



RESTCONF エージェント

- [RESTCONF エージェントについて \(1 ページ\)](#)
- [注意事項と制約事項 \(2 ページ\)](#)
- [RESTCONF エージェントの設定 \(2 ページ\)](#)
- [RESTCONF セッションの管理 \(3 ページ\)](#)
- [RESTCONF Get / Set \(3 ページ\)](#)
- [RESTCONF デバイス YANG RPC \(5 ページ\)](#)
- [RESTCONF エージェントのトラブルシューティング \(8 ページ\)](#)

RESTCONF エージェントについて

Cisco NX-OS RESTCONF は、NETCONF で定義されたデータストアを使用して、YANG バージョン 1 で定義されたデータを構成するための HTTP ベースのプロトコルです。

NETCONF は、設定データストアと、これらのデータストアへのアクセスに使用できる一連の作成、取得、更新、および削除 (CRUD) 操作を定義します。YANG 言語は、データストア格納ファイル、運用データ、プロトコル操作、イベント通知の構文とセマンティクスを定義します。

Cisco NX-OS RESTCONF は、HTTP 操作を使用して、YANG 定義のデータを含む概念データストアで CRUD 操作を提供します。このデータは、NETCONF データストアを実装するサーバーと互換性があります。

RESTCONF プロトコルは、XML と JSON の両方のペイロードエンコーディングをサポートします。ユーザー認証は、HTTP 基本認証によって行われます。

次の表に、Cisco NX-OS RESTCONF エージェントがサポートするプロトコル操作を示します。

表 1: サポートされる操作

RESTCONF	同等の NETCONF
オプション	NETCONF : なし
HEAD	NETCONF : なし

GET	NETCONF: <get-config>, <get>
POST	NETCONF: <edit-config> (operation="create")
PUT	NETCONF: <edit-config> (operation="create/replace")
PATCH	NETCONF: <edit-config> (operation="merge")
DELETE	NETCONF: <edit-config> (operation="delete")

注意事項と制約事項

RESTCONF エージェントには、次の注意事項と制約事項があります。

- Cisco NX-OS の RESTCONF は、RESTCONF Protocol というタイトルの RFC ドラフトに基づいています。 <https://tools.ietf.org/html/draft-ietf-netconf-restconf-10> を参照してください。
- Cisco NX-OS リリース 10.4(3)F 以降では、RFC 8040 (<https://datatracker.ietf.org/doc/html/rfc8040>) の次の項目がサポートされています。
 - 3.2 RESTCONF のメディアタイプ
 - 4.2 ヘッド
- TLS については、v1.3 がサポートされています。最低限サポートされるバージョンは v1.2 です。
- Cisco NX-OS リリース 10.4(3)F 以降、RESTCONF は 92348GC-X でサポートされます。

RESTCONF エージェントの設定

この手順では、RESTCONF を有効にして構成する方法について説明します。

始める前に

RESTCONF を使用してスイッチと通信する前に、RESTCONF エージェントを有効にする必要があります。RESTCONF エージェントを有効または無効にするには、**[no] feature restconf** コマンドを入力します。http サーバを実行するには、**feature nxapi** が必要です。

手順の概要

1. **configure terminal**
2. (任意) **feature nxapi**
3. (任意) **nxapi http port port-num**
4. (任意) **nxapi https port port-num**
5. **feature restconf**

手順の詳細

手順

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： switch# configure terminal	グローバル構成モードを開始します。
ステップ 2	(任意) feature nxapi 例： switch(config)# feature nxapi	HTTP サーバ用の nxapi を有効にします。
ステップ 3	(任意) nxapi http port port-num 例： switch(conf-tm-sub)# nxapi http port 80	使用するポートを指定します。デフォルトのポートは 80 です。
ステップ 4	(任意) nxapi https port port-num 例： switch(conf-tm-sub)# nxapi https port 443	https に使用するポートを指定します。デフォルトのポートは 443 です。
ステップ 5	feature restconf 例： switch(conf-tm-sensor)# feature restconf	restconf を有効にします。

RESTCONF セッションの管理

RESTCONF は HTTP 上で実行されます。スイッチ上の NX-API サーバは、管理ポート IP アドレスのポート 80 / 443（または設定されたポート）でリッスンします。クライアントは HTTP サーバとの接続を確立でき、RESTCONF サブシステムは「/restconf」URL にフックします。

クライアントは、RESTCONF エージェントと対話するために継続期間の長い HTTP 接続を維持できますが、通常は単一の要求を送信して終了する方が簡単です。説明については、以下のセクションを参照してください。

RESTCONF Get / Set

次に、Linux の **curl** コマンドを使用した要求の例を示します。

<GET>

この操作により、指定したデータストアの構成データを取得します。

次に、HTTP の Get 要求と応答メッセージの例を示します。

```
client-host % curl -X GET -H "Authorization: Basic YWRtaW46Y2lzY28=" -H "Accept:
application/yang.data+xml"
"http://192.0.20.123/restconf/data/Cisco-NX-OS-device:System/bgp-items/inst-items/dom-items/Dom-list?content=config"
-i
HTTP/1.1 200 OK
Server: nginx/1.7.10
Date: Tue, 27 Sep 2016 20:26:03 GMT
Content-Type: application/yang.data+xml
Content-Length: 395
Connection: keep-alive
Set-Cookie: nxapi_auth=admin:147500856185650327
Status: 200 OK
  <Dom-list>
    <name>default</name>
    <always>enabled</always>
    <bestPathIntvl>300</bestPathIntvl>
    <holdIntvl>180</holdIntvl>
    <kaIntvl>60</kaIntvl>
    <maxAsLimit>0</maxAsLimit>
    <pfxPeerTimeout>30</pfxPeerTimeout>
    <pfxPeerWaitTime>90</pfxPeerWaitTime>
    <reConnIntvl>60</reConnIntvl>
    <rtrId>2.2.2.2</rtrId>
  </Dom-list>
client-host %
```

<POST>

この操作は、指定された設定をターゲットデータストアに書き込みます。

次に、HTTP の POST 要求と応答メッセージの例を示します。

```
client-host % curl -X POST -H "Authorization: Basic YWRtaW46Y2lzY28=" -H "Content-Type:
application/yang.data+xml" -d '<always>enabled</always><rtrId>2.2.2.2</rtrId>'
"http://192.0.20.123/restconf/data/Cisco-NX-OS-device:System/bgp-items/inst-items/dom-items/Dom-list=default"
-i
HTTP/1.1 201 Created
Server: nginx/1.7.10
Date: Tue, 27 Sep 2016 20:25:31 GMT
Transfer-Encoding: chunked
Connection: keep-alive
Set-Cookie: nxapi_auth=admin:14750085316974134
Status: 201 Created
Location: /System/bgp-items/inst-items/dom-items/Dom-list=default/always/rtrId/
```

エフェメラルデータの取得

エフェメラルデータは、大量のデータです。DME は、各バッチがメモリ使用量の観点から管理可能なサイズになるように、データを取得するバッチメカニズムを提供します。バッチのサイズは、取得する MO の数です。

パブリッシュされた Cisco-NX-OS-device.yang ファイルの「Ephemeral data」というコメントで、どのデータがエフェメラルであるかに関する情報を確認できます。

要求内の RESTCONF URI が以下を指している場合にのみ、エフェメラルデータからの出力が返されます。

- エフェメラルデータのリーフ

- エフェメラル データの子を持つコンテナまたはリスト
- 直接のエフェメラルデータの子を持つリストをラップするために使用される空のコンテナ
- システム レベルの GET クエリは、エフェメラル データを返しません。

例

これは、エフェメラル データを取得する例です。

クライアントは、次の GET 要求メッセージを送信する場合があります。

```
GET
/restconf/data/Cisco-NX-OS-device:System//urib-items/table4-items/Table4-list=management/route4-items
HTTP/1.1
Host: example.com
Accept: application/yang.data+json
```

サーバは次のように応答します。

```
HTTP/1.1 200 OK
Date: Fri, 06 Mar 2020 11:10:30 GMT
Server: nginx/1.7.10
Content-Type: application/yang.data+json
{
  "route4-items": {
    "Route4-list": [{
      "prefix": "172.23.167.255/32", "flags": "0", ...
```

RESTCONF デバイス YANG RPC

RESTCONF の操作コマンドについて

この機能は、スイッチでモデル駆動型操作コマンドを実行する方法を提供します。

次に、サポートされている実行 RPC のリストを示します。RPC に関する情報は、公開されている Cisco-NX-OS-device.yang ファイルで確認できます。

表 2: デバイス YANG RPC

操作	デバイス YANG RPC	CLI
チェックポイント	checkpoint	checkpoint <name> checkpoint <file>
ロールバック	ロールバック	rollback running-config checkpoint rollback running-config checkpoint

インストールするもの	install_all_nxos install_add install_activate install_deactivate install_commit install_remove	すべての nxos をインストールする install {add アクティブ化 非アクティブ化}
暗号化証明書のインポート	import_ca_certificate	crypto ca import <trustpoint> pkcs12
スイッチのリロードまたはモジュールのリロード	reload	reload [タイマー<seconds>] モジュールのリロード<module name>
ファイルのコピー	copy	コピー<destination><source>

デバイス YANG RPC の例

チェックポイントの作成

クライアントは、次のような POST 要求メッセージを送信する場合があります。

```
POST /restconf/operations/Cisco-NX-OS-device:checkpoint
Accept: application/yang.operation+json,application/yang.errors+json
Content-type: application/yang.operation+json
Body: {
  "input": {
    "name": "checkpoint-1",
    "description": "testing checkpoint through Restconf" }
}
```

サーバは次のように応答します。

```
HTTP/1.1 204 No content
```

ロールバック (Rollback)

クライアントは、次のような POST 要求メッセージを送信する場合があります。

```
POST /restconf/operations/Cisco-NX-OS-device:rollback
Accept: application/yang.operation+json,application/yang.errors+json
Content-type: application/yang.operation+json
Body: {
  "input": {
    "name": "checkpoint-1",
    "action": "create"
  }
}
```

サーバは次のように応答します。

```
HTTP/1.1 204 No content
```

インストールするもの

クライアントは、次のような POST 要求メッセージを送信する場合があります。

```
POST /restconf/operations/Cisco-NX-OS-device:install_all_nxos
Accept: application/yang.operation+json,application/yang.errors+json
Content-type: application/yang.operation+json
Body: {
  "input": {
    "nxos": "bootflash:nxos.10.1.1-jcco.bin" }
}
```

サーバは次のように応答します。

```
HTTP/1.1 204 No content
```

ca 証明書 RPC のインポート

クライアントは、次のような POST 要求メッセージを送信する場合があります。

```
POST /restconf/operations/Cisco-NX-OS-device:import_ca_certificate
Accept: application/yang.operation+json,application/yang.errors+json
Content-type: application/yang.operation+json
Body: {
  "input": {
    "trustpoint": "mytrustpoint",
    "pkcs12": "bootflash:server.pfx",
    "passphrase": "mypassphrase"
  }
}
```

サーバは次のように応答します。

```
HTTP/1.1 204 No content
```

スイッチのリロード

クライアントは、次のような POST 要求メッセージを送信する場合があります。

```
POST /restconf/operations/Cisco-NX-OS-device:reload
Accept: application/yang.operation+json,application/yang.errors+json
Content-type: application/yang.operation+json
Body: {
  "input": {
  }
}
```

サーバは次のように応答します。

```
HTTP/1.1 204 No content
```

モジュールのリロード

クライアントは、次のような POST 要求メッセージを送信する場合があります。

```
The client might send the following POST request message:
POST /restconf/operations/Cisco-NX-OS-device:reload
Accept: application/yang.operation+json,application/yang.errors+json
Content-type: application/yang.operation+json
Body: {
  "input": {
    "module": "31"
  }
}
```

```
}
}
```

サーバは次のように応答します。

```
HTTP/1.1 204 No content
```

ファイル RPC のコピー

クライアントは、次のような POST 要求メッセージを送信する場合があります。

```
POST /restconf/operations/Cisco-NX-OS-device:reload
Accept: application/yang.operation+json,application/yang.errors+json
Content-type: application/yang.operation+json
Body: {
  "input": {
    "source": "tftp://10.1.1.1/users/myname/config1.txt",
    "destination": "bootflash:",
    "vrf": "management"
  }
}
```

サーバは次のように応答します。

```
HTTP/1.1 204 No content
```

RESTCONF エージェントのトラブルシューティング

機能ステータスの確認

- Cisco NX-OS で、`show feature | inc restconf` コマンドを入力してエージェントの構成を確認します。
- RESTCONF エージェントのステータスを表示するには、**show feature** コマンドを使用し、式 `restconf` を含めます。

```
switch-1# show feature | grep restconf
restconf 1 enabled
switch-1#
```

NX- API 接続の確認

- **feature nxapi** コマンドを発行して、Web サーバーを有効にします。
- HTTP 用にポートを開くように **nxapi http port 80** コマンドが構成されていることを確認します。
- HTTPS 用にポートを開くように **nxapi https port 443** コマンドが構成されていることを確認します。
- スイッチの管理ポートに **ping** を実行して、スイッチが到達可能であることを確認します。

Restconf エラーの確認

次に、一般的なエラーメッセージと、それを解決するためのガイドラインを示します。

- 要求を送信した直後（数秒後など）にこのメッセージを受信した場合は、次の点を確認してください：

Error Message: *Sorry, the page you are looking for is currently unavailable*

- 「接続のトラブルシューティング」に記載されているように、NXAPI 機能が有効になっていること。
- RESTCONF 機能が有効になっていること（**show feature | grep restconf**）。RESTCONF が有効になっていない場合は、有効にします（**feature restconf**）。
- ポートは、NX-API によって HTTP または HTTPS 用に構成されます。show nxapi を使用して、ポートが構成されていることを確認します。

```
switch-1# show nxapi
nxapi enabled
HTTP Listen on port 80
HTTPS Listen on port 443
```

ポートが HTTP または HTTPS 用に構成されていない場合は、**nxapi http port 80** または **nxapi https port 443**を発行して構成します。

- 要求の送信後にこのメッセージを受信した場合（数分後など）、スイッチの最上位レベルからの過剰な同時クエリー要求でシステムが過負荷になっていないことを確認します。過剰な最上位レベルのクエリは、かなりのリソース負荷を生じる可能性があります。

次のいずれかによって、スイッチが過負荷になっていないことを確認できます。

- クライアントが送信する要求の数をスロットルバックします。
- スイッチで **no feature restconf** そして **feature restconf** を発行して RESTCONF エージェントを再起動します。

RESTCONF エージェントのアカウンティング ログ

POST、PUT、PATCH、またはDELETEなどの書き込み操作の場合、RESTCONF は関連するアカウンティングログを出力します。これには、受信した元の要求と、スイッチに適用された最終的な変更の両方が含まれます。

アカウンティングログは、show accounting log コマンドを使用して表示できます。

次の要求例を考えてみます。

```
---
curl -s -L --request POST --user admin: --header 'Content-Type: application/yang.data+json'
--url "restconf/data/Cisco-NX-OS-device:System/intf-items/lb-items/LbRtdIf-list=lo10"
--data-raw @request.txt
Payload:
<descr>test</descr>
Or
```

```
{ "descr": "test" }
---
```

アカウントティング ログには、次の項目が含まれます。

表 3: スイッチに適用される変更

項目	説明
コンテキスト	セッション ID とユーザー
オペレーション	コミット/中止
データベース	実行または候補
ConfigMO	MO ツリーのテキスト表現。最大 3,000 文字。
ステータス	成功/失敗

例:

```
Wed Jun 29 13:53:37
2022:type=update:id=3180018864:user=admin:cmd=(COMMIT),database=[running],configMo=[
<topSystem childAction="" dn="sys" status="created,modified"><interfaceEntity
childAction=""
rn="intf" status="created,modified"><13LbRtdIf childAction="" descr="test" id="lo10"
rn="lb-
[lo10]" status="created,modified"/></interfaceEntity></topSystem>] (SUCCESS)
```

表 4: 受信した元の要求

項目	説明
コンテキスト	セッション ID とユーザー
オペレーション	RESTCONF:POST、RESTCONF:PUT、RESTCONF:PATCH、RESTCONF:DELETE
送信元 IP (Source IP)	RESTCONF クライアント IP
URL	HTTP URL
ペイロード	受信したXML/JSONリクエスト。最大 3,000 文字。
ステータス	成功/失敗

例:

```
Wed Jun 29 13:53:37
2022:type=update:id=3180018864:user=admin:cmd=(RESTCONF:POST),sourceIp=[192.168.1.2],
url=[/restconf/data/Cisco-NX-OS-device:System/intf-items/lb-items/lbRtdIfList=lo10],payload=[<descr>test</descr>]
(SUCCESS)
```

失敗した要求の場合、失敗のシナリオによっては、ユーザーは両方のログを確認できない場合があります。

無効な要求 :

無効な要求は、構成の変更なしに拒否されるため、元の要求のみがログに記録されます。

例:

```
Wed Jun 29 20:16:26
2022:type=update:id=3180018864:user=admin:cmd=(RESTCONF:POST),
sourceIp=[192.168.1.2],url=[/restconf/data/Cisco-IW-OS-device:System/intf-items/lb-items/LbRtdIfList=ld10],payload=[<descr>test</descr>]
(FAILED)
```

さまざまな構成制限による要求の失敗：

この場合、失敗した構成試行と元の要求の両方がログに記録されます。

例：

```
Wed Jun 29 20:32:01
2022:type=update:id=3180018864:user=admin:cmd=(COMMIT),database=[running],
configMo=[<topSystem childAction="" dn="sys" status="created,modified"><telemetryEntity
childAction="" rn="tm" status="created,modified"><telemetryCertificate childAction=""
filename="foo" hostname="foo" rn="certificate" status="created,modified"
trustpoint="test"/></telemetryEntity></topSystem>] (FAILED)
Wed Jun 29 20:32:01
2022:type=update:id=3180018864:user=admin:cmd=(RESTCONF:PATCH),
sourceIp=[192.168.1.2],url=[/restconf/data/Cisco-IW-OS-device:System/intf-items/certificate-items],payload=[<trustpoint test/><trustpoint><hostname>foo/</hostname><filename>
foo</filename></trustpoint>] (FAILED)
```


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。