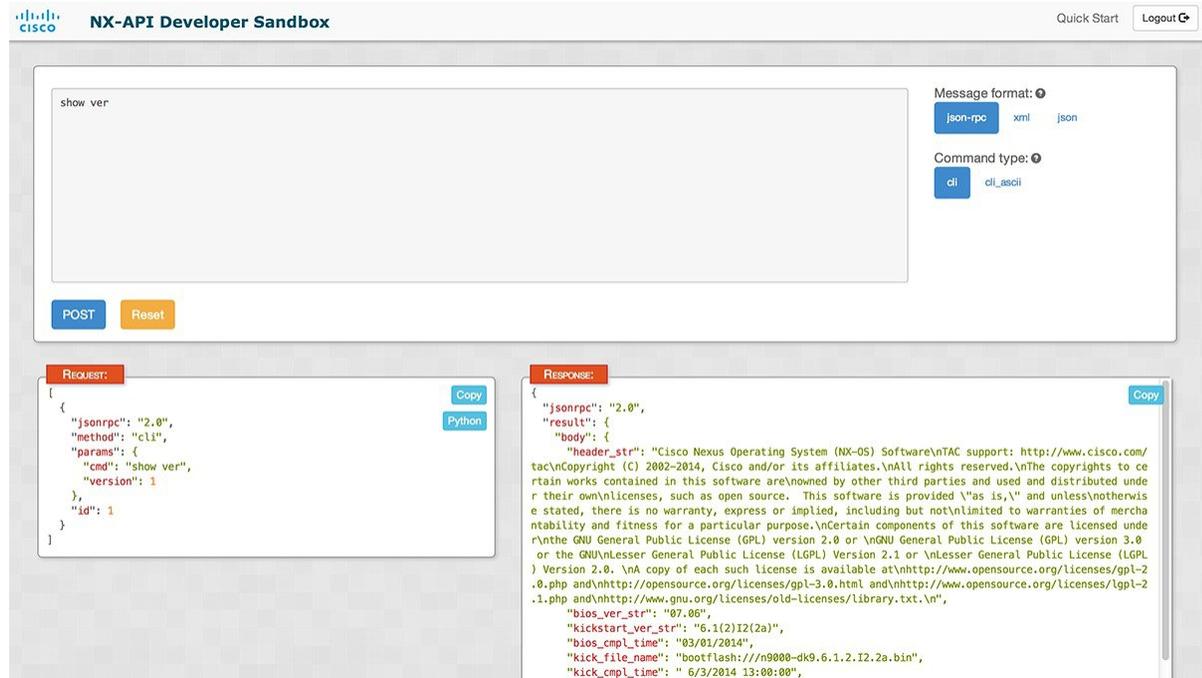
```
switch# conf t
switch(config)# interface mgmt 0
switch(config)# ip address 198.51.100.1/24
switch(config)# vrf context management
switch(config)# ip route 203.0.113.1/0 1.2.3.1
```

- Enable the NX-API **nxapi** feature.

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

- 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.

**Figure 1: NX-API Sandbox with Example 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**

NX-API Management Command	Description
<code>feature nxapi</code>	Enables NX-API.
<code>no feature nxapi</code>	Disables NX-API.
<code>nxapi {http https} port port</code>	Specifies a port.

NX-API Management Command	Description
<b>no nxapi</b> {http https}	Disables HTTP/HTTPS.
<b>show nxapi</b>	Displays port information.
<b>nxapi certificate</b> { <i>httpsct</i>   <i>httpskey</i> }	Specifies the upload of the following: <ul style="list-style-type: none"> <li>• HTTPS certificate when <i>httpsct</i> is specified.</li> <li>• HTTPS key when <i>httpskey</i> is specified.</li> </ul>
<b>nxapi certificate enable</b>	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**

NX-API Request Element	Description
version	Specifies the NX-API version.

NX-API Request Element	Description
<i>type</i>	<p>Specifies the type of command to be executed.</p> <p>The following types of commands are supported:</p> <ul style="list-style-type: none"> <li>• <b>cli_show</b> CLI <b>show</b> commands that expect structured output. If the command does not support XML output, an error message is returned.</li> <li>• <b>cli_show_ascii</b> CLI <b>show</b> commands that expect ASCII output. This aligns with existing scripts that parse ASCII output. Users are able to use existing scripts with minimal changes.</li> <li>• <b>cli_conf</b> CLI configuration commands.</li> <li>• <b>bash</b> Bash commands. Most non-interactive Bash commands are supported by NX-API.</li> </ul> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>• Each command is only executable with the current user's authority.</li> <li>• 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.</li> <li>• A maximum of 10 consecutive <b>show</b> commands are supported. If the number of <b>show</b> commands exceeds 10, the 11th and subsequent commands are ignored.</li> <li>• No interactive commands are supported.</li> </ul>

NX-API Request Element	Description						
<i>chunk</i>	<p>Some <b>show</b> 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 <b>show</b> commands.</p> <p>Enable or disable chunk with the following settings:</p> <table border="1" data-bbox="789 459 1479 590"> <tr> <td data-bbox="789 459 899 522">0</td> <td data-bbox="899 459 1479 522">Do not chunk output.</td> </tr> <tr> <td data-bbox="789 522 899 590">1</td> <td data-bbox="899 522 1479 590">Chunk output.</td> </tr> </table> <p><b>Note</b> Only <b>show</b> commands support chunking. When a series of <b>show</b> commands are entered, only the first command is chunked and returned.</p> <p>The output message format is XML. (XML is the default.) Special characters, such as &lt; or &gt;, are converted to form a valid XML message (&lt; is converted into &amp;lt; ; &gt; is converted into &amp;gt;).</p> <p><b>Note</b> You can use XML SAX to parse the chunked output. When chunking is enabled, the message format is limited to XML. JSON output format is not supported when chunking is enabled.</p>	0	Do not chunk output.	1	Chunk output.		
0	Do not chunk output.						
1	Chunk output.						
<i>sid</i>	<p>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 <i>sid</i> to match the <i>sid</i> of the previous response message.</p>						
<i>input</i>	<p>Input can be one command or multiple commands. However, commands that belong to different message types should not be mixed. For example, <b>show</b> commands are cli_show message type and are not supported in cli_conf mode.</p> <p><b>Note</b> Except for <b>bash</b>, multiple commands are separated with ";". (The ; must be surrounded with single blank characters.)</p> <p>For <b>bash</b>, multiple commands are separated with ";". (The ; is <b>not</b> surrounded with single blank characters.)</p> <p>The following are examples of multiple commands:</p> <table border="1" data-bbox="789 1518 1479 1751"> <tr> <td data-bbox="789 1518 911 1593">cli_show</td> <td data-bbox="911 1518 1479 1593">show version ; show interface brief ; show vlan</td> </tr> <tr> <td data-bbox="789 1593 911 1669">cli_conf</td> <td data-bbox="911 1593 1479 1669">interface Eth4/1 ; no shut ; switchport</td> </tr> <tr> <td data-bbox="789 1669 911 1751">bash</td> <td data-bbox="911 1669 1479 1751">cd /bootflash;mkdir new_dir</td> </tr> </table>	cli_show	show version ; show interface brief ; show vlan	cli_conf	interface Eth4/1 ; no shut ; switchport	bash	cd /bootflash;mkdir new_dir
cli_show	show version ; show interface brief ; show vlan						
cli_conf	interface Eth4/1 ; no shut ; switchport						
bash	cd /bootflash;mkdir new_dir						

NX-API Request Element	Description				
<i>output_format</i>	<p>The available output message formats are the following:</p> <table border="1" data-bbox="824 317 1515 447"> <tbody> <tr> <td data-bbox="824 317 1060 380">xml</td> <td data-bbox="1060 317 1515 380">Specifies output in XML format.</td> </tr> <tr> <td data-bbox="824 380 1060 447">json</td> <td data-bbox="1060 380 1515 447">Specifies output in JSON format.</td> </tr> </tbody> </table> <p><b>Note</b> 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.</p> <p>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.</p> <p>The content-type header in the HTTP/HTTPS headers indicate the type of response format (XML or JSON).</p>	xml	Specifies output in XML format.	json	Specifies output in JSON format.
xml	Specifies output in XML format.				
json	Specifies output in JSON format.				

## 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**

NX-API Response Element	Description
version	NX-API version.
type	Type of command to be executed.
sid	Session ID of the response. This element is valid only when the response message is chunked.
outputs	<p>Tag that encloses all command outputs.</p> <p>When multiple commands are in <code>cli_show</code> or <code>cli_show_ascii</code>, each command output is enclosed by a single output tag.</p> <p>When the message type is <code>cli_conf</code> or <code>bash</code>, there is a single output tag for all the commands because <code>cli_conf</code> and <code>bash</code> commands require context.</p>
output	<p>Tag that encloses the output of a single command output.</p> <p>For <code>cli_conf</code> and <code>bash</code> message types, this element contains the outputs of all the commands.</p>

NX-API Response Element	Description
input	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.
body	Body of the command response.
code	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> ).
msg	Error message associated with the returned error code.