



## NX-API 開発者サンドボックス

---

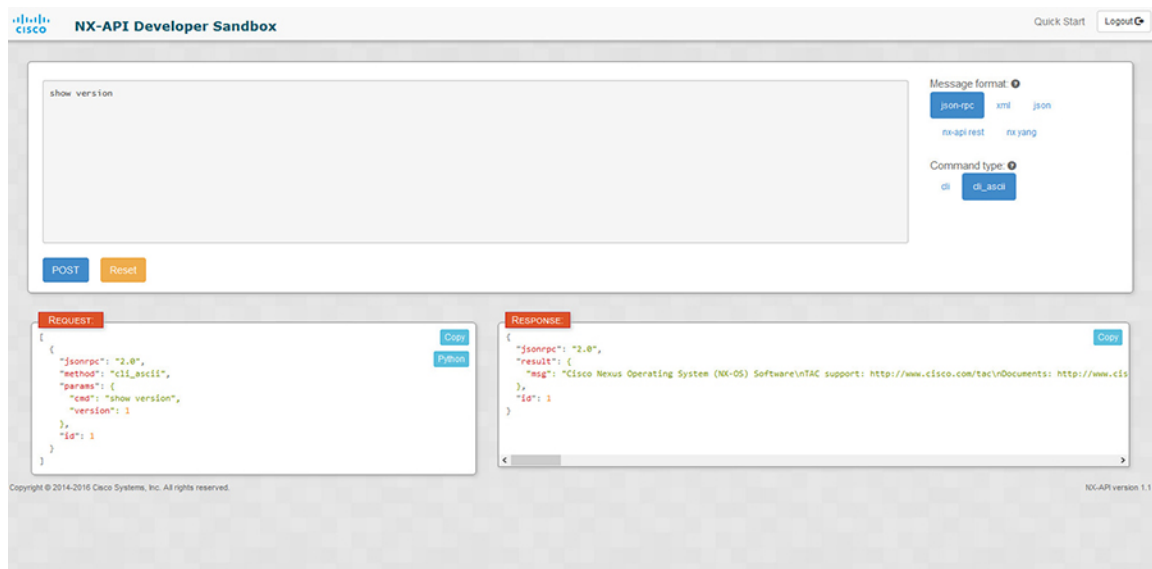
- [About the NX-API デベロッパー サンドボックス \(1 ページ\)](#)
- [デベロッパー サンドボックスの注意事項と制約事項 \(2 ページ\)](#)
- [メッセージフォーマットとコマンドタイプの構成 \(2 ページ\)](#)
- [デベロッパー サンドボックスを使用 \(5 ページ\)](#)
- [デベロッパー サンドボックスを使用して CLI コマンドを REST ペイロードに変換する \(5 ページ\)](#)
- [デベロッパー サンドボックスを使用した REST ペイロードから CLI コマンドへの変換 \(8 ページ\)](#)

### 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 ペイロードに変換し、NX-API REST ペイロードを同等の CLI に変換します。

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。

逆に、コマンドペインに NX-API REST 指定名 (DN) とペイロードを入力し、**nx-api rest** メッセージフォーマットと **[モデル (model)]** コマンドタイプを選択すると、デベロッパーサンドボックスはペイロードの構成エラーをチェックし、応答ペインに同等の CLI が表示されます。

## デベロッパー サンドボックスの注意事項と制約事項

- 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、 <a href="https://jsonrpc.org">jsonrpc.org</a> 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。Cisco Nexus 3000 および 9000 シリーズ NX-API REST SDK の詳細については、 <a href="https://developer.cisco.com/site/cisco-nexus-nx-api-references/">https://developer.cisco.com/site/cisco-nexus-nx-api-references/</a> 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"> <li>cli — show または構成コマンド</li> <li>cli_ascii — show または構成コマンド、フォーマットせずに出力</li> </ul>

メッセージ形式	コマンドタイプ
xml	<ul style="list-style-type: none"> <li>• <code>cli_show</code> — 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。</li> <li>• <code>cli_show_ascii</code> — コマンドを表示、フォーマットせずに出力</li> <li>• <code>cli_conf</code> — 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。</li> <li>• <code>bash</code> — <code>bash</code> コマンド。ほとんどの非対話型 <code>bash</code> コマンドがサポートされています。</li> </ul> <p>(注) スイッチで <code>bash</code> シェルを有効にする必要があります。</p>
json	<ul style="list-style-type: none"> <li>• <code>cli_show</code> — 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。</li> <li>• <code>cli_show_ascii</code> — コマンドを表示、フォーマットせずに出力</li> <li>• <code>cli_conf</code> — 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。</li> <li>• <code>bash</code> — <code>bash</code> コマンド。ほとんどの非対話型 <code>bash</code> コマンドがサポートされています。</li> </ul> <p>(注) スイッチで <code>bash</code> シェルを有効にする必要があります。</p>
nx-api rest	<ul style="list-style-type: none"> <li>• <code>cli</code> — 構成コマンド</li> <li>• モデル — DN および対応するペイロード。</li> </ul>
nx yang	<ul style="list-style-type: none"> <li>• <code>json</code> — ペイロードに JSON 構造が使用されます</li> <li>• <code>xml</code> — XML 構造がペイロードに使用されます</li> </ul>

### 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 REST 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。

### 手順

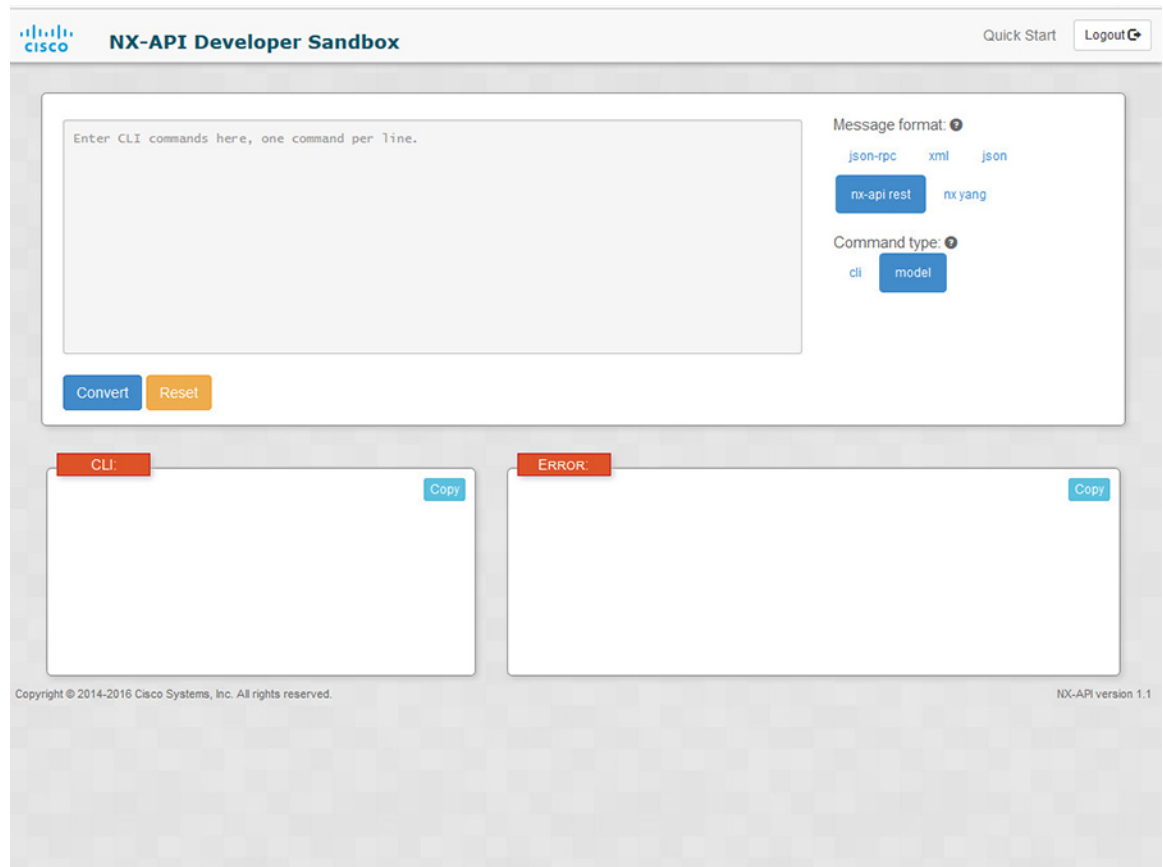
**ステップ 1** 使用する API プロトコルの **[メッセージ形式 (Message Format)]** と **[コマンドタイプ (Command Type)]** を構成します。

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。

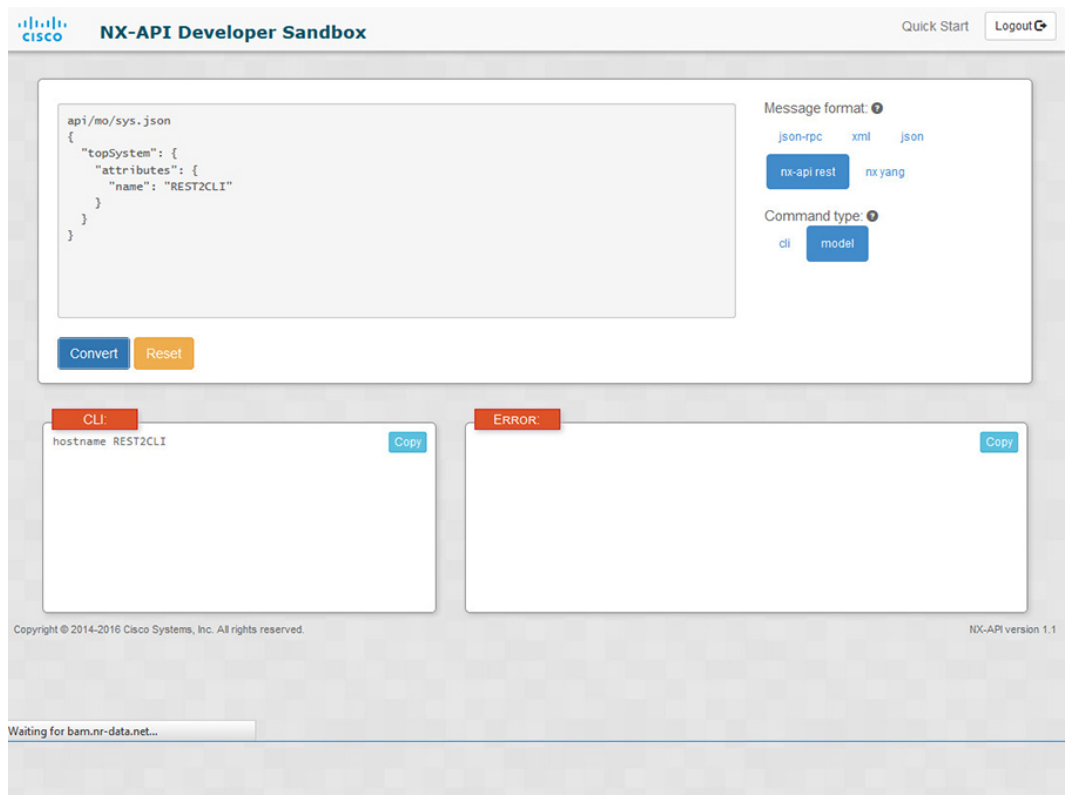
上部ペインの下部にある **[リセット (Reset)]** をクリックすると、テキストエントリボックス (および **[要求 (Request)]** ペインと **[応答 (Response)]** ペイン) の内容を消去できます。

開発者サンドボックスを使用して CLI コマンドを REST ペイロードに変換する



**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。

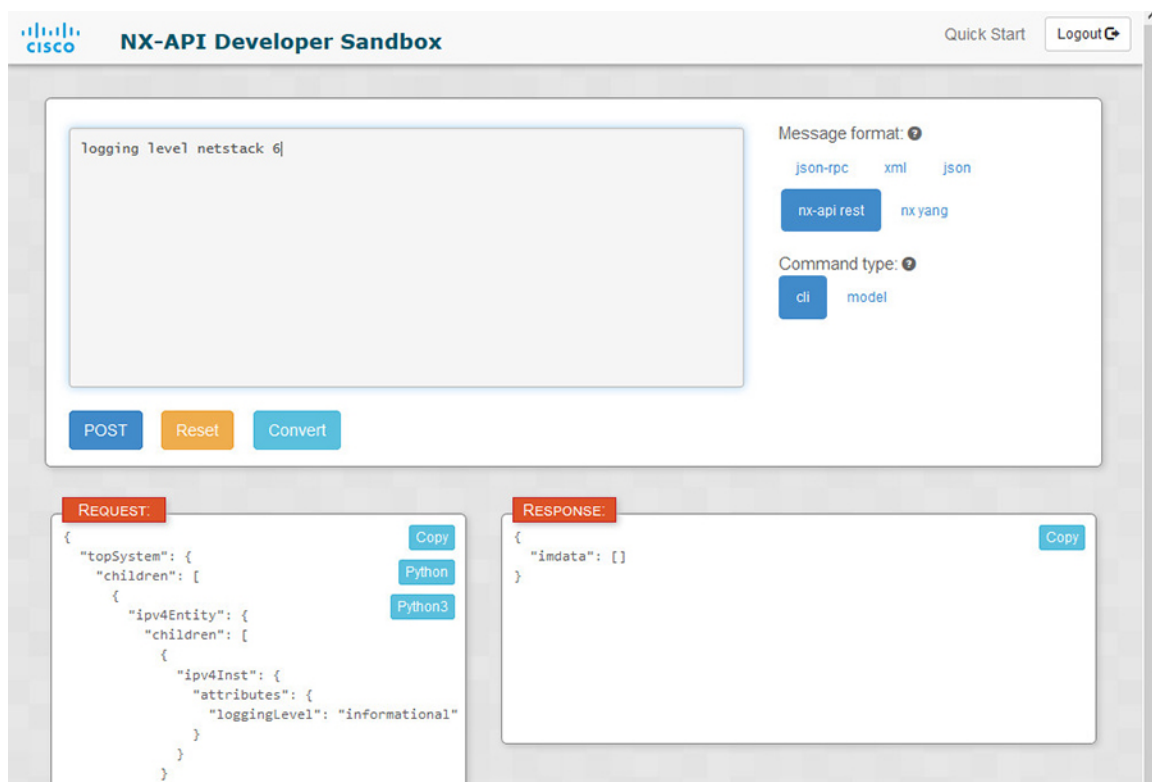
CLI コマンドに構成エラーが含まれていない場合、ペイロードは [要求 (Request)] ペインに表示されます。エラーが存在する場合は、説明のエラーメッセージが [応答 (Response)] ペインに表示されます。



**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。



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た REST ペイロードから CLI コマンドへの変換



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。

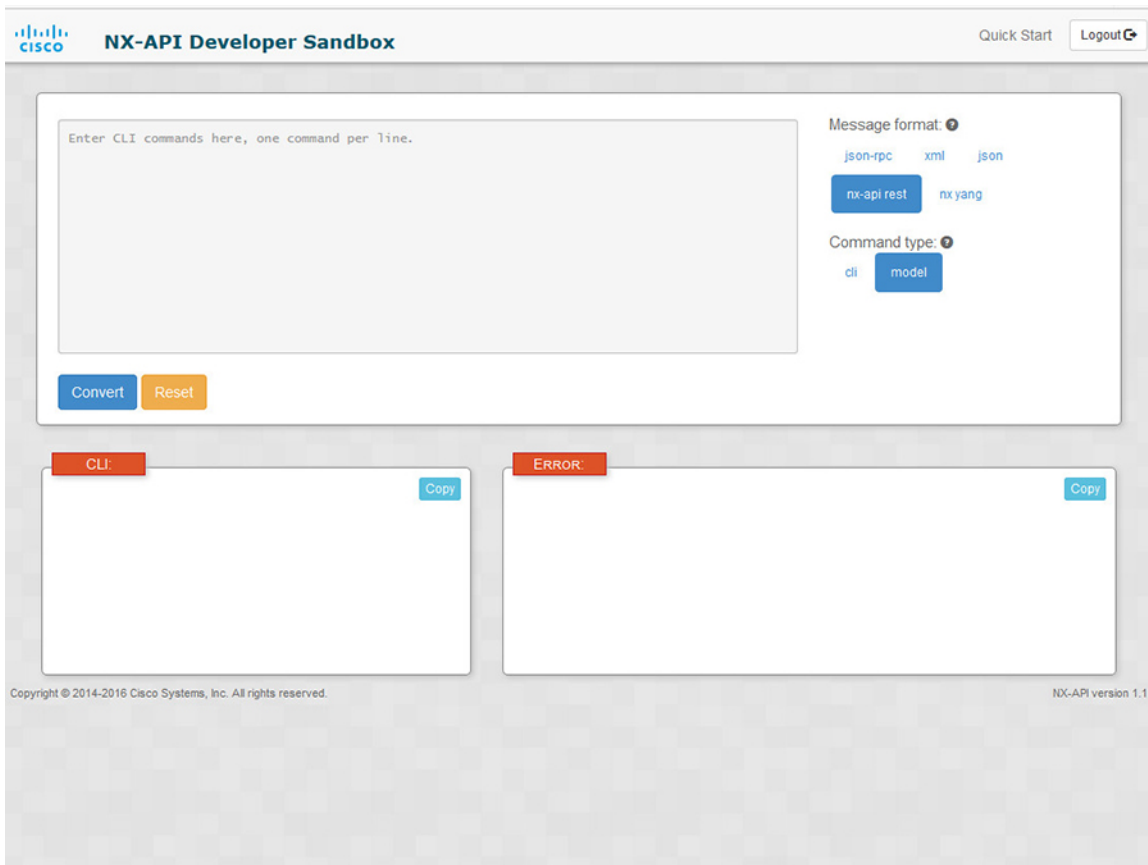
レスポンスコードやセキュリティメソッドなどの詳細については、「NX-API CLI」の章を参照してください。



## 手順

**ステップ 1** メッセージフォーマットとして **nx-api rest** を選択し、コマンドタイプとして **model** を選択します。

例：



The screenshot displays the NX-API Developer Sandbox interface. At the top, there is a header with the Cisco logo and the text "NX-API Developer Sandbox". On the right side of the header, there are links for "Quick Start" and "Logout". The main area features a large text input field with the placeholder text "Enter CLI commands here, one command per line.". To the right of the input field, there are two dropdown menus. The first is labeled "Message format:" and has three options: "json-rpc", "xml", and "json", with "nx-api rest" selected. The second is labeled "Command type:" and has two options: "cli" and "model", with "model" selected. Below the input field, there are two buttons: "Convert" (blue) and "Reset" (orange). At the bottom of the interface, there are two empty panels. The left panel is labeled "CLI" and has a "Copy" button. The right panel is labeled "ERROR" and also has a "Copy" button. The footer contains the text "Copyright © 2014-2016 Cisco Systems, Inc. All rights reserved." and "NX-API version 1.1".

**ステップ 2** 上部ペインのテキスト入力ボックスに DN とペイロードを入力します。次に、上部ペインの下にある [変換 (Convert)] ボタンをクリックします。

例：

この例では、DN は **/api/mo/sys.json** であり、NX-API REST ペイロードは次のとおりです：


```
{
  "topSystem": {
    "attributes": {
      "name": "REST2CLI"
    }
  }
}
```

The screenshot displays the NX-API Developer Sandbox interface. At the top, the Cisco logo and "NX-API Developer Sandbox" are visible, along with "Quick Start" and "Logout" links. The main area contains a text input field with a REST payload:

```
api/mo/sys.json
{
  "topSystem": {
    "attributes": {
      "name": "REST2CLI"
    }
  }
}
```

Below the input field are "Convert" and "Reset" buttons. To the right, 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 these are two empty output boxes labeled "CLI:" and "ERROR:", each with a "Copy" button. At the bottom, there is a status bar with "Waiting for bam.nr-data.net..." and copyright information: "Copyright © 2014-2016 Cisco Systems, Inc. All rights reserved." and "NX-API version 1.1".

[変換 (Convert)] ボタンをクリックすると、次の図に示すように、同等の CLI が CLI ペインに表示されます。

 **NX-API Developer Sandbox**
Quick Start [Logout](#)

```
api/mo/sys.json
{
  "topSystem": {
    "attributes": {
      "name": "REST2CLI"
    }
  }
}
```

Message format: ?

json-rpc   xml   json

nx-api rest   nx yang

Command type: ?

cli   model

Convert
Reset

CLI:

```
hostname REST2CLI
```

[Copy](#)

ERROR:

[Copy](#)

Copyright © 2014-2016 Cisco Systems, Inc. All rights reserved. NX-API version 1.1

Waiting for bam.nr-data.net...

NX-API 開発者サンドボックス

11

(注)

デベロッパー サンドボックスは、サンドボックスが CLI を NX-API REST ペイロードに変換した場合でも、すべてのペイロードを同等の CLI に変換することはできません。以下は、ペイロードが CLI コマンドに完全に変換するのを妨げる可能性のあるエラーの原因のリストです。

表 3: REST2CLI エラーの原因

ペイロードの問題	結果
<p>ペイロードに、MO に存在しない属性が含まれています。</p> <p>例：</p> <pre>api/mo/sys.json {   "topSystem": {     "children": [       {         "interfaceEntity": {           "children": [             {               "l1PhysIf": {                 "attributes": {                   "id": "eth1/1",                   "fakeattribute": "totallyFake"                 }               }             }           ]         }       }     ]   } }</pre>	<p><b>[エラー (Error)]</b> ペインは、属性に関連するエラーを返します。</p> <p>例：</p> <p><b>CLI</b></p> <p>要素「l1PhysIf」の不明な属性「fakeattribute」の<b>[エラー (Error)]</b></p>
<p>ペイロードには、変換がまだサポートされていない MO が含まれています。</p> <p>例：</p> <pre>api/mo/sys.json {   "topSystem": {     "children": [       {         "dhcpEntity": {           "children": [             {               "dhcpInst": {                 "attributes": {                   "SnoopingEnabled": "yes"                 }               }             }           ]         }       }     ]   } }</pre>	<p><b>[エラー (Error)]</b> ペインは、サポートされていない MO に関連するエラーを返します。</p> <p>例：</p> <p><b>CLI</b></p> <p><b>[エラー (Error)]</b> [「sys/dhcp」のサブツリー全体が変換されていません。(The entire subtree of "sys/dhcp" is not converted.) ]</p>

ペイロードの問題	結果
}	}

---

## 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。