



NX-API 開発者サンドボックス

- [NX-API 開発者サンドボックス: 9.2 \(2\) より前の NX-OS リリース](#) (1 ページ)

NX-API 開発者サンドボックス: 9.2 (2) より前の NX-OS リリース

About the NX-API デベロッパー サンドボックス

NX-API Developer Sandbox は、スイッチでホストされる Web フォームです。NX-OS CLI コマンドを同等の XML または JSON ペイロード。

図に示すように、Web フォームは 3 つのペイン（コマンド（上部ペイン）、要求、および応答）を持つ 1 つの画面です。

図 1: リクエストと出力応答の例を含む NX-API デベロッパー サンドボックス

コマンドペインのコントロールを使用すると、サポートされている API のメッセージフォーマット (NX-API REST など) とコマンドタイプ (XML や JSON など) を選択できます。使用可能なコマンドタイプオプションは、選択したメッセージフォーマットによって異なります。

コマンドペインに 1 つ以上の CLI コマンドを入力するか貼り付けると、Web フォームはコマンドを API ペイロードに変換し、構成エラーをチェックし、結果のペイロードを要求ペインに表示します。次に、コマンドペインの POST ボタンを使用して、ペイロードをサンドボックスからスイッチに直接送信することを選択した場合、応答ペインに API 応答が表示されます。

注意事項と制約事項

デベロッパー サンドボックスのガイドラインと制限は次のとおりです：

- サンドボックスで **POST** をクリックすると、コマンドがスイッチにコミットされ、構成または状態が変更される可能性があります。
- 一部の機能構成コマンドは、関連する機能が有効になるまで使用できません。

メッセージフォーマットとコマンドタイプの構成

[メッセージフォーマット (Message Format)] と [コマンドタイプ (Command Type)] は、コマンドペイン (上部ペイン) の右上隅で構成されます。[メッセージフォーマット (Message Format)] で、使用する API プロトコルのフォーマットを選択します。開発者サンドボックスは、次の API プロトコルをサポートしています。

表 1: NX-OS API プロトコル

プロトコル	説明
json-rpc	JSON ペイロードで NX-OS CLI コマンドを配信するために使用できる標準の軽量リモートプロシージャコール (RPC) プロトコル。JSON-RPC 2.0 仕様は、 jsonrpc.org によって概説されています。
xml	XML ペイロードで NX-OS CLI または bash コマンドを配信するための Cisco NX-API 独自のプロトコル。
json	JSON ペイロードで NX-OS CLI または bash コマンドを配信するための Cisco NX-API 独自のプロトコル。
nx-api rest	内部 NX-OS データ管理エンジン (DME) モデルで管理対象オブジェクト (MO) とそのプロパティを操作および読み取るための Cisco NX-API 独自のプロトコル。詳細については、 [Cisco Nexus NX-API リファレンス (Cisco Nexus NX-API References)] を参照してください。
nx yang	構成および状態データ用の YANG (「Yet Another Next Generation」) データモデリング言語。

[メッセージフォーマット (Message Format)] を選択すると、[コマンドタイプ (Command Type)] オプションのセットが[メッセージフォーマット (Message Format)] コントロールのすぐ下に表示されます。[コマンドタイプ (Command Type)] の設定は、入力 CLI を制限でき、[要求 (Request)] と[応答 (Response)] のフォーマットを決定できます。オプションは、選択した[メッセージフォーマット (Message Format)] によって異なります。各[メッセージフォーマット (Message Format)] について、次の表で[コマンドタイプ (Command Type)] オプションについて説明します。

表 2: コマンドタイプ

メッセージ形式	コマンドタイプ
json-rpc	<ul style="list-style-type: none"> • cli — show または構成コマンド • cli_ascii — show または構成コマンド、フォーマットせずに出力
xml	<ul style="list-style-type: none"> • cli_show — コマンドを表示します。コマンドが XML 出力をサポートしていない場合、エラーメッセージが返されます。 • cli_show_ascii — コマンドを表示、フォーマットせずに出力 • cli_conf — 構成コマンド。対話型の構成コマンドはサポートされていません。 • bash — bash コマンド。ほとんどの非対話型 bash コマンドがサポートされています。 (注) スイッチで bash シェルを有効にする必要があります。
json	<ul style="list-style-type: none"> • cli_show — コマンドを表示します。コマンドが XML 出力をサポートしていない場合、エラーメッセージが返されます。 • cli_show_ascii — コマンドを表示、フォーマットせずに出力 • cli_conf — 構成コマンド。対話型の構成コマンドはサポートされていません。 • bash — bash コマンド。ほとんどの非対話型 bash コマンドがサポートされています。 (注) スイッチで bash シェルを有効にする必要があります。
nx-api rest	<ul style="list-style-type: none"> • cli — 構成コマンド

メッセージ形式	コマンドタイプ
nx yang	<ul style="list-style-type: none"> • json — ペイロードに JSON 構造が使用されます • xml — XML 構造がペイロードに使用されます

出力チャンク

大量の show コマンド出力を処理するために、一部の NX-API メッセージフォーマットでは、show コマンドの出力チャンクがサポートされています。この場合、**[チャンクモードを有効にする (Enable chunk mode)]** チェックボックスが、セッション ID (SID) 入力ボックスとともに **[コマンドタイプ (Command Type)]** コントロールの下に表示されます。

チャンクが有効な場合、応答は複数の「チャンク」で送信され、最初のチャンクが即時のコマンド応答で送信されます。応答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セッション ID に設定された **SID** を使用して NX-API 要求を送信する必要があります。

デベロッパー サンドボックスを使用

デベロッパー サンドボックスを使用して CLI コマンドをペイロードに変換する



ヒント オンライン ヘルプは、サンドボックス ウィンドウの右上隅にある **[クイック スタート (Quick Start)]** をクリックすると利用できます。

レスポンス コードやセキュリティ メソッドなどの詳細については、NX-API CLI の章を参照してください。

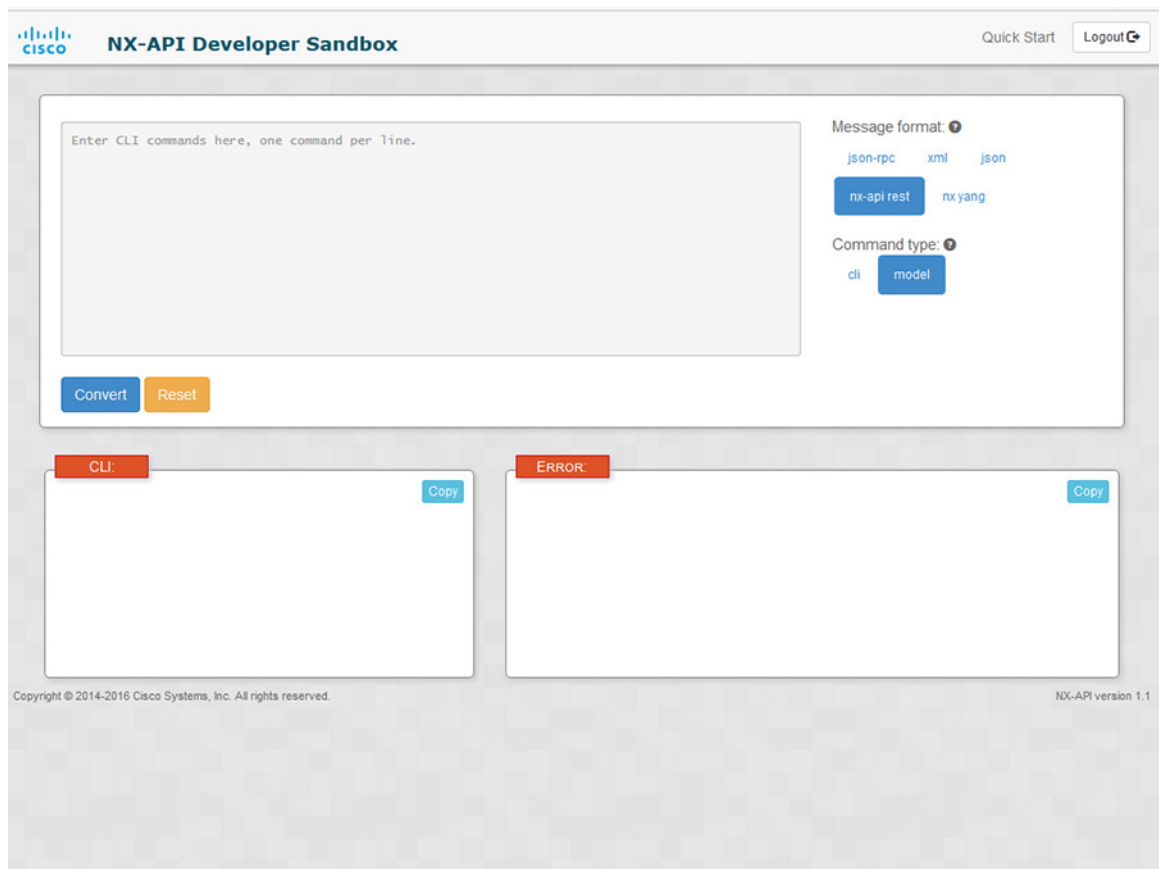
構成コマンドはサポートされていません。

ステップ 1 使用する API プロトコルの **[メッセージ形式 (Message Format)]** と **[コマンドタイプ (Command Type)]** を構成します。

詳細な手順については、[メッセージフォーマットとコマンドタイプの構成 \(2 ページ\)](#) を参照してください。

ステップ 2 上部ペインのテキスト エントリ ボックスに、NX-OS CLI 構成コマンドを 1 行に 1 つずつ入力するか貼り付けます。

上部ペインの下部にある **[リセット (Reset)]** をクリックすると、テキスト エントリ ボックス (および **[要求 (Request)]** ペインと **[応答 (Response)]** ペイン) の内容を消去できます。



ステップ3 トップペインの最下部にある **[変換 (Convert)]** をクリックします。

CLI コマンドに構成エラーが含まれていない場合、ペイロードは **[要求 (Request)]** ペインに表示されます。エラーが存在する場合は、説明のエラーメッセージが **[応答 (Response)]** ペインに表示されます。

デベロッパーサンドボックスを使用して CLI コマンドをペイロードに変換する

The screenshot displays the NX-API Developer Sandbox interface. At the top left is the Cisco logo and the title "NX-API Developer Sandbox". On the top right are "Quick Start" and "Logout" links. The main workspace is divided into several sections:

- Code Editor:** Contains a JSON payload:

```
api/mo/sys.json
{
  "topSystem": {
    "attributes": {
      "name": "REST2CLI"
    }
  }
}
```
- Message format:** A dropdown menu with options "json-rpc", "xml", "json", "nx-api rest" (selected), and "nx.yang".
- Command type:** A dropdown menu with options "cli" and "model" (selected).
- Buttons:** "Convert" (blue) and "Reset" (orange) buttons are located below the code editor.
- CLI Panel:** A red header "CLI:" is above a text area containing "hostname REST2CLI". A "Copy" button is on the right.
- ERROR Panel:** A red header "ERROR:" is above an empty text area. A "Copy" button is on the right.
- Footer:** "Copyright © 2014-2016 Cisco Systems, Inc. All rights reserved." on the left and "NX-API version 1.1" on the right. A status bar at the bottom says "Waiting for bam.nr-data.net..."

ステップ 4 [リクエスト (Request)] ペインに有効なペイロードが表示されている場合は、**POST** をクリックして、ペイロードを API 呼び出しとしてスイッチに送信できます。

スイッチからのレスポンスは **[Response (応答)]** ペインに表示されます。

警告 **POST** をクリックすると、コマンドがスイッチにコミットされ、構成または状態が変更される可能性があります。

The screenshot shows the NX-API Developer Sandbox interface. At the top, there is a header with the Cisco logo, the title "NX-API Developer Sandbox", and links for "Quick Start" and "Logout". The main workspace contains a text input field with the command "logging level netstack 6". To the right of the input, there are dropdown menus for "Message format" (with options: json-rpc, xml, json, nx-api rest, nx yang) and "Command type" (with options: cli, model). Below the input field are three buttons: "POST", "Reset", and "Convert". At the bottom of the interface, there are two panels: "REQUEST:" and "RESPONSE:". The "REQUEST:" panel displays a JSON object:

```
{  "topSystem": {    "children": [      {        "ipv4Entity": {          "children": [            {              "ipv4Inst": {                "attributes": {                  "loggingLevel": "informational"                }              }            ]          }        }      ]    }  }
```

 It includes "Copy", "Python", and "Python3" buttons. The "RESPONSE:" panel displays a JSON object:

```
{  "imdata": []}
```

 It includes a "Copy" button.

ステップ 5 ペインで [コピー (Copy)] をクリックすると、[要求 (Request)] ペインまたは [応答 (Response)] ペインの格納ファイルをクリップボードにコピーできます。

ステップ 6 [リクエスト (Request)] ペインで **Python** をクリックすると、クリップボード上のリクエストの Python 導入を取得できます。

■ デベロッパー サンドボックスを使用して CLI コマンドをペイロードに変換する

翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。