



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 エージェントの使用方法 \(2 ページ\)](#)
- [RESTCONF エージェントのトラブルシューティング \(3 ページ\)](#)
- [エフェメラル データ \(5 ページ\)](#)
- [実行操作 \(6 ページ\)](#)
- [RESTCONF エージェントのアカウンティング ログ \(8 ページ\)](#)

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設定するための 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。

RESTCONF	同等の NETCONF
オプション	NETCONF : なし
HEAD	NETCONF : なし
GET	NETCONF: <get-config>, <get>

RESTCONF	同等の NETCONF
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 RESTCONF は、RESTCONF Protocol draft-ietf-netconf-restconf-10 というタイトルの RFC ドラフトに基づいています。[RESTCONF Protocol draft-ietf-netconf-restconf-10](#) を参照します。
- RESTCONF は、RFC 6536 で指定されている拡張ロールベース アクセス コントロール (RBAC) をサポートしていません。「network-admin」ロールを持つユーザのみに、RESTCONF エージェントへのアクセス権が付与されます。
- 10.2(1) F リリース以降、操作チェックポイント、ロールバック、インストール、CA 証明書のインポート、スイッチのリロード、個々のモジュールのリロード、およびファイルのコピーがサポートされています。

RESTCONF エージェントの使用方法

一般的なコマンド

- HTTP または HTTPS アクセスを有効にするには、次のコマンドを設定します。
 - **feature nxapi**
 - **nxapi http port 80**
 - **nxapi https port 443**

一般的な制御コマンド

RESTCONF エージェントの **[no] feature restconf** コマンドを有効または無効にできます。

エージェントステータスの表示

RESTCONF エージェントのステータスを表示するには、コマンドを使用し、式 **restconf** を含めます。 **show feature**

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

Curl を使用したサーバーへの POST 要求の送信

```
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:147500853169574134
Status: 201 Created
Location: /System/bgp-items/inst-items/dom-items/Dom-list=default/always/rtrId/
```

Curl を使用したサーバーへの GET 要求の送信

```
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 %
```

RESTCONF エージェントのトラブルシューティング

接続のトラブルシューティング

- **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。

トラブルシューティングのエラー

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。

エラーメッセージ：申し訳ありませんが、検索しているページは現在使用できません (Sorry, the page you are looking for is currently unavailable)

- 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：
 - 「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ます。
 - RESTCONF 機能が有効になっています (**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、ポートが設定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
...
switch-1#
```

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。

エフェメラル データ

RESTCONF のエフェメラル データについて

この機能は、エフェメラル データへのアクセスを提供します。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。

リクエスト内の 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。

RESTCONF のエフェメラル データの例

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
```

The server might respond:

```
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",
      ...
    }
  ]
}
```

実行操作

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 1: RESTCONF でモデル駆動型操作について

オペレーション	RESTCONF 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 <name> rollback running-config checkpoint <file>
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 <image> install {add アクティブ化 非アクティブ化 コミット remove}<rpm>
暗号化証明書のインポート	import_ca_certificate	crypto ca import <trustpoint> pkcs12 <file> <passphrase>
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 number>
ファイルのコピー	copy	コピー<destination><source>

RESTCONF 操作コマンドの例

チェックポイント RPC の例

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"
  }
}
```

The server might respond:
HTTP/1.1 204 No content

ロールバック RPC の例

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"
  }
}
```

The server might respond:
HTTP/1.1 204 No content

インストール RPC の例

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"
  }
}
```

The server might respond:
HTTP/1.1 204 No content

CA 証明書 RPC のインポートの例

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: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"
  }
}
```

The server might respond:
HTTP/1.1 204 No content

スイッチリロード RPC の例

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": {
    }
}
```

The server might respond:
HTTP/1.1 204 No content

モジュールリロード RPC の例

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"
    }
}
```

The server might respond:
HTTP/1.1 204 No content

コピー ファイル RPC の例

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"
    }
}
```

The server might respond:
HTTP/1.1 204 No content

RESTCONF エージェントのアカウントング 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ト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。

アカウントングログは、**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。

- 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"><l3LbRtdIf childAction="" descr="test" id="lo10"
rn="lb-
[lo10]" status="created,modified"/></interfaceEntity></topSystem>] (SUCCESS)
```

- 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/LbRtdIf-
list=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-NX-OS-device:System/intf-items/lb-items/LbRtdIf-
list=lo10],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-NX-OS-device:System/tm-items/certificate-
items],payload=[<trustpoint>test</trustpoint><hostname>foo</hostname><filename>
foo</filename>] (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。