



NX-API CLI

- [NX-API CLI について, 1 ページ](#)
- [NX-API CLI の使用, 3 ページ](#)
- [NX-API 応答コードの表, 11 ページ](#)

NX-API CLI について

NX-API CLI は、Cisco Nexus 9000 シリーズ CLI システムの拡張であり、XML 出力をサポートします。NX-API CLI は、特定コマンドに対して JSON 出力形式もサポートしています。

Cisco Nexus デバイスでは、コマンドライン インターフェイス (CLI) はデバイス上のみで実行されます。NX-API CLI は HTTP/HTTPS を使用してスイッチの外部で CLI を使用できるようにすることで、これらの CLI のアクセシビリティを向上します。この拡張は、Cisco Nexus 9000 シリーズ デバイス上の既存の Cisco Nexus の CLI システムに使用できます。NX-API CLI は、**show** コマンド、設定、Linux Bash をサポートします。

NX-API CLI は、JSON-RPC をサポートします。

Transport

NX-API では、トランスポート層として HTTP または HTTPS を使用します。CLI は、HTTP または HTTPS POST の本体にエンコードされます。

NX-API は、ホスト上またはゲストシェル内でネイティブに動作するアプリケーションの UNIX ドメイン ソケットを通じてサポートされます。

NX-API バックエンドは、Nginx HTTP サーバを使用します。Nginx プロセスおよびその子プロセスのすべては、Linux cgroup の保護が適用され、CPU およびメモリ使用率の上限が設定されます。Nginx のメモリ使用量が cgroup の限度を超えると、Nginx プロセスの再起動と復元が行われます。

メッセージ形式

NX-API は、Cisco Nexus 9000 シリーズの CLI システムの拡張であり、XML 出力をサポートします。NX-API は、特定のコマンドの JSON 出力形式もサポートしています。



(注)

- NX-API の XML 出力には、わかりやすい形式で情報が示されます。
- NX-API の XML は、Cisco NX-OS NETCONF 実装に直接マッピングするわけではありません。
- NX-API の XML 出力は、JSON に変換できます。

セキュリティ

NX-API は、HTTPS をサポートしています。HTTPS を使用する場合、デバイスへのすべての通信は暗号化されます。

NX-API は、デバイスの認証システムに統合されています。ユーザは、NX-API 経由でデバイスにアクセスするための適切なアカウントを持つ必要があります。NX-API では、HTTP 基本認証を使用します。すべての要求の HTTP ヘッダーには、ユーザ名とパスワードが含まれる必要があります。



(注)

ユーザのログイン資格情報を保護するには、HTTPS の使用を検討する必要があります。

feature manager CLI コマンドを使用して、NX-API を有効化できます。NX-API はデフォルトでは無効化されています。

ユーザが初めて認証に成功したときに、NX-API は、**nxapi_auth** というセッションベースの Cookie を提供します。デバイスに送信される後続の NX-API 要求のすべてには、セッション Cookie とともに、ユーザ名とパスワードが含まれます。ユーザ名とパスワードをセッション Cookie とともに使用することにより、完全な認証プロセスの再実行が省略されます。セッション Cookie が後続の要求に含まれていない場合は、別のセッション Cookie が要求され、認証プロセスで提供されません。不要な認証プロセスの回避は、デバイスの負荷軽減に役立ちます。



(注)

nxapi_auth Cookie は、600 秒（10 分）で期限切れとなります。この値は固定されており、変更できません。



(注)

NX-API は、スイッチ上の Programmable Authentication Module (PAM) を使用して認証を行います。Cookie を使用して PAM の認証数を減らすことにより、PAM の負荷が軽減されます。

NX-API CLI の使用

Cisco Nexus 9000 シリーズ デバイスのコマンド、コマンドタイプ、出力タイプは、HTTP/HTTPs POST の本体に CLI をエンコードすることによって、NX-API を使用して入力されます。要求に対する応答は、XML または JSON の出力形式で返されます。



(注) NX-API の応答コードの詳細については、[NX-API 応答コードの表](#)、(11 ページ) を参照してください。

NX-API CLI は、ローカルアクセスに対してデフォルトで有効化されています。リモート HTTP アクセスは、デフォルトで無効化されています。

次の例は、NX-API サンドボックスの設定および起動の方法を示しています。

- 管理インターフェイスを有効にします。

```
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
```

- NX-API `nxapi` 機能を有効にします。

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

次の例は、要求と XML 形式の応答を示しています。

要求 :

```
<?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>
```

応答 :

```
<?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>
```

次の例は、要求と JSON 形式の応答を示しています。

要求：

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

応答：

```
{
  "ins_api": {
    "type": "cli_show",
    "version": "0.1",
    "sid": "eoc",
    "outputs": {
      "output": {
        "body": {
          "hostname": "switch"
        },
        "input": "show switchname",
        "msg": "Success",
        "code": "200"
      }
    }
  }
}
```

NX-API サンプルスクリプト

サンプルスクリプトは、NX-API でどのようにスクリプトが使用されるかを示しています。スクリプトは、https://github.com/datacenter/nexus9000/tree/master/nx-os/nxapi/check_cable から入手できます。

- ケーブルチェッカー (check_cable.py)
- ケーブルチェッカーのブループリント (connectivity.json)
- UDS での NX-API の使用 (rest_client.py)

NX-API サンドボックス

NX-API サンドボックスは、HTTP または HTTPS を使用して Cisco Nexus 9000 シリーズデバイスのコマンド、コマンドタイプ、出力タイプを入力する場合に使用する Web ベースのユーザーインターフェイスです。要求を送信すると、出力応答が表示されます。

デフォルトでは、NX-API は無効化されています。スイッチで **feature manager CLI** コマンドを使用して NX-API の有効化を開始します。次に、**nxapi sandbox** コマンドで NX-API を有効化します。

ブラウザを使用して NX-API サンドボックスにアクセスします。



(注) NX-API サンドボックスを使用する場合、Firefox ブラウザのリリース 24.0 以降を使用することが推奨されます。

次の例は、NX-API サンドボックスの設定および起動の方法を示しています。

- 管理インターフェイスを有効にします。

```
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
```

- NX-API `nxapi` 機能を有効にします。

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

- ブラウザを開いて `http://mgmt-ip` と入力し、NX-API サンドボックスを起動します。次の図は、要求と出力応答の例です。

図 1: 要求と出力応答の例を示した NX-API サンドボックス

NX-API サンドボックスでは、上部のペインにコマンド、コマンドタイプ、出力タイプを指定します。リクエストを送信するには、左ペインの上の [POST Request] ボタンをクリックします。要求要素の簡単な説明が左ペインの下に表示されます。

要求が送信された後、出力応答が右側のペインに表示されます。

NX-API を管理するコマンドの説明および要求と出力応答の要素に関する説明については、以降の項で示します。

NX-API 管理コマンド

次の表に示す CLI コマンドを使用して、NX-API の有効化や管理を行えます。

表 1: NX-API 管理コマンド

NX-API 管理コマンド	説明
feature nxapi	NX-API を有効化します。
no feature nxapi	NX-API を無効化します。
nxapi {http https} portport	ポートを指定します。
no nxapi {http https}	HTTP/HTTPS を無効化します。
show nxapi	ポート情報を表示します。
nxapi certificate {httpsctrl httpskey}	次のいずれかのアップロードを指定します。 <ul style="list-style-type: none"> • <i>httpsctrl</i> が指定されている場合は HTTPS 証明書。 • <i>httpskey</i> が指定されている場合は HTTPS キー。
nxapi certificateenable	証明書を有効化します。
nxapi use-vrfrvf	デフォルト VRF、管理 VRF、または名前付き VRF を指定します。 (注) Cisco NX-OS リリース 7.0(3)I2(1) では、nginx は 1 つの VRF のみをリスニングします。

NX-API 要求要素

NX-API 要求要素は、XML 形式または JSON 形式でデバイスに送信されます。要求の HTTP ヘッダーにより、要求のコンテンツ タイプが特定される必要があります。

次の表に示す NX-API 要素を使用して CLI コマンドを指定できます。

表 2: NX-API 要求要素

NX-API 要求要素	説明
version	NX-API のバージョンを指定します。

NX-API 要求要素	説明
<p><i>type</i></p>	<p>実行するコマンドのタイプを指定します。</p> <p>次のタイプのコマンドがサポートされます。</p> <ul style="list-style-type: none"> • cli_show 構造化された出力を示す CLI show コマンド。コマンドが XML 出力をサポートしない場合、エラーメッセージが返されます。 • cli_show_ascii ASCII 出力を示す CLI show コマンド。これは、ASCII 出力を解析する既存のスクリプトに合わせられます。ユーザは、最小限の変更で既存のスクリプトを使用できます。 • cli_conf CLI 設定コマンド。 • bash Bash コマンド。ほとんどの非インタラクティブ bash コマンドは、NX-API でサポートされています。 <p>(注)</p> <ul style="list-style-type: none"> • 各コマンドは、現在のユーザ権限でのみ実行できます。 • メッセージタイプが ASCII の場合、出力でパイプオペレーションがサポートされます。出力が XML 形式の場合、パイプオペレーションはサポートされません。 • 最大 10 個の連続した show コマンドがサポートされます。show コマンドの数が 10 個を超えると、11 番目以降のコマンドは無視されます。 • インタラクティブ コマンドはサポートされません。

NX-API 要求要素	説明				
<p><i>chunk</i></p>	<p>一部の show コマンドは、多数の出力を返すことができます。コマンド全体が完了する前に出力の処理を開始する NX-API クライアントでは、NX-API は show コマンドの出力チャンクをサポートします。</p> <p>チャンクの有効化または無効化には、次の設定を使用します。</p> <table border="1" data-bbox="787 520 1479 659"> <tbody> <tr> <td data-bbox="787 520 899 590">0</td> <td data-bbox="899 520 1479 590">出力をチャンクしない。</td> </tr> <tr> <td data-bbox="787 590 899 659">1</td> <td data-bbox="899 590 1479 659">出力をチャンクする。</td> </tr> </tbody> </table> <p>(注) show コマンドのみがチャンクをサポートします。連続して show コマンドが入力された場合、最初のコマンドのみがチャンクされて返されます。</p> <p>出力メッセージの形式は XML です (XML がデフォルト)。特殊文字 (< や > など) は有効な XML メッセージになるように変換されます (< は &lt; に変換され、> は &gt; に変換されます)。</p> <p>チャンクされた出力は、XML SAX を使用して解析できます。</p> <p>(注) チャンクを有効化すると、メッセージ形式は XML に限定されます。チャンクが有効化されている場合、JSON 出力形式はサポートされません。</p>	0	出力をチャンクしない。	1	出力をチャンクする。
0	出力をチャンクしない。				
1	出力をチャンクする。				
<p><i>sid</i></p>	<p>セッション ID 要素は、応答メッセージがチャンクされている場合のみ有効です。メッセージの次のチャンクを取得するには、前の応答メッセージの <i>sid</i> と一致するように <i>sid</i> を指定する必要があります。</p>				

NX-API 要求要素	説明						
<p><i>input</i></p>	<p>input には、1 つまたは複数のコマンドを含むことができます。ただし、異なるメッセージタイプに属するコマンドを混同することはできません。たとえば、show コマンドは cli_show メッセージタイプであり、cli_conf モードではサポートされません。</p> <p>(注) bash を除き、複数のコマンドは「;」で区切ります (; の前後にそれぞれ 1 つの空白文字が必要です)。</p> <p>bash の場合、複数のコマンドは「;」で区切ります (; の前後に空白文字を加えません)。</p> <p>次に、複数のコマンドの例を示します。</p> <table border="1" data-bbox="824 716 1515 947"> <tr> <td data-bbox="824 716 948 793">cli_show</td> <td data-bbox="948 716 1515 793">show version ; show interface brief ; show vlan</td> </tr> <tr> <td data-bbox="824 793 948 871">cli_conf</td> <td data-bbox="948 793 1515 871">interface Eth4/1 ; no shut ; switchport</td> </tr> <tr> <td data-bbox="824 871 948 947">bash</td> <td data-bbox="948 871 1515 947">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						
<p><i>output_format</i></p>	<p>使用可能な出力メッセージ形式は次のとおりです。</p> <table border="1" data-bbox="824 1062 1515 1203"> <tr> <td data-bbox="824 1062 1062 1131">xml</td> <td data-bbox="1062 1062 1515 1131">出力を XML 形式で指定します。</td> </tr> <tr> <td data-bbox="824 1131 1062 1203">json</td> <td data-bbox="1062 1131 1515 1203">出力を JSON 形式で指定します。</td> </tr> </table> <p>(注) Cisco Nexus 9000 シリーズの CLI は、XML 出力をサポートします。これは、JSON 出力が XML から変換されることを意味します。この変換はスイッチで処理されます。</p> <p>コンピューティングのオーバーヘッドを管理するために、JSON 出力の処理は出力の量によって決定されます。出力が 1 MB を超える場合、出力は XML 形式で返されます。出力がチャンクされる場合は、XML 出力のみがサポートされます。</p> <p>HTTP/HTTPS ヘッダー内のコンテンツタイプヘッダーは、応答形式のタイプを示しています (XML または JSON)。</p>	xml	出力を XML 形式で指定します。	json	出力を JSON 形式で指定します。		
xml	出力を XML 形式で指定します。						
json	出力を JSON 形式で指定します。						

NX-API 応答要素

CLI コマンドに応答する NX-API 要素を次の表に示します。

表 3: NX-API 応答要素

NX-API 応答要素	説明
version	NX-API のバージョン。
type	実行するコマンドのタイプ。
sid	応答のセッションID。この要素は、応答メッセージがチャンクされている場合のみ有効です。
outputs	すべてのコマンド出力を囲むタグ。 複数のコマンドが <code>cli_show</code> または <code>cli_show_ascii</code> に存在する場合、各コマンドの出力は単一の <code>output</code> タグで囲まれます。 メッセージタイプが <code>cli_conf</code> または <code>bash</code> の場合、 <code>cli_conf</code> および <code>bash</code> コマンドにはコンテキストが必要であるため、すべてのコマンドに対して単一の <code>output</code> タグが使用されます。
output	単一のコマンド出力を含むタグ。 メッセージタイプが <code>cli_conf</code> および <code>bash</code> の場合、この要素にはすべてのコマンドの出力が含まれています。
input	要求で指定された単一のコマンドを囲むタグ。この要素は、要求入力要素を適切な応答出力要素に関連付けるために役立ちます。
body	コマンド応答の本文。
code	コマンドの実行から返されるエラーコード。 NX-API は、Hypertext Transfer Protocol (HTTP) のステータスコードレジストリ (http://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml) に記載されている標準 HTTP エラーコードを使用します。
msg	返されたエラーコードに関連付けられているエラーメッセージ。

NX-API 応答コードの表

以下の NX-API エラー、エラー コード、NX-API 応答のメッセージが表示される可能性があります。

以下の NX-API エラー、エラー コード、NX-API 応答のメッセージが表示される可能性があります。



(注) 標準の HTTP エラー コードは、Hypertext Transfer Protocol (HTTP) のステータス コード レジストリ (<http://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml>) で確認できます。

表 4: NX-API 応答コード

NX-API 応答	コード	メッセージ
SUCCESS	200	成功。
CUST_OUTPUT_PIPEDED	204	要求によって出力が別の場所にパイプされました。
BASH_CMD_ERR	400	入力された Bash コマンドのエラー。
CHUNK_ALLOW_ONE_CMD_ERR	400	チャンクは 1 つのコマンドに対してのみ許可されます。
CLI_CLIENT_ERR	400	CLI 実行エラー。
CLI_CMD_ERR	400	入力された CLI コマンドのエラー。
IN_MSG_ERR	400	要求メッセージが無効です。
NO_INPUT_CMD_ERR	400	input コマンドがありません。
PERM_DENY_ERR	401	権限が拒否されました。
CONF_NOT_ALLOW_SHOW_ERR	405	設定モードで show が許可されていません。
SHOW_NOT_ALLOW_CONF_ERR	405	show モードでは設定が許可されていません。
EXCEED_MAX_SHOW_ERR	413	連続した show コマンドの数が最大数を超えました。最大値は 10 です。

MSG_SIZE_LARGE_ERR	413	応答のサイズが大きすぎます。
BACKEND_ERR	500	バックエンド処理エラー。
FILE_OPER_ERR	500	システム内部のファイル操作エラー。
LIBXML_NS_ERR	500	システム内部の LIBXML NS エラー。
LIBXML_PARSE_ERR	500	システム内部の LIBXML 解析エラー。
LIBXML_PATH_CTX_ERR	500	システム内部の LIBXML パス コンテキストエラー。
MEM_ALLOC_ERR	500	システム内部のメモリ割り当てエラー。
USER_NOT_FOUND_ERR	500	入力またはキャッシュからユーザが見つかりません。
XML_TO_JSON_CONVERT_ERR	500	XML から JSON への変換エラー。
BASH_CMD_NOT_SUPPORTED_ERR	501	Bash コマンドはサポートされていません。
CHUNK_ALLOW_XML_ONLY_ERR	501	チャンクでは、XML 出力のみが許可されています。
JSON_NOT_SUPPORTED_ERR	501	出力が大きいため JSON はサポートされていません。
MSG_TYPE_UNSUPPORTED_ERR	501	メッセージタイプはサポートされていません。
PIPE_OUTPUT_NOT_SUPPORTED_ERR	501	パイプオペレーションはサポートされていません。
PIPE_XML_NOT_ALLOWED_IN_INPUT	501	入力では、パイプ XML は許可されません。
RESP_BIG_JSON_NOT_ALLOWED_ERR	501	応答の出力が大きすぎます。JSON はサポートされていません。
STRUCT_NOT_SUPPORTED_ERR	501	サポートされていない構造化出力です。
ERR_UNDEFINED	600	未定義