



NX-API REST

この章は次のトピックで構成されています。

- [NX-API REST について \(1 ページ\)](#)
- [REST による DME 構成置換 \(2 ページ\)](#)

NX-API REST について

NX-API REST

Cisco Nexus スイッチでは、構成はコマンドライン インターフェイス (CLI) を使用して実施します。CLI は、当該スイッチ上でしか実行できません。NX-API REST は、HTTP/HTTPS API を提供することにより、Cisco Nexus 構成のアクセシビリティを向上させます。

- 特定の CLI コマンドをスイッチの外部から実行可能です。
- 多数の CLI コマンドを発行する必要がある構成を、比較的少数の HTTP/HTTPS 操作の構成アクションを組み合わせることで実現します。

NX-API REST は、**show** コマンド、基本および詳細スイッチ構成と Linux Bash をサポートします。

NX-API REST は HTTP/HTTPS をトランスポートとして使用します。CLI は、HTTP/HTTPS POST 本文にエンコードされます。NX-API REST バックエンドは Nginx HTTP サーバーを使用します。Nginx プロセスとそのすべての子プロセスは、CPU とメモリの使用量の上限が定められている Linux cgroup の保護下に置かれます。NX-API プロセスは、cgroup ext_ser_nginx の一部であり、2,147,483,648 バイトのメモリに制限されています。Nginx のメモリ使用量が cgroup の制限を超えると、Nginx プロセスは再起動されて、NX-API 構成 (VRF、ポート、証明書構成) が復元されません。

Cisco Nexus 3000 および 9000 シリーズ NX-API REST SDK の詳細については、<https://developer.cisco.com/docs/nx-os-n3k-n9k-api-ref/> を参照してください。

RESTによるDME構成置換

REST PutによるDMEフル構成置換について

Cisco NX-OS リリース 9.3(1) 以降、Cisco NX-OS は REST PUT 操作によるモデルベースの完全な設定置換をサポートします。設定を置き換えるこの方法では、Cisco DME モデルを使用します。

DME フル構成置換機能を使用すると、REST プログラム インターフェイスを使用してスイッチの実行構成を置き換えられます。DME フル構成置換機能には、置換を PUT 操作によって行えるという追加のメリットがあります。設定ツリーのすべての部分（システムレベル、サブツリー、およびリーフ）が、DME フル構成置換をサポートします。

- スイッチ構成の無停止交換のサポート
- 自動化のサポート
- 他の機能やその構成に影響を与えることなく、機能を選択的に変更する機能を提供します。
- 最終的な構成結果を指定できるようにすることで、構成変更を簡素化し、人的エラーを排除します。スイッチは差分を計算し、構成ツリーの影響を受ける部分にプッシュします。



(注) プログラム的なインターフェイスを使用して実行することはできませんが、**config replace config-file-name** Cisco NX-OS CLI コマンドにより、フル構成置換を実行することもできます。

Guidelines and Limitations

The following are the guidelines and limitations for the DME full config replace feature:

- For information about supported platforms, see the [Nexus Switch Platform Matrix](#).
- It is important for you to know the tree and know where you are applying the config replace. If you are using the Sandbox for the config replace operation, the Sandbox defaults to the subtree, so you might need to change the URI to target the correct node in the config tree.
- If you use the NX-OS Sandbox to Convert (for Replace), you must use the POST operation because of the presence of the `status: 'replaced'` attribute in the request. If you are using any other conversion option, you can use the PUT operation.
- If you use the REST PUT option for this feature on a subtree node, config replace operation is applied to the entire subtree. The target subtree node is correctly changed with the config replace data in the PUT, but be aware that leaf nodes of the subtree node are also affected by being set to default values.

If you do not want the leaf nodes to be affected, do not use a PUT operation. Instead, you can use a POST operation with the `status: 'replaced'` attribute.

If you are applying the config replace to a leaf node, the PUT operation operates predictably.

REST POST によるプロパティレベルの構成置換

シスコの DME モデルは、REST POST 操作による CLI ベースの機能のプロパティレベルの構成置換をサポートしています。要求ペイロードを生成し、REST POST 操作を介してスイッチに送信することにより、NX-OS サンドボックスを介して機能のプロパティの構成を置き換えることができます。NX-OS サンドボックスの詳細については、[NX-API 開発者サンドボックス](#)を参照してください。

-
- Step 1** HTTPS を介し、NX-OS サンドボックスを介してスイッチに接続し、ログイン情報を入力します。
 - Step 2** 作業エリアで、変更する機能の CLI を入力します。
 - Step 3** 作業エリアの下のフィールドで、構成する機能に対するツリー内の MO への URI を設定します。この MO レベルは Put 要求の送信先です。
 - Step 4** [方法 (Method)] で、NX-API (DME) を選択します。
 - Step 5** [入力タイプ (Input Type)] で、[CLI] を選択します。
 - Step 6** [変換 (Convert)] ドロップダウンリストから Convert (for replace) を選択して、[要求 (Request)] ペインでペイロードを生成します。
 - Step 7** スイッチへの **POST** 操作を使用する要求をクリックします。

(注) プロパティレベルの構成置換は、構成がデフォルト構成の場合に失敗する可能性があります。これは、置換操作はすべての子 MO を削除し、すべてのプロパティをデフォルトにリセットしようと試みるからです。

REST PUT による機能レベルの構成置換

Cisco DME は、REST PUT 操作による機能レベルの構成置換をサポートしています。モデルの機能レベルで PUT を送信することで、特定の機能の構成を置き換えることができます。

次の手順を使用します。

-
- Step 1** クライアントから、機能のモデル オブジェクト (MO) で REST PUT 操作を発行します。
 - a) Put は、最上位システムレベルから機能の MO への URL を指定する必要があります。たとえば、BGP の場合、`/api/mo/sys/bgp.json` のようになります。

ペイロードは有効な構成である必要があります、機能の DN で GET を発行することで、いつでもスイッチから構成を取得できる必要があります。たとえば、BGP の場合、`/api/mo/sys/bgp.json?rsp-subtree=full&rsp-prop-include=set-config-only` のようになります。
 - b) 機能のペイロードは、置き換える MO (たとえば `bgp`) で始まる必要があります。次に例を示します。

```
{
  "bgpInst": {
    "attributes": {
```

```

    "asn": "100",
    "rn": "inst"
  },
  "children": [
    ... content removed for brevity ...

    {
      "bgpDom": {
        "attributes": {
          "name": "vrf1",
          "rn": "dom-vrf1"
        },
        "children": [
          {
            "bgpPeer": {
              "attributes": {
                "addr": "10.1.1.1",
                "inheritContPeerCtrl": "",
                "rn": "peer-[10.1.1.1]"
              }
            }
          }
        ]
      }
    },
    {
      "bgpDom": {
        "attributes": {
          "name": "default",
          "rn": "dom-default",
          "rtrId": "1.1.1.1"
        }
      }
    }
  ]
}

```

Step 2 /api/mo/sys/bgp.json?rsp-subtree=full&rsp-prop-include=set-config-only を使用して、構成の置換に使用した DN で GET を送信します。

Step 3 (オプション) 送信したペイロードを、置き換えた DN の GET と比較します。GET のペイロードは、送信したペイロードと同じである必要があります。

REST PUT の構成置換のトラブルシューティング

以下は、REST Put 操作による構成置換が成功しない場合のトラブルシューティングに役立つ手順です。

Step 1 要求が有効かどうかを確認します。

URL、操作、およびペイロードが有効である必要があります。たとえば、URL が `api/mo/sys/foo.json` の場合、ペイロードは `foo` で始まる必要があります。

Step 2 ペイロードが有効であり、次のような構成プロパティのみが含まれていることを確認します。

- 正常に設定されたもの
- 有効なデバイス構成から取得したもの

構成プロパティのみを取得するには、`rsp-subtree=full&rsp-prop-include=set-config-only` をフィルタリングする GET を使用します。

- Step 3** ペイロードを検証するには、DME POST 操作を使用してペイロードをスイッチに送信します。
- Step 4** エラーをチェックして、MO の名前とプロパティがあることを確認します。
- Step 5** ペイロードにも MO の名前とプロパティが含まれていることを確認します。
-

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。