

# Cisco UCS Director REST API スタートアップガイド、リリース 5.4

---

初版：2015 年 11 月 03 日

## Cisco UCS Director REST API の開始

### 概要

#### REST API を使用する理由

Cisco UCS Director REST API を使用して、アプリケーションは Cisco UCS Director とプログラマ的に対話できるようになります。これらの要求は、Cisco UCS Director のリソースへのアクセスを提供します。API コールを使用して、Cisco UCS Director ワークフローを実行し、スイッチ、アダプタ、ポリシー、およびその他のハードウェアおよびソフトウェア コンポーネントの設定に変更を加えることができます。

API は JavaScript Object Notation (JSON) または Extensible Markup Language (XML) のドキュメントを含む HTTP メッセージを受け入れて返します。HTTP メッセージに含まれる JSON または XML ペイロードは、Cisco UCS Director のメソッドまたは管理対象オブジェクト (MO) について説明します。プログラミング言語を使用して、メッセージや JSON または XML ペイロードを生成できます。

#### API の動作

RESTful API では、ユーザが実行するアクションを HTTP メソッドが指定し、ユーザがアクセスするリソースを URI が指定します。

REST API は次の HTTP メソッドを使用して、操作を作成、読み取り、更新、および削除 (CRUD) します。

| HTTP メソッド | 説明   |
|-----------|--|
| GET       | <p>指定されたリソースを取得します。GETは、エンジン状態を変えない読み取り専用操作です。</p> <ul style="list-style-type: none"> <li>• HTTP GET 操作には、要求本文を含めません。情報が GET 要求で渡される場合は、クエリー パラメータを代わりに使用する必要があります。</li> <li>• 指定がない場合、HTTP GET 操作では設定済みの状態が返されます。たとえば、グローバルルーティングテーブルの HTTP GET 操作は、ダイナミック ランタイム状態を返します。</li> </ul>   |
| POST      | <p>指定したリソースによって処理されるデータを送信します。処理されるデータは、要求本文に含まれています。POST操作では、新しいリソースを作成できます。</p> <ul style="list-style-type: none"> <li>• すべての POST 要求には、新しいリソースの定義を含む JSON 本文が含まれている必要があります。</li> <li>• POST 操作が新しいリソースを作成するには、HTTP 応答のロケーションヘッダーに後続の PUT、GET および DELETE コマンドに使用される完全な URL が含まれている必要があります。</li> <li>• 作成要求への HTTP POST 応答には、201 戻りコードと、HTTP ヘッダーで新しく作成されたリソースの URI を含むロケーションヘッダーが含まれている必要があります。</li> </ul> |
| PUT       | <p>新しい情報で指定したリソースを更新します。PUT操作に含まれているデータが前のデータに置き換わります。</p> <ul style="list-style-type: none"> <li>• PUT 操作は、新しいリソースを作成するために使用できません。</li> <li>• PUT 操作の要求本文には、リソースの必須属性の完全な表現を JSON 形式で含める必要があります。</li> </ul>   |

| HTTP メソッド | 説明   |
|-----------|--|
| DELETE    | <p>リソースを削除します。</p> <ul style="list-style-type: none"> <li>•すでに削除されているリソースを削除すると、404 Not Found 応答が返されます。</li> <li>•HTTP DELETE 操作には、要求本文を含めません。情報が GET 要求で渡される場合は、クエリーパラメータを代わりに使用する必要があります。</li> </ul> |

## REST API の使用方法

Cisco UCS Director を通じて REST API ブラウザにアクセスするには、有効な Cisco UCS Director ユーザアカウントと API アクセス キーが必要です。API アクセス キーは Cisco UCS Director による API 要求の認証に必要です。このアクセス キーは、特定の Cisco UCS Director ユーザアカウントに関連付けられた一意のセキュリティアクセス キー コードです。API アクセス キーを生成する方法の詳細については、[API アクセス キーの生成](#)、(5 ページ) を参照してください。

標準の HTTP 構文および意味規則に従い、*name:value* ヘッダーとして REST API アクセス キーを渡す必要があります。たとえば、有効な *name:value* のヘッダーの一例は、*X-Cloupia-Request-Key:F90ZZF12345678ZZ90Z12ZZ3456FZ789* です。API 要求ヘッダーの詳細については、[REST API 要求の作成方法](#)、(11 ページ) と [RFC2616](#) のヘッダー フィールド定義を参照してください。

REST API コールは、次のいずれかの方法で作成できます。

- Cisco UCS Director GUI : Cisco UCS Director には、開発者向けのレポートメタデータと REST API ブラウザを提供する開発者メニューオプションがあります。これらの機能にアクセスするには、開発者メニューを有効にします。開発者メニューを有効にする方法の詳細については、[開発者メニューオプションの有効化](#)、(7 ページ) を参照してください。

開発者メニューを有効にすると、次の機能にアクセスできるようになります。

- レポートメタデータ : レポートメタデータを使用すると、Cisco UCS Director に表示されたすべてのレポートの REST API URL を表示できます。レポートメタデータにアクセスする方法の詳細については、[レポートメタデータへのアクセス](#)、(9 ページ) を参照してください。
- REST API ブラウザ : REST API ブラウザは、Cisco UCS Director の [オーケストレーション (Orchestration)] メニューからアクセス可能です。REST API ブラウザは、API 情報および API コード生成機能を提供し、REST API と Java API の両方を含む利用可能なあらゆる API を見やすく表示し、作業を促進します。REST API ブラウザにアクセスする方法の詳細については、[REST API ブラウザの使用](#)、(8 ページ) を参照してください。

- REST クライアント : REST クライアントは、API 要求および応答を解析して表示するための便利なウィジェットです。このウィジェットでは、データ操作に REST URL を入力し、POST、PUT または DELETE などの HTTP メソッドをその URL に適用できます。REST クライアントでは、Cisco UCS Director サーバからデータを取得するための URL を入力するための単純なユーザ インターフェイスが提供されます。
  - Mozilla Firefox を使用している場合は、[Add-ons for Firefox](#) から RESTClient をダウンロードします。
  - Google Chrome を使用している場合は、[Chrome Web Store](#) から Advanced REST クライアントをダウンロードします。



(注) Cisco UCS Director にログインしている場合は、サポートされている Web ブラウザを使用して、API 要求を送信して応答を取得します。

- コード : [REST API ブラウザ (REST API Browser) ] タブの [サンプル Java コード (Sample Java Code) ] タブでは、管理サービスを自動化するために使用できるコード スニペットが提供されます。

[ソフトウェアダウンロード (software download) ] <https://software.cisco.com/download/navigator.html?mdfid=284775897> 領域または DevNet サイトからダウンロードした Cisco UCS Director SDK バイナリには、クライアント コードを実行するためのプラットフォームとして使用できるサンプルプロジェクトが含まれています。

## 前提条件

Cisco UCS Director SDK の使用を開始する前に、次のことを確認します。

- Cisco UCS Director インストールされ、システム上で動作している。Cisco UCS Director をインストールする方法の詳細については、『[Cisco UCS Director Installation Guide](#)』を参照してください。
- Cisco UCS Director SDK がシステムにインストールされている。
- Java 開発環境に Java バージョン 1.8 がある。



(注) リリース 5.3 以前の Cisco UCS Director の場合は、Java バージョン 1.6 が必要です。

- API アクセス キーがある。API アクセス キーを生成する方法の詳細については、[API アクセス キーの生成](#)、(5 ページ) を参照してください。
- RESTful Web サービスを実行する REST クライアントがある。



- (注)
- Mozilla Firefox を使用している場合は、[Add-ons for Firefox](#) から RESTClient をダウンロードします。
  - Google Chrome を使用している場合は、[Chrome Web Store](#) から Advanced REST クライアントをダウンロードします。

## API アクセス キーの生成

- ステップ 1** Cisco UCS Director で右上にあるログイン名をクリックします。  
たとえば、admin としてログインしている場合、Cisco UCS Director の右上に admin と表示されます。
- ステップ 2** [ユーザ情報 (User Information)] ダイアログボックスで、[詳細設定 (Advanced)] タブをクリックします。
- ステップ 3** [REST APIアクセスキー (REST API Access Key)] 領域に表示された値をコピーするには、[キー値のコピー (Copy Key Value)] をクリックします。
- ステップ 4** 安全な場所にアクセス キーを保存し、それを API 要求ヘッダーで使用します。  
API 要求ヘッダーの詳細については、[要求の形式](#) を参照してください。
- ステップ 5** API アクセス キーを変更する場合には、[キーの再生成 (Regenerate Key)] をクリックします。  
新しいキーの生成後は、古いキー コードは無効になり、使用できなくなります。

## Cisco UCS Director のダウンロードとインストール

### Cisco UCS Director REST API SDK バンドルについて

Cisco UCS Director REST API SDK バンドルは Cisco UCS Director REST API の一部です。Cookbook などのドキュメントに加え、SDK バンドルは REST API で使用できる例を提供します。これらの例には、テストケースや、SDK クラスの使用を例示するサンプルコードが含まれます。

SDK バンドルは `ucsd-rest-api-sdk-v2.zip` というアーカイブ ファイルの中にあります。

### Cisco UCS Director SDK のインストール

- ステップ 1** [ソフトウェアダウンロード (software download)] <https://software.cisco.com/download/navigator.html?mdfid=284775897> 領域または [DevNet](#) サイトから `ucsd-rest-api-sdk-x.x.0.0.zip` ファイルをダウンロードします。
- ステップ 2** ファイルを解凍します。

ucsd-rest-api-sdk-x.x.0.0 フォルダはローカル ドライブに作成されます。

このフォルダには、REST API でのアプリケーション開発を支援する次のフォルダとファイルが含まれています。

- javadocs : このフォルダには、*Cisco UCS Director REST API Javadocs* が含まれています。これらの Javadoc は SDK バンドルで提供されるすべてのクラスを網羅しています。
- lib : このフォルダには、ucsd-rest-api-sdk-v2.0.jar ファイルが含まれています。このファイルは、Cisco UCS Director の管理に必要なすべてのクラスおよびライブラリのアーカイブです。
- src : このフォルダには、テスト ケースとサンプル コードの例が含まれています。これらのテスト ケースとコードは、ucsd-rest-api-sdk-v2.jar ファイルでの SDK クラスの使用を例示します。サンプル コードは src\com\cisco\cuic\api\examples フォルダで使用可能です。
- .classpath : このファイルは、Java 仮想マシン (JVM) または Java コンパイラに SDK クラスおよびパッケージを探す場所を伝えます。
- .project : このファイルを使用すると、SDK バンドルを Eclipse IDE にインポートすることができます。このファイルは開発目的に必要なファイルです。これはアーカイブの最上位レベルで使用可能です。
- README.txt : このファイルには、SDK バンドルに含まれるすべてのコンポーネントのリストが含まれています。
- rest-server.properties : このファイルには src\com\cisco\cuic\api\examples フォルダの中の例によって使用されるプロパティが含まれています。

**ステップ 3** クライアントプログラムをコンパイルして実行するときに、ucsd-rest-api-sdk-v2.jar ファイルを Java クラスパスに追加します。Cisco UCS Director SDK を実行するには、Java バージョン 1.8 を使用する必要があります。

(注) Cisco UCS Director リリース 5.3 以前のリリースの場合は、Cisco UCS Director SDK を実行するには Java バージョン 1.6 を使用する必要があります。

## 非推奨の API のリスト

開発者には、非推奨の API を使用する代わりに新しい API に移行することを推奨しています。

表 1: 非推奨の API と新しい API

| 非推奨の API          | サポートされる Till | 新しい API  |
|-------------------|--------------|--|
| userAPICheckFunds | リリース 4.1     | Cisco UCS Director リリース 5.x 以降のリリースでは、userAPIGlobal フォルダから使用できる UserAPICheckFunds API は廃止されています。使用可能な顧客資金を調べるには、userAPICheckBack フォルダの UserAPICheckFunds API を使用します。 |

| 非推奨の API                             | サポートされる Till | 新しい API                       |
|--------------------------------------|--------------|-------------------------------|
| userAPISubmitServiceRequest          | リリース 4.1     | userAPIProvisionRequest       |
| userAPISubmitServiceRequestCustom    | リリース 4.1     | userAPIProvisionRequest       |
| UserAPIProvisionRequestWithPortGroup | リリース 4.1     | userAPIVMWareProvisionRequest |
| userAPIGetWorkflows                  | リリース 4.1     | userAPIGetWorkflowInputs      |
| userAPIImportWorkflows               | リリース 5.3     |                               |
| UserAPICreateAPICContainer           | リリース 5.4     |                               |

## 開発環境の設定

### GUI を介して REST API を使用する環境の設定

Cisco UCS Director の REST API ブラウザとレポート メタデータ情報にアクセスするには、開発者メニュー オプションを有効にします。REST API ブラウザおよびレポート メタデータ機能によって、サイト固有の API データが提供されます。

レポート メタデータ ビューから提供される HTTP 要求コードを使用して、API サービス結果をすぐに得ることができます。API 情報が必要なあらゆる状況でこれらのオプションを使用できます。

Cisco UCS Director REST API ブラウザは、API 情報および API コード生成機能を提供し、REST API と Java API の両方を含む利用可能なあらゆる API を見やすく表示し、作業を促進します。

GUI で REST API を使用するには、次のタスクを実行します。

- [開発者メニュー オプションの有効化](#), (7 ページ)
- [REST API ブラウザの使用](#), (8 ページ)
- [レポート メタデータへのアクセス](#), (9 ページ)

### 開発者メニュー オプションの有効化

#### はじめる前に

アプリケーションユーザに付与されるものとまったく同じデータへの管理アクセス権を提供する 1 つ以上のユーザ アカウントを取得します。異なる管理者およびエンドユーザのロールに関連付けられたデータ アクセスの制限については、Cisco UCS Director の管理者にお問い合わせください。異なるデータ アクセスとセキュリティ制御に関連付けられたユーザ体験をテストするために、複数のユーザ アカウントが必要な場合があります。

**ステップ 1** Cisco UCS Director で右上にあるログイン名をクリックします。

たとえば、admin としてログインしている場合、Cisco UCS Director の右上に admin と表示されます。

- ステップ 2** [ユーザ情報 (User Information)] ダイアログボックスで、[詳細設定 (Advanced)] タブをクリックします。
- ステップ 3** [[開発者]メニューの有効化(このセッション) (Enable Developer Menu (for this session))] チェックボックスをオンにします。  
現在のセッションの間中、[REST API ブラウザ (REST API Browser)] が [オーケストレーション (Orchestration)] メニューでアクティブになり、このセッションで開かれるレポート ビューで [レポート メタデータ (Report Metadata)] オプションが利用可能になります。
- ヒント** [詳細 (Advanced)] タブにはアカウントの REST API アクセス キー コードも表示されません。
- ステップ 4** [ユーザ情報 (User Information)] ダイアログボックスを選択します。

## REST API ブラウザの使用

Cisco UCS Director REST API ブラウザは、XML 形式の REST API および Java API を含むすべての利用可能な Cisco UCS Director API の使用において開発者に支援と教育を提供する、API 情報および API コード生成機能を提供します。プライマリ ビューには、API が含まれているタスク フォルダがリストされます。タスク名は、API がリストされるカテゴリを提供します。たとえば、NetApp ONTAP タスクおよび NetApp OnCommand タスクに関連するすべての API は、これらの名前が付いたフォルダの中にあります。

### はじめる前に

- アプリケーションユーザに付与されるものとまったく同じデータへの管理アクセス権を提供する 1 つ以上のユーザ アカウントを取得します。異なる管理者およびエンドユーザのロールに関連付けられたデータ アクセスの制限については、Cisco UCS Director の管理者にお問い合わせください。異なるデータアクセスとセキュリティ制御に関連付けられたユーザ体験をテストするために、複数のユーザ アカウントが必要な場合があります。
- セッションの開発者メニュー オプションを有効にします。

- ステップ 1** メニューバーで、[ポリシー (Policies)] > [オーケストレーション (Orchestration)] の順に選択します。
- ステップ 2** [REST API ブラウザ (REST API Browser)] タブをクリックします。  
必要に応じて、右のスクロール矢印をクリックし、[REST API ブラウザ (REST API Browser)] タブに移動します。
- ステップ 3** 表示する API を含むタスク フォルダを開きます。  
**ヒント** 特定の API がどのタスク フォルダに属するかが分からない場合には、[REST API ブラウザ (REST API Browser)] タブの右上隅にある [検索 (Search)] フィールドを使用して検索できます。API の [リソース (Resource)]、[操作 (Operation)]、または [説明 (Description)] フィールドに含まれる文字列を入力して検索を絞り込みます。また、[高度な検索フィルタを追加 (Add Advanced Filter)] など、メニュー バーにある他のオプションを使用して特定の API を検索することもできます。



**ステップ 4** 必要な API リソースと操作を含む行をダブルクリックします。  
REST API ブラウザには次が表示されます。

- [APIの例 (API Examples) ] タブ : 選択された API データが表示され、サンプル URL を生成することができます。選択された操作とリソースに基づいて、このタブにはパラメータ値を受け入れるデータ入力ボックスも含まれる場合があります。データ入力ボックスで利用可能な場合、[選択 (Select) ] をクリックすると、データ検索フィルタが開き、入力する必要があるデータを絞り込み、選択することができます。
- [詳細 (Details) ] タブ : API の定義、入力パラメータ、出力パラメータを含む API についての追加情報が表示されます。
- [サンプルJavaコード (Sample Java Code) ] タブ : API のサンプル コードが表示されます。

#### レポート メタデータへのアクセス

レポート メタデータを使用すると、Cisco UCS Director で表示される各レポートの API 要求コードを含む、Cisco UCS Director によって使用される API コードを確認することができます。このコードには、すぐにブラウザに貼り付けて URL 要求を Cisco UCS Director に送信できる完全な URL が含まれています。直接の API 応答は開発者に多くの情報を提供します。API 要求コードを表示するには、レポートに移動し、[レポートメタデータ (Report Metadata) ] を選択します。

#### はじめる前に

- アプリケーションユーザに付与されるものとまったく同じデータへの管理アクセス権を提供する 1 つ以上のユーザアカウントを取得します。異なる管理者およびエンドユーザのロールに関連付けられたデータ アクセスの制限については、Cisco UCS Director の管理者にお問い合わせください。異なるデータアクセスとセキュリティ制御に関連付けられたユーザ体験をテストするために、複数のユーザアカウントが必要な場合があります。
- セッションの開発者メニュー オプションを有効にします。

**ステップ 1** Cisco UCS Director で API コードを確認するページに移動します。  
たとえば、次のいずれかを選択します。

- [ポリシー (Policies) ] > [オーケストレーション (Orchestration) ]
- [物理 (Physical) ] > [ストレージ (Storage) ] > [Storage\_Account] > [ファイラ (Filers) ]

**ステップ 2** [レポートメタデータ (Report Metadata) ] をクリックします。

**ステップ 3** [情報 (Information) ] ダイアログ ボックスで、サンプル コードを確認します。

## REST クライアントを介して REST API を使用する環境の設定

REST クライアントによって、便利かつ有益な方法で API データの解析とラベル付けを行うスタンドアロンのクライアントが提供されます。また、サポートされている Web ブラウザを使用して、API 要求を送信して応答を取得することもできます。REST URL を実行するには、Web ブラウザでサポート対象の REST クライアントをダウンロードします。

- Mozilla Firefox では、[Add-ons for Firefox](#) から RESTClient をダウンロードします。
- Google Chrome では、[Chrome Web Store](#) から Advanced REST クライアントをダウンロードします。

REST クライアントを使用して、JSON API および XML API を実行できます。

以下は、GET メソッドを使用してすべてのカタログを取得するためのサンプルの JSON ベースの API URL です。

```
http://<serverip>/ app/api/rest?formatType=json&opName=userAPIGetAllCatalogs&opData={}
```

以下は、POST メソッドを使用してユーザを作成するためのサンプルの XML ベースの API URL です。

```
http://serverip/cloupia/api-v2/user
<cuicOperationRequest>
<payload>

<AddUserConfig>
<userType>Admin</userType>
<!-- Accepts value from the list: userGroupByType-->
<userGroup>1</userGroup>
<mspOrganization></mspOrganization>
<loginName>apiuser</loginName>
<!-- Accepts value from the list: password-->
<password>bGthbmRhc2E=</password>
<!-- Accepts value from the list: password-->
<confirmPassword>bGthbmRhc2E=</confirmPassword>
<userContactEmail>apiuser@cisco.com</userContactEmail>
<firstName>API</firstName>
<lastName>User</lastName>
<phone></phone>
<address></address>
</AddUserConfig>

</payload>
</cuicOperationRequest>
```

以下は、ユーザ作成 URL の HTTP 応答です。

```
HTTP Response Code : 200
Response data :
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><cuicOperationResponse>
<Log><Messages>1</Messages><Message><TimeStamp>2015-08-28 06:55:17.240</TimeStamp>
<Severity>INFO</Severity><Text>User Admin Created Successfully</Text>
</Message></Log><operationStatus>0</operationStatus><response><AddUserConfigResponse>
<OUTPUT_USER_NAME>apiuser</OUTPUT_USER_NAME></AddUserConfigResponse></response>
<responseMap><entry><key>OUTPUT_USER_NAME</key><value>apiuser</value></entry>
</responseMap></cuicOperationResponse>
```

## コードを介して REST API を使用する環境の設定

Cisco UCS Director SDK バンドルを Java プロジェクトとして Eclipse IDE にインポートして、Eclipse でコードを実行します。Cisco UCS Director SDK バンドルをインポートする方法の詳細について

は、[Eclipse IDE への SDK バンドルプロジェクトのインポート](#)、(11 ページ) を参照してください。コードは、[REST APIブラウザ (REST API Browser)] の [サンプルJavaコード (Sample Java Code)] タブから取得できます。REST API ブラウザにアクセスする方法の詳細については、[REST API ブラウザの使用](#)、(8 ページ) を参照してください。



- (注) Eclipse で Cisco UCS Director SDK バンドルを使用する手順については、このドキュメントで説明しています。同様のプロセスを他の IDE に適用できますが、開発者はそれらの詳細を把握する必要があります。

## Eclipse IDE への SDK バンドルプロジェクトのインポート

### はじめる前に

SDK バンドルアーカイブを取得し、そのコンテンツを適切なフォルダに抽出します。

- ステップ 1 Eclipse IDE で、[ファイル (File)] > [新規 (New)] > [Javaプロジェクト (Java Project)] を選択します。  
[Javaプロジェクトの作成 (Create a Java Project)] ダイアログボックスが表示されます。
- ステップ 2 [プロジェクト名 (Project Name)] フィールドにプロジェクトの名前を入力します。
- ステップ 3 プロジェクトを右クリックして、[インポート (Import)] を選択します。  
[インポート (Import)] ダイアログボックスが表示されます。
- ステップ 4 [ファイルシステム (File System)] を選択して、[次へ (Next)] をクリックします。
- ステップ 5 [参照 (Browse)] をクリックして、SDK バンドルを抽出したフォルダに移動します。
- ステップ 6 [終了 (Finish)] をクリックします。  
Eclipse IDE の [プロジェクトエクスプローラ (Project Explorer)] タブに SDK バンドルプロジェクトが表示されます。

## REST API 要求の作成方法

API クライアントは HTTP 要求を使用して Cisco UCS Director と対話します。REST API アクセスキーをパスするには、各要求が X-Cloupia-Request-Key と呼ばれる http ヘッダーに関連付けられており、その値が現在の REST API アクセス キーに設定されている必要があります。REST API アクセス キーを生成する方法については、[API アクセス キーの生成](#)、(5 ページ) を参照してください。

API に対する要求には次の特性があります。

- 要求が HTTP で送信されます。
- 要求には、次の形式のいずれかで有効な URL が含まれている必要があります。

- **JSON 形式**

```
http://SERVER/app/api/rest?formatType=json&opName=operationName&opData=operationData
```

## 引数の説明

- **SERVER** : Cisco UCS Director VM の IP アドレスまたはホスト名。
- **formatType** : ここで説明するサポートされる形式は JavaScript Object Notation (JSON) のみです。このパラメータ値を `json` に設定します。
- **opName** : 要求に関連付けられた API 操作の名前です。例 :  
`userAPIGetMyLoginProfile` または `userAPIGetVMActionStatus`。
- **opData** : 操作に関連付けられたパラメータ (または引数) です。Cisco UCS Director はパラメータの JSON エンコーディングを使用します。操作に引数が必要ではない場合、空のセットとして `{}` を使用します。要求で JSON データを送信する前に、必要に応じてエスケープ文字を適用して URL をエンコードする必要があります。URL エンコーディングの詳細については、<http://www.ietf.org/rfc/rfc1738.txt> の RFC を参照してください。JSON 構文とデータタイプの詳細については、[http://en.wikipedia.org/wiki/JSON#Data\\_types.2C\\_syntax\\_and\\_example](http://en.wikipedia.org/wiki/JSON#Data_types.2C_syntax_and_example) を参照してください。JSON 以外の形式の API 要求については、[REST API ブラウザの使用](#)、(8 ページ) を参照してください。

### • XML 形式

`http://server/cloupia/api-v2/group`

HTTP メソッド : POST

```
<cuicOperationRequest>
<payload>

<AddGroupConfig>
<groupName>TestGroup</groupName>
<groupDescription></groupDescription>
<parentGroup>0</parentGroup>
<groupCode></groupCode>
<groupContact>jbesai@cisco.com</groupContact>
<firstName></firstName>
<lastName></lastName>
<phone></phone>
<address></address>
<groupSharePolicyId></groupSharePolicyId>
<allowPrivateUsers>>false</allowPrivateUsers>
</AddGroupConfig>

</payload>
</cuicOperationRequest>
```

REST API の例については、『[Cisco UCS Director REST API Cookbook](#)』を参照してください。

## 操作のデータ パラメータまたは引数の概要

メソッドと API リソース タイプは `opName` を通じて伝えられるため、操作のパラメータは操作を行うリソースの特定のインスタンスを指定するために必要なあらゆる引数を提示する必要があります。

## 操作データ パラメータ構文

次の表に JSON 形式の操作データ パラメータ構文の例を示します。

| 操作に次のパラメータが必要か否か (opData) | JSON で表す方法                                   |
|---------------------------|--|
| パラメータなし                   | {}   |
| 1つのパラメータ：整数（例：10）         | {param0:10}                                  |
| 1つのパラメータ：文字列（例：cloud）     | {param0:"cloud"}                             |
| 2つのパラメータ：文字列と整数           | {param0:"cloud",param1:10}                   |
| 2つのパラメータ：null 値の文字列と整数    | {param0:null,param1:10}                      |
| 3つのパラメータ                  | {param0:"cloud",param1:"cloupia",param2:100} |

### 操作データ パラメータの例

```
...&opData={param0:"datacenter",param1:"DataCenter1",param2:"STORAGE-ACCOUNTS-T51"}
...&opData={param0:"Create NFS Datastore",param1:{"list":[{"name":"Volume Size","value":100},
{"name":"Select Group","value":"14"},{"name":"Select vDC","value":"18"}]},param2:212}
```

- param0：REST API 経由で呼び出されるワークフローの名前。
- param1：ワークフローに渡される入力。複数の入力がある場合は、入力をカンマで区切り、入力の名前と値を引用符で囲みます。入力がない場合は、パラメータの値としてキーワード NULL を使用します。
- param2：このワークフローが他のサービス要求の子ワークフローとして呼び出される場合は、サービス要求 (SR) ID を使用します。このワークフローが子ワークフローとして呼び出されない場合は、-1 を使用します。-1 を使用した場合は、新しいサービス要求が作成されます。



#### ヒント

Cisco UCS Director は、URL 形式で切り取ってブラウザに貼り付けることができる、多くの完全な API 要求を提供します。開発者メニュー オプションの有効化、(7 ページ) を参照してください。

### コンテキストパラメータ

上の例で、param0 は Cisco UCS Director のコンテキストを指定するために使用されます。コンテキスト データ値は Cisco UCS Director によって管理される主要なドメイン

("global-services"、"datacenter"、"storage\_accounts" など) の 1 つを参照します。標準の Cisco UCS Director コンテキストのリストは [API 要求コンテキストパラメータ](#) に記載されています。

### レポートパラメータ

レポートパラメータ値は常に *reportId* です。一般的な reportId に

は、"STORAGE-ACCOUNTS-T51"、"CPU-S0"、"VOLUMES--X1"、"NETWORK-USAGE-H0"、"PORT-SUMMARY-V50"、

および "PRIVATE-CLOUD-FREE-STORAGE-S1" が含まれます。通常、reportId は、レポートの API 要求のパラメータリストの最後のパラメータです。よって、コンテキストが2つのパラメータで指定されている場合には、通常レポートパラメータは3つ目の param2 です。コンテキスト別にカテゴリ分けされた、レポート名および reportId の包括的なリストについては、[利用可能な Cisco UCS Director レポートのリスト](#) を参照してください。

### サンプル API 要求 1

ログインプロファイルを要求する操作はログインしているユーザを参照するため、パラメータは必要ありません。ほとんどのその他の操作は複数の引数を必要とします。

```
http://10.10.1.153/app/api/rest?formatType=json&opName=userAPIGetMyLoginProfile&opData={}
```

### サンプル API 要求 1 に対する応答

```
{ "serviceResult": {"userId": "jsmith", "firstName": "John", "lastName": "Smith", "email": "jsmith@example.com", "groupName": "Eng Group", "role": "Regular"}, "serviceError": null, "serviceName": "InfraMgr", "opName": "userAPIGetMyLoginProfile" }
```

### サンプル API 要求 2

データセンターのシャーシに関するレポートのこの要求では、操作に3つのパラメータが必要であり、これはレポートの要求では一般的です。

```
http://172.99.999.142/app/api/rest?opName=userAPIGetTabularReport&opData={param0:"datacenter",param1:"datacenter",param2:"UCS-CHASSIS-T50"}
```

### サンプル API 要求 2 に対する応答

```
{"serviceResult":
  {"rows": [{"ID": "PHY-ACC;sys/chassis-2",
    "Account_Name": "PHY-ACC", "DN": "sys/chassis-2",
    "Serial_Number": "1558", "Model": "N20-C6508", "Power_State": "ok",
    "Operation_State": "accessibility-problem",
    "Configuration_State": "ok", "License_State": "license-ok", "Servers": 5,
    "IO_Modules": 2, "PSUs": 4, "Fan_Modules": 8,
    "Vendor": "Cisco Systems Inc"}, {"ID": "UCSCirrus;sys/chassis-1",
    "Account_Name": "UCSCirrus", "DN": "sys/chassis-1",
    "Serial_Number": "FOX1352GDX4", "Model": "N20-C6508",
    "Power_State": "redundancy-failed", "Operation_State": "power-problem",
    "Configuration_State": "ok", "License_State": "license-ok", "Servers": 7,
    "IO_Modules": 2, "PSUs": 4, "Fan_Modules": 8,
    "Vendor": "Cisco Systems Inc"}, {"ID": "UCSM237;sys/chassis-1",
    "Account_Name": "UCSM237", "DN": "sys/chassis-1",
    "Serial_Number": "1557", "Model": "N20-C6508", "Power_State": "ok",
    "Operation_State": "operable",
    "Configuration_State": "unsupported-connectivity", "License_State": "license-ok",
    "Servers": 6, "IO_Modules": 2,
    "PSUs": 4, "Fan_Modules": 8, "Vendor": "Cisco Systems Inc"},
    {"ID": "real108;sys/chassis-1", "Account_Name": "real108",
    "DN": "sys/chassis-1", "Serial_Number": "FOX1352GDX4", "Model": "N20-C6508",
    "Power_State": "redundancy-failed",
    "Operation_State": "power-problem", "Configuration_State": "ok",
    "License_State": "license-ok", "Servers": 7,
    "IO_Modules": 2, "PSUs": 4, "Fan_Modules": 8, "Vendor": "Cisco Systems Inc"}},
  "columnMetaData": null, "serviceError": null, "serviceName": "InfraMgr",
  "opName": "userAPIGetTabularReport" }
```



## ヒント

上級 Cisco UCS Director API ユーザ向け：データセンターの UCS-CHASSIS-T50 レポートのレポートメタデータ要求内に含まれるコードでは、最初のパラメータが `param0:"23"` になっています。サンプル API 要求 2 で使用された要求では、コンテキスト "datacenter" が値 "23" で置き換えられ、要求は成功しました。

## HTTP 応答の解釈方法

次の HTTP ステータス コードが Cisco UCS Director より返されます。

- 401 Unauthorized : API キーは無効です。
- 200 OK : Cisco UCS Director は要求を処理しました。要求の実際のステータスは応答の本文に含まれます。

Cisco UCS Director の応答の本文は JSON 形式です。これは API 要求で指定された `FormatType` パラメータにより決定されます。

## API 応答のコンポーネント

| API 応答のコンポーネント | 説明  | コンポーネント例 (成功のシナリオ)   |
|----------------|---|--|
| serviceResult  | 要求が成功すると、この結果には名前/値ペアのセットまたは JSON オブジェクト、あるいはレポートが含まれます。  | <pre>"serviceResult": {   "userId": "jsmith",   "firstName": "John", "lastName": "Smith",   "email": "jsmith@example.com",   "groupName": "Eng Group", "role": "Regular" }</pre> |
| serviceError   | 要求が成功すると、 <code>serviceError</code> が <code>null</code> に設定されます。操作が失敗すると、 <code>serviceError</code> にはエラーメッセージが含まれます。 | <pre>"serviceError": null</pre>  |
| serviceName    | バックエンドサービスの名前。たとえば、往々にして <i>InfraMgr</i> に設定されています。   | <pre>"serviceName": "InfraMgr"</pre>   |
| opName         | 要求で提供された操作の名前です。  | <pre>"opName": "userAPIGetMyLoginProfile"</pre>  |

**例：成功シナリオでの API 応答**

```
{
  "serviceResult":{"userId":"jsmith","firstName":"John","lastName":"Smith",
  "email":"jsmith@example.com","groupName":"Eng Group","role":"Regular"}, "serviceError":null,
  "serviceName":"InfraMgr",
  "opName":"userAPIGetMyLoginProfile"
}
```

**例：失敗シナリオでの API 応答**

```
{ "serviceResponse":null,
  "serviceError":"SERVICE_CALL_EXCEPTION: Service InfraMgr does not support operation test",
  "serviceName":"InfraMgr", "opName":"test"
}
```

**API 応答（サービス結果）データ タイプ**

Cisco UCS Director REST API 要求への応答で送信されるサービス結果（ペイロード）はその操作向けに指定されています。サービス結果は操作に特化した名前/値ペアのセット、またはこの API の標準データ タイプ、つまりレポートまたは JSON オブジェクトとしてフォーマットされている場合があります。

**Cisco UCS Director REST API SDK バンドル - サンプル**

SDK をダウンロードすると、サンプルの Java コードを `ucsd-rest-api-sdk-v2/src/com/cisco/cuic/api/examples` パスで検索できます。

**使用例：ログイン ユーザに基づくグループ内の VDC の取得**

VM をプロビジョニングするときは、特定のグループで使用できる VDC のリストを表示し、必要な VDC を選択する必要があります。これを行うには、Cisco UCS Director は、ユーザが属するグループを特定し、そのグループで使用できる VDC のリストを取得する必要があります。その後、VM がプロビジョニングされる VDC を作成できます。

次の REST API を順に使用してワークフローを作成する必要があります。

- 1 `userAPIGetMyLoginProfile` : ユーザが属するグループを特定するためにログイン ユーザのプロファイルを取得します。
- 2 `userAPIGetAllVDCs` : ユーザグループで使用できる VDC のリストを取得します。
- 3 `userAPICreateVDC` : VM をプロビジョニングする VDC を選択するためのサービス要求を作成します。

---

**ステップ 1** `userAPIGetMyLoginProfile` API を使用して、ユーザが属するグループを特定します。

要求 :

```
/app/api/rest?formatType=json&opName=userAPIGetMyLoginProfile&opData={}
```



回答：

```
{ "serviceResult": {"userId": "sdk", "firstName": null, "lastName": null, "email": "xyz@test.com",
"groupName": "Default Group", "groupId": 1, "role": "Regular"}, "serviceError": null,
"serviceName": "InfraMgr", "opName": "userAPIGetMyLoginProfile"}
```

この例では、ユーザの sdk は DefaultGroup グループに属しています。

**ステップ 2** userAPIGetAllVDCs API を使用してユーザ グループ内の VDC のリストを取得します。

要求：

```
/app/api/rest?formatType=json&opName=userAPIGetAllVDCs&opData={}
```

回答：

```
{ "serviceResult": {"rows": [{"Cloud": "", "vDC_ID": 1, "Group": "Default Group", "vDC": "Default
vDC", "Type": "Standard", "Lock_State": "Locked", "Total_VMs": 82, "Active_VMs": 72, "Custom_Categories": 0,
"vDC_Description": "", "Approvers": "", "Status": "Default vDC: Not for new provisioning", "Tag": null},
{"Cloud": "VCenter", "vDC_ID": 4, "Group": "Default Group", "vDC": "SDKcont1", "Type": "Container",
"Lock_State": "Unlocked", "Total_VMs": 2, "Active_VMs": 2, "Custom_Categories": 0, "vDC_Description": "",
"Approvers": "", "Status": "Error(s):\nNo NIC configuration found.", "Tag": null}], "columnMetaData":
null, "reportParams": {}}, "serviceError": null, "serviceName": "InfraMgr", "opName": "userAPIGetAllVDCs"
}
```

**ステップ 3** userAPICreateVDC API を使用して、VM をプロビジョニングする VDC を選択するためのサービス要求を作成します。承認者がサービス要求を承認し、ユーザにはこの VDC を使用した VM プロビジョニングについて通知されます。システム ポリシー、コンピューティング ポリシー、ネットワーク ポリシー、ストレージポリシー、および VDC に適用できるコストモデルが選択されます。非アクティブな VM を削除するまで待機する日数が選択されます。VDC で実行できるタスクまたはアクションを定義するセルフサービス ポリシーが選択されます。

要求：

```
/app/api/rest?formatType=json&opName=userAPICreateVDC&opData={param0: {"vdcName": "sdk",
"vdcDescription": "sdk vdc", "cloudName": "VCenter", "groupName": 1, "approver1": "", "approver2": "",
"vdcSupportEmail": "", "vdcCustomerNotificationEmail": "", "systemPolicy": "Default System Policy",
"deploymentPolicy": "", "slaPolicy": "", "computingPolicy": "VCenter - Default Computing Policy",
"storagePolicy": "VCenter - Default Storage Policy", "networkPolicy": "VCenter - Default Network
Policy", "costModel": "", "isLocked": false, "isDeletable": true, "inactivityPeriodForDeletion": -1,
"selfServiceEndUserPolicy": ""}}
```

回答：

```
{ "serviceResult": true, "serviceError": null, "serviceName": "InfraMgr", "opName": "userAPICreateVDC"
}
```

■ 使用例 : ログイン ユーザに基づくグループ内の VDC の取得

---

このマニュアルに記載されている仕様および製品に関する情報は、予告なしに変更されることがあります。このマニュアルに記載されている表現、情報、および推奨事項は、すべて正確であると考えていますが、明示的であれ黙示的であれ、一切の保証の責任を負わないものとします。このマニュアルに記載されている製品の使用は、すべてユーザ側の責任になります。

対象製品のソフトウェア ライセンスおよび限定保証は、製品に添付された『Information Packet』に記載されています。添付されていない場合には、代理店にご連絡ください。

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

ここに記載されている他のいかなる保証にもよらず、各社のすべてのマニュアルおよびソフトウェアは、障害も含めて「現状のまま」として提供されます。シスコおよびこれら各社は、商品性の保証、特定目的への準拠の保証、および権利を侵害しないことに関する保証、あるいは取引過程、使用、取引慣行によって発生する保証をはじめとする、明示されたまたは黙示された一切の保証の責任を負わないものとします。

いかなる場合においても、シスコおよびその供給者は、このマニュアルの使用または使用できないことによって発生する利益の損失やデータの損傷をはじめとする、間接的、派生的、偶発的、あるいは特殊な損害について、あらゆる可能性がシスコまたはその供給者に知らされていても、それらに対する責任を一切負わないものとします。

このマニュアルで使用している IP アドレスおよび電話番号は、実際のアドレスおよび電話番号を示すものではありません。マニュアル内の例、コマンド出力、ネットワーク トポロジ図、およびその他の図は、説明のみを目的として使用されています。説明の中に実際のアドレスおよび電話番号が使用されていたとしても、それは意図的なものではなく、偶然の一致によるものです。

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <http://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

© 2015 Cisco Systems, Inc. All rights reserved.