Cisco IMC Supervisor REST API の手順書、リリース 2.0

初版：2016 年 03 月 23 日

シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

【注意】シスコ製品をご使用になる前に、安全上の注意（**www.cisco.com/jp/go/safety_warning/**）をご確認ください。本書は、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。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

目 次

はじめに

ここでは、次の項について説明します。

対象読者

このマニュアルは主に、Cisco IMC Supervisor を使用し、サーバ管理に関する専門知識を持っているデータセンターの管理者向けに準備されています。

表記法

テキストのタイプ	説明
GUI 要素	タブの見出し、領域名、フィールドのラベルのようなGUI要素は、[GUI要素] のように示しています。 ウィンドウ、ダイアログボックス、ウィザードのタイトルのようなメイン タイトルは、[メイン タイトル] のように示しています。
マニュアルのタイトル	マニュアルのタイトルは、イタリック体（*italic*）で示しています。
TUI 要素	テキストベースのユーザ インターフェイスでは、システムによって表示されるテキストは、`courier` フォントで示しています。

テキストのタイプ	説明
システム出力	システムが表示するターミナル セッションおよび情報は、courier フォントで示しています。
CLI コマンド	CLI コマンドのキーワードは、ボールド体（**bold**）で示しています。 CLI コマンド内の変数は、イタリック体（*italic*）で示しています。
[ ]	角カッコの中の要素は、省略可能です。
{x | y | z}	どれか1つを選択しなければならない必須キーワードは、波カッコで囲み、縦棒で区切って示しています。
[x | y | z]	どれか1つを選択できる省略可能なキーワードは、角カッコで囲み、縦棒で区切って示しています。
string	引用符を付けない一組の文字。stringの前後には引用符を使用しません。引用符を使用すると、その引用符も含めて string とみなされます。
< >	パスワードのように出力されない文字は、山カッコで囲んで示しています。
[ ]	システム プロンプトに対するデフォルトの応答は、角カッコで囲んで示しています。
!、#	コードの先頭に感嘆符（!）またはポンド記号（#）がある場合には、コメント行であることを示します。

（注） 「注釈」です。役立つ情報や、このマニュアル以外の参照資料などを紹介しています。

注意 「要注意」の意味です。機器の損傷またはデータ損失を予防するための注意事項が記述されています。

ヒント 「問題解決に役立つ情報」です。ヒントには、トラブルシューティングや操作方法ではなく、ワンポイントアドバイスと同様に知っておくと役立つ情報が記述される場合もあります。

ワンポイント アドバイス 「時間の節約に役立つ操作」です。ここに紹介している方法で作業を行うと、時間を短縮できます。

警告　安全上の重要事項

「危険」の意味です。人身事故を予防するための注意事項が記述されています。機器の取り扱い作業を行うときは、電気回路の危険性に注意し、一般的な事故防止対策に留意してください。警告の各国語版については、各警告文の末尾に提示されている番号をもとに、この機器に付属している各国語で記述された安全上の警告を参照してください。

これらの注意事項を保存しておいてください

マニュアルに関するフィードバック

このマニュアルに関する技術的なフィードバック、または誤りや記載もれなどお気づきの点がございましたら、HTML ドキュメント内のフィードバック フォームよりご連絡ください。ご協力をよろしくお願いいたします。

マニュアルの入手方法およびテクニカル サポート

マニュアルの入手方法、テクニカル サポート、その他の有用な情報について、毎月更新される『What's New in Cisco Product Documentation』を参照してください。シスコの新規および改訂版の技術マニュアルの一覧も示されています。

『What's New in Cisco Product Documentation』は RSS フィードとして購読できます。また、リーダー アプリケーションを使用してコンテンツがデスクトップに直接配信されるように設定することもできます。RSS フィードは無料のサービスです。シスコは現在、RSS バージョン 2.0 をサポートしています。

関連資料

Cisco IMC Supervisor ドキュメント セット

以下に、Cisco IMC Supervisor で利用可能なドキュメントを示します。

- 『Cisco IMC Supervisor Release Notes』
- 『Cisco IMC Supervisor Installation and Upgrade on VMware Vsphere Guide』
- 『Cisco IMC Supervisor Rack-Mount Servers Management Guide』
- 『Cisco IMC Supervisor Shell Guide』
- 『Cisco IMC Supervisor REST API Getting Started Guide』
- 『Cisco IMC Supervisor REST API Cook Book』

その他の資料

すべての C シリーズ マニュアルの一覧については、http://www.cisco.com/go/unifiedcomputing/c-series-doc で入手できる『*Cisco UCS C-Series Servers Documentation Roadmap*』を参照してください。

（注） 『*Cisco UCS C-Series Servers Documentation Roadmap*』には Cisco Integrated Management Controller のドキュメントへのリンクが含まれています。

第 1 章

概要

この章の内容は、次のとおりです。

例の構造

記述表題の下で、例はそれぞれ次のセクションから構成されます。

目的

例を使用するときの目的。

前提条件

例を機能させるために必要な条件。

REST URL

REST API を渡すための REST URL。

コンポーネント

例で使用されるオブジェクトとメソッド、表示される入力変数。

入力 XML の例

入力コードのサンプル。

実装

例を実行するときは、変更内容を含める必要がある場合があることに注意してください。

関連項目

関連する例

例の使用方法

このマニュアルは、Cisco IMC Supervisor で使用するサーバ側のスクリプト ソリューションである REST API を使用する際に役立つ例のコレクションです。手順書と同様に、少なくとも 3 つの方法でこのマニュアルを使用できます。

- 記載されているとおりに例に従うことで（もちろん使用する変数に置き換えて）、従う手順のすべてを把握していなくてもタスクを完了することができます。
- テンプレートとして例を使用し、ワーク内の同様のタスクに適合させることができます。
- 例を検討することで、REST API で「どのようなことが行われるか」について把握でき、スクリプトを作成する必要があるその他のタスクでの異なるメソッドの使用について概要を把握できます。

例は、一般的な使用例を説明するために選択されています。その目的はこれらの 3 つのすべての方法で円滑に使用できるようにすることです。

（注） API は、HTTP POST と GET のどちらかを使用します。次の例では、すべての READ API は GET で、それ以外は POST です。

第 2 章

例

この章の内容は、次のとおりです。

ファームウェアの管理

概要

このカテゴリの例としては、Cisco IMC Supervisor のさまざまなファームウェア管理タスクがあります。ネットワークの場所におけるファームウェアイメージの管理、それらのイメージの cisco.com からのダウンロード、サーバにおけるファームウェア アップグレード操作などが含まれます。

ファームウェア ネットワーク イメージの作成

目的

ネットワークの場所でファームウェア イメージを作成します。

前提条件

HUU イメージは、（NFS/CIFS/HTTP を使用して）ネットワークの場所で使用可能でなければなりません。

REST URL

```
/cloupia/api-v2/CreateNetworkImage
```

コンポーネント

NETWORK_IMAGE_CREATE API のパラメータは次のとおりです。

- 文字列 profileName：プロファイルの一意の名前。
- 文字列 platform：プラットフォームの名前。
- 文字列 networkServerType：ネットワーク ファイル システム（NFS）、Common Internet File System（CIFS）または HTTP/S サーバ タイプを選択します。
- 文字列 locationLink：イメージの場所の有効な HTTP/HTTPS URL リンク。
- 文字列 networkPath：ネットワーク パス。
- 文字列 sharePath：ネットワーク共有パス。
- 文字列 remoteFileName：リモート ファイル名。
- 文字列 nwPathUserName：オプション。ネットワーク パスのユーザ名。
- 文字列 nwPathPassword：オプション。ネットワーク パスのパスワード。
- 文字列 mountOptions：オプション。有効なマウント オプション。

入力 XML の例

```
<cuicOperationRequest>
<operationType>NETWORK_IMAGE_CREATE</operationType>
<payload>
<![CDATA[
<CreateNetworkImage>
<profileName></profileName>

<platform></platform>

<networkServerType>NFS</networkServerType>

   <!-- Set this value only when networkServerType equals to HTTP  -->
<locationLink></locationLink>

   <!-- Set this value only when networkServerType not equals to HTTP  -->
<networkPath></networkPath>

   <!-- Set this value only when networkServerType not equals to HTTP  -->
<sharePath></sharePath>

   <!-- Set this value only when networkServerType not equals to HTTP  -->
<remoteFileName></remoteFileName>

<nwPathUserName></nwPathUserName>

<nwPathPassword></nwPathPassword>

   <!-- Set this value only when networkServerType equals to CIFS  -->
<mountOptions></mountOptions>

</CreateNetworkImage>

]]>
</payload>
</cuicOperationRequest>
```

実装

プロファイル名は必須で、一意にする必要があります。プラットフォーム、サーバ タイプ（NFS/CIFS/HTTP）は必須です。NFS/CIFS の場合、リモート IP、リモート共有、リモート ファイル名が必須です。HTTP の場所は、システムから到達できなければなりません。

関連項目

ファームウェア ネットワーク イメージの更新, （5 ページ）

ファームウェア イメージ プロファイルの削除, （11 ページ）

ファームウェア ネットワーク イメージの更新

目的

ネットワークの場所のファームウェア イメージを更新します。

前提条件

HUU イメージは、（NFS/CIFS/HTTP を使用して）ネットワークの場所で使用可能でなければなりません。

REST URL

```
/cloupia/api-v2/UpdateNetworkImage
```

コンポーネント

NETWORK_IMAGE_UPDATE API のパラメータは次のとおりです。

- 文字列 imageId：イメージの一意の ID。
- ブール値 platform：サーバを管理するプラットフォーム。
- 文字列 networkServerType：ネットワーク ファイル システム（NFS）、Common Internet File System（CIFS）または HTTP/S サーバ タイプを選択します。
- 文字列 locationLink：イメージの場所の有効な HTTP/HTTPS URL リンク。
- 文字列 networkPath：ネットワーク パス。
- 文字列 sharePath：ネットワーク共有パス。
- 文字列 remoteFileName：リモート ファイル名。
- 文字列 nwPathUserName：オプション。ネットワーク パスのユーザ名。
- 文字列 nwPathPasswprd：オプション。ネットワーク パスのパスワード。
- 文字列 mountOptions：オプション。有効なマウント オプション。

入力 XML の例

```
<cuicOperationRequest>
<operationType>NETWORK_IMAGE_UPDATE</operationType>
<payload>
<![CDATA[
<UpdateNetworkImage>
<imageId></imageId>

<platform></platform>

<networkServerType>NFS</networkServerType>

   <!-- Set this value only when networkServerType equals to HTTP  -->
<locationLink></locationLink>

   <!-- Set this value only when networkServerType not equals to HTTP  -->
<networkPath></networkPath>

   <!-- Set this value only when networkServerType not equals to HTTP  -->
<sharePath></sharePath>

   <!-- Set this value only when networkServerType not equals to HTTP  -->
<remoteFileName></remoteFileName>

<nwPathUserName></nwPathUserName>

<nwPathPassword></nwPathPassword>

   <!-- Set this value only when networkServerType equals to CIFS  -->
<mountOptions></mountOptions>

</UpdateNetworkImage>

]]>
</payload>
</cuicOperationRequest>
```

実装

プロファイル名は変更できません。プラットフォーム、サーバ タイプ（NFS/CIFS/HTTP）は必須です。NFS/CIFS の場合、リモート IP、リモート共有、リモート ファイル名が必須です。HTTP の場所は、システムから到達できなければなりません。

関連項目

ファームウェア ネットワーク イメージの作成, （3 ページ）

ファームウェア イメージ プロファイルの削除, （11 ページ）

ファームウェア ネットワーク イメージの検索

目的

Cisco.com のファームウェア イメージを検索します。

前提条件

ユーザには Cisco.com にログインするための有効な一連のクレデンシャルと、HUU ISO イメージに関するアクセス権が必要です。

REST URL

```
/cloupia/api-v2/FindFirmwareImage
```

コンポーネント

LOCAL_IMAGE_FIND API のパラメータは次のとおりです。

- 文字列 platform：プラットフォームの名前。
- 文字列 username：ISO 共有ログインのユーザ名。
- 文字列 password：ISO 共有ログインのパスワード。
- ブール値 enableProxy：オプション。プロキシ コンフィギュレーションを有効にします。
- 文字列 host：プロキシ コンフィギュレーションのホスト名。
- 文字列 port：プロキシ コンフィギュレーションのポート。
- ブール値 enableProxyAuth：オプション。プロキシ認証を有効にします。
- 文字列 proxyAuthUserName：プロキシ認証のプロキシ ユーザ名。
- 文字列 proxyAuthPassword：プロキシ ユーザ名のパスワード。

入力 XML の例

```
<cuicOperationRequest>
<operationType>LOCAL_IMAGE_FIND</operationType>
<payload>
<![CDATA[
<FindFirmwareImage>
<platform></platform>

<username></username>

<password></password>

<enableProxy>false</enableProxy>

   <!-- Set this value only when enableProxy equals to true  -->
<host></host>

   <!-- Set this value only when enableProxy equals to true  -->
<port>0</port>

   <!-- Set this value only when enableProxy equals to true  -->
<enableProxyAuth>false</enableProxyAuth>

   <!-- Set this value only when enableProxyAuth equals to true  -->
<proxyAuthUserName></proxyAuthUserName>

   <!-- Set this value only when enableProxyAuth equals to true  -->
<proxyAuthPassword></proxyAuthPassword>

</FindFirmwareImage>

]]>
</payload>
</cuicOperationRequest>
```

実装

Cisco.com のユーザ名とパスワード、およびプラットフォームは必須です。システムに既に追加されている、サーバのプラットフォームです。

関連項目

ファームウェア ローカル イメージの作成, （8 ページ）

ファームウェア ローカル イメージの作成

目的

アプライアンスのローカル場所にファームウェア イメージを作成します。

前提条件

ユーザには Cisco.com にログインするための有効な一連のクレデンシャルと、HUU ISO イメージに関するアクセス権が必要です。HUU イメージを cisco.com からダウンロードし、FindFirmwareImage API を使用して見つける必要があります。

REST URL

```
/cloupia/api-v2/CreateLocalImage
```

コンポーネント

LOCAL_IMAGE_CREATE API のパラメータは次のとおりです。

- 文字列 profileName：プロファイルの一意の名前。
- 文字列 platform：プラットフォームの名前。
- 文字列 username：ISO 共有ログインのユーザ名。
- 文字列 password：ISO 共有ログインのパスワード。
- 文字列 availableImage：使用可能な .iso イメージ。
- ブール値 enableProxy：オプション。プロキシ コンフィギュレーションを有効にします。
- 文字列 host：プロキシ コンフィギュレーションのホスト名。
- 文字列 port：プロキシ コンフィギュレーションのポート。
- ブール値 enableProxyAuth：オプション。プロキシ認証を有効にします。
- 文字列 proxyAuthUserName：プロキシ認証のプロキシ ユーザ名。
- 文字列 proxyAuthPassword：プロキシ ユーザ名のパスワード。
- ブール値 acceptLicense：ライセンス契約書を承認します。
- ブール値 downloadNow：プロファイルを追加した直後に、.iso イメージをダウンロードします。

入力 XML の例

```
<cuicOperationRequest>
<operationType>LOCAL_IMAGE_CREATE</operationType>
<payload>
<![CDATA[
<CreateLocalImage>
<profileName></profileName>

<platform></platform>

<username></username>

<password></password>

<availableImage></availableImage>

<enableProxy>false</enableProxy>

   <!-- Set this value only when enableProxy equals to true  -->
<host></host>

   <!-- Set this value only when enableProxy equals to true  -->
<port>0</port>

   <!-- Set this value only when enableProxy equals to true  -->
<enableProxyAuth>false</enableProxyAuth>

   <!-- Set this value only when enableProxyAuth equals to true  -->
<proxyAuthUserName></proxyAuthUserName>

   <!-- Set this value only when enableProxyAuth equals to true  -->
<proxyAuthPassword></proxyAuthPassword>

<acceptLicense>false</acceptLicense>

<downloadNow>false</downloadNow>

</CreateLocalImage>

]]>
</payload>
</cuicOperationRequest>
```

実装

プロファイル名は必須で、一意にする必要があります。Cisco.com のユーザ名とパスワード、およびプラットフォームは必須です。プラットフォームは、すでにシステムに追加されているサーバのプラットフォームにする必要があります。

関連項目

ファームウェア ネットワーク イメージの検索, （6 ページ）

ファームウェア ローカル イメージのダウンロード

目的

既に設定されているファームウェア イメージ プロファイルのイメージを cisco.com から、アプライアンス内のローカル場所にダウンロードします。

前提条件

ファームウェア イメージ プロファイルは事前に設定されている必要があります。

REST URL

```
/cloupia/api-v2/DownloadLocalImage
```

コンポーネント

LOCAL_IMAGE_DOWNLOAD API のパラメータは次のとおりです。

- 文字列 profileName：プロファイルの一意の名前。

入力 XML の例

```
<cuicOperationRequest>
<operationType>LOCAL_IMAGE_DOWNLOAD</operationType>
<payload>
<![CDATA[
<DownloadLocalImage>
<profileName></profileName>

</DownloadLocalImage>

]]>
</payload>
</cuicOperationRequest>
```

実装

プロファイル名は必須で、ローカル イメージの有効な既存のプロファイルでなければなりません。イメージはまだダウンロードされていない必要があります。

関連項目

ファームウェア ローカル イメージの作成, （8 ページ）
ファームウェア イメージ プロファイルの削除, （11 ページ）

ファームウェア イメージ プロファイルの削除

目的

1 つ以上の既存のファームウェア イメージ プロファイルを削除します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCFirmwareUpgradeConfig
```

コンポーネント

FIRMWARE_IMAGE_DELETE API のパラメータは次のとおりです。

- 文字列 profileId：プロファイルの一意の名前。

入力 XML の例

```
<cuicOperationRequest>
<operationType>FIRMWARE_IMAGE_DELETE</operationType>
<payload>
<![CDATA[
<DeleteFirmwareImage>
<profileId></profileId>

</DeleteFirmwareImage>

]]>
</payload>
</cuicOperationRequest>
```

実装

プロファイル名は必須で、一意にする必要があります。IP アドレスの検索条件は必須ですが、CSV ファイル オプションは API ではサポートされていません。

関連項目

ファームウェア ローカル イメージの作成, （8 ページ）
ファームウェア ネットワーク イメージの作成, （3 ページ）
ファームウェア ネットワーク イメージの更新, （5 ページ）

ファームウェア アップグレードの実行

目的

設定済みのファームウェア イメージ プロファイルを使用して、1 つ以上のサーバでファームウェア アップグレードを実行します。

前提条件

ファームウェア イメージ プロファイルはすでに設定されていて、有効な HUU ISO イメージが含まれている必要があります。

REST URL

```
/cloupia/api-v2/UpgradeFirmWareConfig
```

コンポーネント

RUN_FIRMWARE_UPGRADE API のパラメータは次のとおりです。

- 文字列 profileName：プロファイルの一意の名前。
- 文字列 servers：選択したプロファイルで設定されているプラットフォームが含まれるサーバ。
- ブール値 enableSchedule：スケジュールを有効にします。
- 文字列 associatedScheduleName：アソシエイト スケジュールの名前。

入力 XML の例

```
<cuicOperationRequest>
<operationType>RUN_FIRMWARE_UPGRADE</operationType>
<payload>
<![CDATA[
<UpgradeFirmWareConfig>
<profileName></profileName>

<servers></servers>

<enableSchedule>false</enableSchedule>

   <!-- Set this value only when enableSchedule not equals to false  -->
<associatedScheduleName></associatedScheduleName>

</UpgradeFirmWareConfig>

]]>
</payload>
</cuicOperationRequest>
```

実装

プロファイル名は必須で、有効な既存のプロファイルにする必要があります。ローカル プロファイルの場合、イメージがまだダウンロードされていない必要があります。serverIdKey は、ID のカンマ区切りのリストでなければなりません。各 ID の形式は次のとおりです。{AccountName};{ServerIPAddress}スケジュール オプションでは、有効なスケジュール名を指定しなければなりません。

関連項目

プロファイル名によるファームウェア アップグレード ステータスの読み取り，（16 ページ）

IP アドレスによるファームウェア アップグレード ステータスの読み取り，（16 ページ）

プロファイル名によるファームウェア イメージの読み取り

目的

プロファイル名でファームウェア イメージを取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCFirmwareUpgradeConfig/{CIMCFirmwareUpgradeConfigId}
```

実装

このタスクでは、プロファイル名に基づいて、ファームウェア イメージの詳細に対するクエリーを実行できます。CIMCFirmwareUpgradeConfigId 引数は有効なプロファイル名でなければなりません。引数を指定しないと、システム内に設定されているすべてのファームウェア イメージが返されます。

関連項目

プラットフォームによるファームウェア イメージの読み取り，（15 ページ）
タイプによるファームウェア イメージの読み取り，（14 ページ）

タイプによるファームウェア イメージの読み取り

目的

タイプによってファームウェア イメージを取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCFirmwareImageByType/{CIMCFirmwareImageByTypeId}
```

実装

このタスクでは、場所のタイプ（ネットワークまたはローカル）に基づいてファームウェア イメージの詳細に対するクエリーを実行できます。CIMCFirmwareImageByTypeId 引数は、NETWORK と LOCAL のどちらかの値でなければなりません。引数を指定しないと、システム内に設定されているすべてのファームウェア イメージが返されます。

関連項目

プラットフォームによるファームウェア イメージの読み取り，（15 ページ）
プロファイル名によるファームウェア イメージの読み取り，（13 ページ）

プラットフォームによるファームウェア イメージの読み取り

目的

プラットフォームによってファームウェア イメージを取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCFirmwareImageByPlatform/{CIMCFirmwareImageByPlatformId}
```

実装

このタスクでは、プラットフォームに基づいて、ファームウェアイメージの詳細にクエリーを実行できます。CIMCFirmwareImageByPlatformId 引数は有効なプロファイル名でなければなりません。引数を指定しないと、システム内に設定されているすべてのファームウェア イメージが返されます。

関連項目

プロファイル名によるファームウェア イメージの読み取り，（13 ページ）
タイプによるファームウェア イメージの読み取り，（14 ページ）

プロファイル名によるダウンロード ステータスの読み取り

目的

プロファイル名によってイメージ ダウンロード ステータスを取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/LocalImageDownloadStatusByProfileName/{LocalImageDownloadStatusByProfileNameId
```

実装

このタスクでは、プロファイル名に基づいて、ローカルのファームウェア イメージのダウンロード ステータスに対するクエリーを実行できます。
LocalImageDownloadStatusByProfileNameId 引数は、有効なプロファイル名でなければなりません。引数を指定しないと、空の結果セットが返されます。

関連項目

ファームウェア ローカル イメージのダウンロード, （10 ページ）

プロファイル名によるファームウェア アップグレード ステータスの読み取り

目的

プロファイル名によってファームウェア アップグレード ステータスを取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCFirmwareUpgradeStatusbyProfileName/{CIMCFirmwareUpgradeStatusbyProfileNameId}
```

実装

このタスクでは、イメージのプロファイル名に基づいて、1つ以上のサーバのファームウェア アップグレード ステータスに対するクエリーを実行できます。
CIMCFirmwareUpgradeStatusbyProfileNameId 引数は有効なプロファイル名でなければなりません。引数を指定しないと、すべてのファームウェア アップグレード操作ステータスが返されます。

関連項目

ファームウェア アップグレードの実行, （12 ページ）
IP アドレスによるファームウェア アップグレード ステータスの読み取り, （16 ページ）

IP アドレスによるファームウェア アップグレード ステータスの読み取り

目的

サーバの IP アドレスによってファームウェア アップグレード ステータスを取得します。

前提条件

なし

REST URL

```
>/cloupia/api-v2/CIMCFirmwareUpgradeStatusbyServerIP/{CIMCFirmwareUpgradeStatusbyServerIPId}
```

実装

このタスクでは、イメージのプロファイル名に基づいて、1つ以上のサーバのファームウェア アップグレード ステータスに対するクエリーを実行できます。
CIMCFirmwareUpgradeStatusbyProfileNameId 引数は有効なプロファイル名でなければなりません。引数を指定しないと、すべてのファームウェア アップグレード操作ステータスが返されます。IP アドレス内のドットはアンダースコアに置き換える必要があります。

関連項目

ファームウェア アップグレードの実行，（12 ページ）
プロファイル名によるファームウェア アップグレード ステータスの読み取り，（16 ページ）

プラットフォーム タスクの管理

概要

このカテゴリの例としては、Cisco IMC Supervisor における電子メール アラート ルールの管理があります。

電子メール アラート ルールの作成

目的

障害に関する通知の電子メール アラート ルールを作成します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCEmailAlertRuleConfig
```

コンポーネント

EMAIL_ALERT_RULE_CREATE API のパラメータは次のとおりです。

- 文字列 name：電子メール アラートの名前。
- 文字列 alertLevel：アラート レベル。
- 文字列 serverGroups：オプション。電子メール アラートの送信先サーバ グループ。
- 文字列 emailAddress：電子メール アラートの対象受信者の電子メール アドレス。
- 文字列 severity：電子メール アラートを送信する障害重大度レベル。
- ブール値 enabled：オプション。設定された電子メール アドレスに対する電子メール アラートを有効にします。

入力 XML の例

```
<cuicOperationRequest>
<operationType>EMAIL_ALERT_RULE_CREATE</operationType>
<payload>
<![CDATA[
<CIMCEmailAlertRuleConfig>
<name></name>

<alertLevel>SYSTEM</alertLevel>

   <!-- Set this value only when alertLevel not equals to SYSTEM  -->
<serverGroups></serverGroups>

<emailAddress></emailAddress>

<severity>critical</severity>

<enabled>false</enabled>

</CIMCEmailAlertRuleConfig>

]]>
</payload>
</cuicOperationRequest>
```

実装

ルール名は必須で、一意にする必要があります。電子メール アドレスは必須です。

関連項目

電子メール アラート ルールの読み取り

電子メール アラート ルールの更新

電子メール アラート ルールの削除

電子メール アラート ルールの読み取り

目的

電子メール アラート ルールの詳細を取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCEmailAlertRuleConfig/{CIMCEmailAlertRuleConfigId}
```

実装

Id引数は有効なルール名でなければなりません。引数を指定しないと、システム内に設定されているすべての電子メール アラート ルールが返されます。

関連項目

電子メール アラート ルールの作成

電子メール アラート ルールの更新

電子メール アラート ルールの削除

電子メール アラート ルールの更新

目的

既存の電子メール アラート ルールを更新します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCEmailAlertRuleConfig
```

コンポーネント

EMAIL_ALERT_RULE_UPDATE API のパラメータは次のとおりです。

- 文字列 emailAlertRule：電子メール アラート ルール。
- 文字列 alertLevel：アラート レベル。
- 文字列 serverGroups：オプション。電子メール アラートの送信先サーバ グループ。
- 文字列 emailAddress：この電子メールは、必要に応じてサービス リクエストおよびリクエスト承認のステータスをグループ所有者に通知する目的で使用されます。
- 文字列 severity：電子メール アラートを送信する障害重大度レベル。
- ブール値 enabled：オプション。設定された電子メール アドレスに対する電子メール アラートを有効にします。

入力 XML の例

```
<cuicOperationRequest>
<operationType>EMAIL_ALERT_RULE_UPDATE</operationType>
<payload>
<![CDATA[
<ModifyEmailAlertRuleConfig>
<emailAlertRule></emailAlertRule>

<alertLevel>SYSTEM</alertLevel>

   <!-- Set this value only when alertLevel not equals to SYSTEM  -->
<serverGroups></serverGroups>

<emailAddress></emailAddress>

<severity></severity>

<enabled>false</enabled>

</ModifyEmailAlertRuleConfig>

]]>
</payload>
</cuicOperationRequest>
```

実装

ルール名は変更できません。

関連項目

電子メール アラート ルールの読み取り

電子メール アラート ルールの作成

電子メール アラート ルールの削除

電子メール アラート ルールの削除

目的

1 つ以上の既存の電子メール アラート ルールを削除します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCEmailAlertRuleConfig
```

コンポーネント

文字列 emailAlertRule：電子メール アラート ルール。

入力 XML の例

```
<cuicOperationRequest>
<operationType>EMAIL_ALERT_RULE_DELETE</operationType>
<payload>
<![CDATA[
<DeleteEmailAlertRuleConfig>
<emailAlertRule></emailAlertRule>

</DeleteEmailAlertRuleConfig>

]]>
</payload>
</cuicOperationRequest>
```

実装

ルール名のカンマ区切りリスト。すべてが有効な既存のルールでなければなりません。

関連項目

電子メール アラート ルールの読み取り

電子メール アラート ルールの作成

電子メール アラート ルールの更新

サーバ タスクの管理

概要

このカテゴリの例としては、さまざまなサーバ管理タスク、つまりIPアドレスによるサーバの検出、検出されたサーバのインポート、サーバにおける電源操作、サーバ データ、インベントリ データ、障害データに対するクエリーを実行するための各種方法などがあります。

ラック グループの作成

目的

Cisco IMC Supervisor でサーバを論理的にグループ化するためのラック グループを作成します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCRackGroup
```

コンポーネント

RACK_GROUP_CREATE API のパラメータは次のとおりです。

- 文字列 groupName：グループまたは顧客組織の名前。
- 文字列 groupDescription：オプション。グループまたは顧客組織の説明（必要な場合）。

入力 XML の例

```
<cuicOperationRequest>
<operationType>RACK_GROUP_CREATE</operationType>
<payload>
<![CDATA[
<CIMCRackGroup>
<groupName></groupName>

<description></description>

</CIMCRackGroup>

]]>
</payload>
</cuicOperationRequest>
```

実装

グループ名は必須で、一意にする必要があります。

関連項目

すべてのラック グループの読み取り，（23 ページ）
ラック グループの更新，（24 ページ）
ラック グループの削除，（25 ページ）

すべてのラック グループの読み取り

目的

ラック グループの詳細を取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCRackGroup/{CIMCRackGroupId}
```

コンポーネント

なし

入力 XML の例

```
<cuicOperationResponse><cuicOperationStatus>0</cuicOperationStatus>
<response><CIMCRackGroup><actionId>0</actionId><configEntryId>0</configEntryId>
<defaultGroup>true</defaultGroup><description>Default provided rack group
</description><groupName>Default Group</groupName></CIMCRackGroup><CIMCRackGroup>
<actionId>0</actionId><configEntryId>0</configEntryId><defaultGroup>false
</defaultGroup><description>Test55</description><groupName>Test66</groupName>
</CIMCRackGroup><CIMCRackGroup><actionId>0</actionId><configEntryId>0
</configEntryId><defaultGroup>false</defaultGroup><description>apitest
</description><groupName>apitest-ren</groupName></CIMCRackGroup><CIMCRackGroup>
<actionId>0</actionId><configEntryId>0</configEntryId><defaultGroup>false
</defaultGroup><description></description><groupName>Test3-SumanthRen</groupName>
</CIMCRackGroup></response></cuicOperationResponse>
```

実装

ID 引数は、有効なラック グループ名でなければなりません。引数を指定しないと、システム内に設定されているすべてのラック グループが返されます。

関連項目

ラック グループの作成, （22 ページ）
ラック グループの更新, （24 ページ）
ラック グループの削除, （25 ページ）

ラック グループの更新

目的

既存のラック グループを更新します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCRackGroup
```

コンポーネント

RACK_GROUP_UPDATE API のパラメータは次のとおりです。

- 文字列 groupName：グループまたは顧客組織の名前。
- 文字列 groupDescription：オプション。グループまたは顧客組織の説明（必要な場合）。

入力 XML の例

```
<cuicOperationRequest>
<operationType>RACK_GROUP_UPDATE</operationType>
<payload>
<![CDATA[
<ModifyRackGroup>
<groupID></groupID>

<groupName></groupName>

<description></description>

</ModifyRackGroup>

]]>
</payload>
</cuicOperationRequest>
```

実装

グループ名は必須で、一意にする必要があります。

関連項目

ラック グループの作成, （22 ページ）
すべてのラック グループの読み取り, （23 ページ）
ラック グループの削除, （25 ページ）

ラック グループの削除

目的

1 つ以上の既存のラック グループを削除します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCRackGroup
```

コンポーネント

RACK_GROUP_DELETE API のパラメータは次のとおりです。

- 文字列 groupName：グループまたは顧客組織の名前。
- 文字列 groupDescription：オプション。グループまたは顧客組織の説明（必要な場合）。

入力 XML の例

```
<cuicOperationRequest>
<operationType>RACK_GROUP_DELETE</operationType>
<payload>
<![CDATA[
<DeleteRackGroup>
<groupID></groupID>

<forceDelete>false</forceDelete>

</DeleteRackGroup>

]]>
</payload>
</cuicOperationRequest>
```

実装

グループ名のカンマ区切りリスト。すべてが有効な既存のラック グループでなければなりません。

関連項目

ラック グループの作成, （22 ページ）
すべてのラック グループの読み取り, （23 ページ）
ラック グループの更新, （24 ページ）

検出プロファイルの作成

目的

IP アドレスに基づいてサーバを検出してインポートする際に使用する検出プロファイルを作成します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCDeviceDiscoveryConfig
```

コンポーネント

DISCOVERY_PROFILE_CREATE API のパラメータは次のとおりです。

- 文字列 profileName：プロファイルの名前。
- ブール値 isRange：オプション。範囲
- 文字列 option：オプション。
- 文字列 ipList：IP アドレスのリスト。
- 文字列 startRange：IP アドレス範囲の有効な先頭。
- 文字列 endRange：IP アドレス範囲の有効な最後。
- 文字列 networkAddress：ネットワーク IP アドレス。
- 文字列 subnetMask：サブネット マスクの範囲。
- 文字列 csvFile：csv ファイルによる検索。
- ブール値 credentialPolicy：オプション。クレデンシャル ポリシーを作成します。
- 文字列 policy：オプション。ポリシー名。
- 文字列 username：サーバのログイン名。
- 文字列 password：サーバのログイン パスワード。
- 文字列 protocol：オプション。HTTP プロトコルまたは HTTPS プロトコル。
- Int 型 port：ポート番号。

入力 XML の例

```
<cuicOperationRequest>
<operationType>DISCOVERY_PROFILE_CREATE</operationType>
<payload>
<![CDATA[
<CIMCDeviceDiscoveryConfig>
<profileName></profileName>

<option>IP</option>

   <!-- Set this value only when option equals to IPLIST  -->
<ipList></ipList>

   <!-- Set this value only when option equals to IP  -->
<startRange></startRange>

   <!-- Set this value only when option equals to IP  -->
<endRange></endRange>

   <!-- Set this value only when option equals to SUBNET  -->
<networkAddress></networkAddress>

   <!-- Set this value only when option equals to SUBNET  -->
<subnetMask></subnetMask>

   <!-- Set this value only when option equals to CSV  -->
<csvFile></csvFile>

<credentialPolicy>false</credentialPolicy>

   <!-- Set this value only when credentialPolicy not equals to false  -->
<policy></policy>

   <!-- Set this value only when credentialPolicy not equals to true  -->
<username></username>

   <!-- Set this value only when credentialPolicy not equals to true  -->
<password></password>

   <!-- Set this value only when credentialPolicy not equals to true  -->
<protocol>https</protocol>

   <!-- Set this value only when credentialPolicy not equals to true  -->
<port>443</port>

</CIMCDeviceDiscoveryConfig>

]]>
</payload>
</cuicOperationRequest>
```

実装

プロファイル名は必須で、一意にする必要があります。IP アドレスの検索条件は必須ですが、CSV ファイル オプションは、API ではサポートされていません。

関連項目

検出プロファイルの更新, （29 ページ）

検出プロファイルの削除, （32 ページ）

検出プロファイルの読み取り

目的

検出プロファイルの詳細を取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCDeviceDiscoveryConfig/{CIMCDeviceDiscoveryConfigId}
```

実装

Id引数は有効なプロファイル名でなければなりません。引数を指定しないと、システム内に設定されているすべての検出プロファイルが返されます。

関連項目

検出プロファイルの作成, （26 ページ）
検出プロファイルの更新, （29 ページ）
検出プロファイルの削除, （32 ページ）

検出プロファイルの更新

目的

既存の検出プロファイルを更新します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCDeviceDiscoveryConfig
```

コンポーネント

DISCOVERY_PROFILE_UPDATE API のパラメータは次のとおりです。

- 文字列 profileName：プロファイルの一意の名前。
- 文字列 option：オプション。
- 文字列 ipList：IP アドレスのリスト。
- 文字列 startRange： IP アドレス範囲の有効な先頭。
- 文字列 endRange：IP アドレス範囲の有効な最後。
- 文字列 networkAddress：ネットワーク IP アドレス。
- 文字列 subnetMask：サブネット マスクの範囲。
- 文字列 csvFile：csv ファイルによる検索。
- ブール値 credentialPolicy：オプション。クレデンシャル ポリシーを作成します。
- ブール値 policy：オプション。ポリシー名。
- 文字列 username：サーバのログイン名。
- 文字列 password：サーバのログイン パスワード。
- 文字列 protocol：オプション。HTTP プロトコルまたは HTTPS プロトコル。
- Int 型 port：ポート番号。

入力 XML の例

```
<cuicOperationRequest>
<operationType>DISCOVERY_PROFILE_UPDATE</operationType>
<payload>
<![CDATA[
<ModifyCIMCDeviceDiscoveryProfile>
<profileName></profileName>

<option>IP</option>

   <!-- Set this value only when option equals to IPLIST  -->
<ipList></ipList>

   <!-- Set this value only when option equals to IP  -->
<startRange></startRange>

   <!-- Set this value only when option equals to IP  -->
<endRange></endRange>

   <!-- Set this value only when option equals to SUBNET  -->
<networkAddress></networkAddress>

   <!-- Set this value only when option equals to SUBNET  -->
<subnetMask></subnetMask>

   <!-- Set this value only when option equals to CSV  -->
<csvFile></csvFile>

<credentialPolicy>false</credentialPolicy>

   <!-- Set this value only when credentialPolicy not equals to false  -->
<policy></policy>

   <!-- Set this value only when credentialPolicy not equals to true  -->
<username></username>

   <!-- Set this value only when credentialPolicy not equals to true  -->
<password></password>

   <!-- Set this value only when credentialPolicy not equals to true  -->
<protocol>https</protocol>

   <!-- Set this value only when credentialPolicy not equals to true  -->
<port>443</port>

</ModifyCIMCDeviceDiscoveryProfile>

]]>
</payload>
</cuicOperationRequest>
```

実装

プロファイル名は変更できません。

関連項目

検出プロファイルの作成, （26 ページ）

検出プロファイルの削除, （32 ページ）

検出プロファイルの削除

目的

1 つ以上の既存の検出プロファイルを削除します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCDeviceDiscoveryConfig
```

コンポーネント

DISCOVERY_PROFILE_DELETE API のパラメータは次のとおりです。

- 文字列 profileName：オプション。プロファイル名。

入力 XML の例

```
<cuicOperationRequest>
<operationType>DISCOVERY_PROFILE_DELETE</operationType>
<payload>
<![CDATA[
<DeleteCIMCDeviceDiscoveryProfile>
<profileName></profileName>

</DeleteCIMCDeviceDiscoveryProfile>

]]>
</payload>
</cuicOperationRequest>
```

実装

プロファイル名のカンマ区切りリスト。すべてが有効な既存のプロファイルでなければなりません。

関連項目

検出プロファイルの作成，（26 ページ）
検出プロファイルの更新，（29 ページ）
検出プロファイルの読み取り，（29 ページ）

サーバ ディスカバリの実行

目的

1つ以上の設定された検出プロファイルを使用して、IPアドレスに基づいてサーバを検出するためのディスカバリ操作を実行します。

前提条件

検出プロファイルを設定しておく必要があります。

REST URL

```
/cloupia/api-v2/CIMCAutoDiscoveryConfig
```

コンポーネント

RUN_SERVER_DISCOVERY API のパラメータは次のとおりです。

- 文字列 profileNames：プロファイルの名前。
- ブール値 enableSchedule：スケジュールを有効にします。
- 文字列 associatedScheduleName：アソシエイト スケジュールの名前。

入力 XML の例

```
<cuicOperationRequest>
<operationType>RUN_SERVER_DISCOVERY</operationType>
<payload>
<![CDATA[
<CIMCAutoDiscoveryConfig>
<profileNames></profileNames>

<enableSchedule>false</enableSchedule>

   <!-- Set this value only when enableSchedule not equals to false  -->
<associatedScheduleName></associatedScheduleName>

</CIMCAutoDiscoveryConfig>

]]>
</payload>
</cuicOperationRequest>
```

実装

有効なプロファイル名のカンマ区切りリスト。スケジュール オプションでは、有効なスケジュール名を指定しなければなりません。

関連項目

検出されたデバイスのインポート，（34 ページ）

検出されたデバイスの読み取り

目的

検出されたデバイスの詳細を取得します。

前提条件

1 つ以上のサーバが、検出プロファイルを使用して検出されている必要があります。

REST URL

```
/cloupia/api-v2/CIMCDiscoveredDevice/{CIMCDiscoveredDeviceId}/State/{StateId}
```

実装

CIMCDiscoveredDeviceId 引数は有効なプロファイル名でなければならず、必須です。StateId 引数は、All、Imported、NotImported のいずれかにする必要があります。

検出されたデバイスのインポート

目的

1 つ以上の検出されたデバイスをインポートします。

前提条件

1 つ以上のサーバが、検出プロファイルを使用して検出されている必要があります。

REST URL

```
/cloupia/api-v2/ImportRackServersConfig
```

コンポーネント

IMPORT_SERVER API のパラメータは次のとおりです。

- 文字列 devices：検出されたデバイス。
- 文字列 userPrefix：オプション。ユーザのプレフィックスです。
- 文字列 description：オプション。ユーザに関する説明です。
- 文字列 contact：オプション。ユーザの連絡先詳細です。
- 文字列 location：オプション。ユーザの所在地です。
- 文字列 rackGroup：ラック グループを作成します。

入力 XML の例

```
<cuicOperationRequest>
<operationType>IMPORT_SERVER</operationType>
<payload>
<![CDATA[
<ImportRackServersConfig>
<devices></devices>

<userPrefix></userPrefix>

<description></description>

<contact></contact>

<location></location>

<rackGroup>Default Group</rackGroup>

</ImportRackServersConfig>

]]>
</payload>
</cuicOperationRequest>
```

実装

検出された 1 つ以上の有効なサーバ IP アドレスのカンマ区切りリスト。既存のラック グループのグループ名。

関連項目

サーバ ディスカバリの実行, （33 ページ）

サーバのハード リセット

目的

1 つ以上のサーバをハード リセットします。

前提条件

1 つ以上のサーバがラック アカウントとして設定されている必要があります。

REST URL

```
/cloupia/api-v2/HardResetAction
```

コンポーネント

HARD_RESET_SERVER API のパラメータは次のとおりです。

- 文字列 serverIdKey：サーバ ID キー。

入力 XML の例

```
<cuicOperationRequest>
<operationType>HARD_RESET_SERVER</operationType>
<payload>
<![CDATA[
<HardResetServer>
<serverIdKey></serverIdKey>

</HardResetServer>

]]>
</payload>
</cuicOperationRequest>
```

実装

serverIdKey は、ID のカンマ区切りのリストでなければなりません。各 ID の形式は次のとおりです。{AccountName};{ServerIPAddress }

関連項目

電源の再投入 サーバ, （36 ページ）
電源オン サーバ, （38 ページ）
電源オフ サーバ, （37 ページ）
サーバのシャットダウン, （39 ページ）
サーバのラベルの設定, （40 ページ）
サーバのロケータ LED の切り替え, （41 ページ）

電源の再投入 サーバ

目的

1 つ以上のサーバの電源を再投入します。

前提条件

1 つ以上のサーバがラック アカウントとして設定されている必要があります。

REST URL

```
/cloupia/api-v2/PowerCycleAction
```

コンポーネント

POWER_CYCLE_SERVER API のパラメータは次のとおりです。

- 文字列 serverIdKey：サーバ ID キー。

入力 XML の例

```
<cuicOperationRequest>
<operationType>POWER_CYCLE_SERVER</operationType>
<payload>
<![CDATA[
<PowerCycleServer>
<serverIdKey></serverIdKey>

</PowerCycleServer>

]]>
</payload>
</cuicOperationRequest>
```

実装

serverIdKeyは、IDのカンマ区切りのリストでなければなりません。各IDの形式は次のとおりです。{AccountName};{ServerIPAddress }

関連項目

サーバのハード リセット, （35 ページ）
電源オン サーバ, （38 ページ）
電源オフ サーバ, （37 ページ）
サーバのシャットダウン, （39 ページ）
サーバのラベルの設定, （40 ページ）
サーバのロケータ LED の切り替え, （41 ページ）

電源オフ サーバ

目的

1 つ以上のサーバの電源をオフにします。

前提条件

1 つ以上のサーバがラック アカウントとして設定されている必要があります。

REST URL

```
/cloupia/api-v2/PowerOffAction
```

コンポーネント

POWER_OFF_SERVER API のパラメータは次のとおりです。

- 文字列 serverIdKey：サーバ ID キー。

入力 XML の例

```
<cuicOperationRequest>
<operationType>POWER_OFF_SERVER</operationType>
<payload>
<![CDATA[
<PowerOffServer>
<serverIdKey></serverIdKey>

</PowerOffServer>

]]>
</payload>
</cuicOperationRequest>
```

実装

serverIdKey は、ID のカンマ区切りのリストでなければなりません。各 ID の形式は次のとおりです。{AccountName};{ServerIPAddress}

関連項目

サーバのハード リセット, （35 ページ）
電源の再投入 サーバ, （36 ページ）
電源オン サーバ, （38 ページ）
サーバのシャットダウン, （39 ページ）
サーバのラベルの設定, （40 ページ）
サーバのロケータ LED の切り替え, （41 ページ）

電源オン サーバ

目的

サーバの電源をオンにします。

コンテキスト

1 つ以上のサーバの電源をオンにします。

前提条件

1 つ以上のサーバがラック アカウントとして設定されている必要があります。

REST URL

```
/cloupia/api-v2/PowerOnAction
```

コンポーネント

POWER_ON_SERVER API のパラメータは次のとおりです。

- 文字列 serverIdKey：サーバ ID キー。

入力 XML の例

```
<cuicOperationRequest>
<operationType>POWER_ON_SERVER</operationType>
<payload>
<![CDATA[
<PowerOnServer>
<serverIdKey></serverIdKey>

</PowerOnServer>

]]>
</payload>
</cuicOperationRequest>
```

実装

serverIdKey は、ID のカンマ区切りのリストでなければなりません。各 ID の形式は次のとおりです。{AccountName};{ServerIPAddress}

関連項目

サーバのハード リセット，（35 ページ）
電源の再投入 サーバ，（36 ページ）
電源オフ サーバ，（37 ページ）
サーバのシャットダウン，（39 ページ）
サーバのラベルの設定，（40 ページ）
サーバのロケータ LED の切り替え，（41 ページ）

サーバのシャットダウン

目的

1 つ以上のサーバをシャットダウンします。

前提条件

1 つ以上のサーバがラック アカウントとして設定されている必要があります。

REST URL

```
/cloupia/api-v2/ShutDownAction
```

コンポーネント

SHUT_DOWN_SERVER API のパラメータは次のとおりです。

- 文字列 serverIdKey：サーバ ID キー。

入力 XML の例

```
<cuicOperationRequest>
<operationType>SHUT_DOWN_SERVER</operationType>
<payload>
<![CDATA[
<ShutDownServer>
<serverIdKey></serverIdKey>

</ShutDownServer>

]]>
</payload>
</cuicOperationRequest>
```

実装

serverIdKey は、ID のカンマ区切りのリストでなければなりません。各 ID の形式は次のとおりです。{AccountName};{ServerIPAddress}

関連項目

電源の再投入 サーバ, （36 ページ）
電源オン サーバ, （38 ページ）
電源オフ サーバ, （37 ページ）
サーバのハード リセット, （35 ページ）
サーバのラベルの設定, （40 ページ）
サーバのロケータ LED の切り替え, （41 ページ）

サーバのラベルの設定

目的

1 つ以上のサーバのラベルを設定します。

前提条件

1 つ以上のサーバがラック アカウントとして設定されている必要があります。

REST URL

```
/cloupia/api-v2/SetLabelAction
```

コンポーネント

SET_LABEL API のパラメータは次のとおりです。

- 文字列 serverIdKey：サーバ ID キー。
- 文字列 setLabel：ラベル名。

入力 XML の例

```
<cuicOperationRequest>
<operationType>SET_LABEL</operationType>
<payload>
<![CDATA[
<SetLabelServer>
<serverIdKey></serverIdKey>

<setLabel></setLabel>

</SetLabelServer>

]]>
</payload>
</cuicOperationRequest>
```

実装

serverIdKey は、ID のカンマ区切りのリストでなければなりません。各 ID の形式は次のとおりです。{AccountName};{ServerIPAddress}

関連項目

電源の再投入 サーバ, （36 ページ）
電源オン サーバ, （38 ページ）
電源オフ サーバ, （37 ページ）
サーバのシャットダウン, （39 ページ）
サーバのハード リセット, （35 ページ）
サーバのロケータ LED の切り替え, （41 ページ）

サーバのロケータ LED の切り替え

目的

1 つ以上サーバのロケータ LED を切り替えます。

前提条件

1 つ以上のサーバがラック アカウントとして設定されている必要があります。

REST URL

```
/cloupia/api-v2/LocatorLedAction
```

コンポーネント

LOCATOR_LED API のパラメータは次のとおりです。

- 文字列 serverIdKey：サーバ ID キー。
- 文字列 locatorLed：ロケータ LED。

入力 XML の例

```
<cuicOperationRequest>
<operationType>LOCATOR_LED</operationType>
<payload>
<![CDATA[
<LocatorLedServer>
<serverIdKey></serverIdKey>

<locatorLed>ON</locatorLed>

</LocatorLedServer>

]]>
</payload>
</cuicOperationRequest>
```

実装

serverIdKey は、ID のカンマ区切りのリストでなければなりません。各 ID の形式は次のとおりです。{AccountName};{ServerIPAddress}

関連項目

電源の再投入 サーバ, （36 ページ）
電源オン サーバ, （38 ページ）
電源オフ サーバ, （37 ページ）
サーバのシャットダウン, （39 ページ）
サーバのラベルの設定, （40 ページ）
サーバのハード リセット, （35 ページ）

タグ名によるサーバの読み取り

目的

特定の名前によってタグ付けされているサーバを取得します。

前提条件

1 つ以上のサーバがラック アカウントとして設定され、タグ付けされている必要があります。

REST URL

```
/cloupia/api-v2/ServersByTagName/{ServersByTagNameId}
```

実装

ServersByTagValueId 引数は、タグ ライブラリで定義されている有効なタグ値でなければなりません。

関連項目

アカウント名によるサーバの読み取り，（50 ページ）
ラック グループによるサーバの読み取り，（53 ページ）
シリアル番号によるサーバの読み取り，（53 ページ）
サーバ IP によるサーバの読み取り，（52 ページ）
タグ値によるサーバの読み取り，（43 ページ）
UUID によるサーバの読み取り，（51 ページ）
製品 ID によるサーバの読み取り，（50 ページ）

タグ値によるサーバの読み取り

目的

特定の値によってタグ付けされているサーバを取得します。

前提条件

1 つ以上のサーバがラック アカウントとして設定され、タグ付けされている必要があります。

REST URL

```
/cloupia/api-v2/ServersByTagValue/{ServersByTagValueId}
```

実装

ServersByTagValueId 引数は、タグ ライブラリで定義されている有効なタグ値でなければなりません。

関連項目

タグ名によるサーバの読み取り，（42 ページ）
アカウント名によるサーバの読み取り，（50 ページ）
ラック グループによるサーバの読み取り，（53 ページ）
シリアル番号によるサーバの読み取り，（53 ページ）
サーバ IP によるサーバの読み取り，（52 ページ）
UUID によるサーバの読み取り，（51 ページ）
製品 ID によるサーバの読み取り，（50 ページ）

DN によるサーバ障害の読み取り

目的

関係する DN によってサーバ障害を取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCFaultsByDN/{CIMCFaultsByDNId}
```

実装

CIMCFaultsByDNId 引数は有効な DN 値でなければなりません。DN の RN は、スラッシュではなくアンダースコアで区切る必要があります。

関連項目

アカウント名によるサーバ障害の読み取り，（45 ページ）
障害コードによるサーバ障害の読み取り，（46 ページ）
IP アドレスによるサーバ障害の読み取り，（44 ページ）
重大度によるサーバ障害の読み取り，（46 ページ）

IP アドレスによるサーバ障害の読み取り

目的

サーバの IP アドレスによってその障害を取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCFaultsByServerIP/{CIMCFaultsByServerIPId}
```

実装

CIMCFaultsByServerIPId 引数は、有効な IP アドレスでなければなりません。IP アドレス内のドットはアンダースコアに置き換える必要があります。

関連項目

DN によるサーバ障害の読み取り，（44 ページ）
障害コードによるサーバ障害の読み取り，（46 ページ）
アカウント名によるサーバ障害の読み取り，（45 ページ）
重大度によるサーバ障害の読み取り，（46 ページ）

アカウント名によるサーバ障害の読み取り

目的

サーバのアカウント名によってその障害を取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCFaultsByAccountName/{CIMCFaultsByAccountNameId}
```

実装

CIMCFaultsByAccountNameId 引数は、IMCS によって管理されているサーバの有効なアカウント名でなければなりません。

関連項目

DN によるサーバ障害の読み取り，（44 ページ）
障害コードによるサーバ障害の読み取り，（46 ページ）
IP アドレスによるサーバ障害の読み取り，（44 ページ）
重大度によるサーバ障害の読み取り，（46 ページ）

重大度によるサーバ障害の読み取り

目的

重大度レベルによってサーバ障害を取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCFaultsBySeverity/{CIMCFaultsBySeverityId}
```

実装

CIMCFaultsBySeverityId 引数は、有効な重大度レベルでなければなりません。

関連項目

DN によるサーバ障害の読み取り，（44 ページ）
障害コードによるサーバ障害の読み取り，（46 ページ）
IP アドレスによるサーバ障害の読み取り，（44 ページ）
アカウント名によるサーバ障害の読み取り，（45 ページ）

障害コードによるサーバ障害の読み取り

目的

障害コードによってサーバ障害を取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCFaultsByCode/{CIMCFaultsByCodeId}
```

実装

CIMCFaultsByCodeId 引数は、有効な障害コードでなければなりません。

関連項目

DN によるサーバ障害の読み取り，（44 ページ）
アカウント名によるサーバ障害の読み取り，（45 ページ）
IP アドレスによるサーバ障害の読み取り，（44 ページ）
重大度によるサーバ障害の読み取り，（46 ページ）

DN によるサーバ障害履歴の読み取り

目的

関係する DN によってサーバ障害を取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCFaultsHistoryByDN/{CIMCFaultsHistoryByDNId}
```

実装

CIMCFaultsHistoryByDNId 引数は、有効な DN 値でなければなりません。DN の RN は、スラッシュではなくアンダースコアで区切る必要があります。

関連項目

障害コードによるサーバ障害履歴の読み取り，（49 ページ）
IP アドレスによるサーバ障害履歴の読み取り，（47 ページ）
重大度によるサーバ障害履歴の読み取り，（49 ページ）
アカウント名によるサーバ障害履歴の読み取り，（48 ページ）

IP アドレスによるサーバ障害履歴の読み取り

目的

サーバの IP アドレスによってその障害履歴を取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCFaultsHistoryByServerIP/{CIMCFaultsHistoryByServerIPId}
```

実装

CIMCFaultsHistoryByServerIPId 引数は、IMCS によって管理されているサーバの有効な IP アドレスでなければなりません。IP アドレス内のドットはアンダースコアに置き換える必要があります。

関連項目

障害コードによるサーバ障害履歴の読み取り，（49 ページ）
DN によるサーバ障害履歴の読み取り，（47 ページ）
重大度によるサーバ障害履歴の読み取り，（49 ページ）
アカウント名によるサーバ障害履歴の読み取り，（48 ページ）

アカウント名によるサーバ障害履歴の読み取り

目的

サーバのアカウント名によってその障害履歴を取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCFaultsHistoryByAccountName/{CIMCFaultsHistoryByAccountNameId}
```

実装

CIMCFaultsHistoryByAccountNameId 引数は、Cisco IMC Supervisor によって管理されているサーバの有効なアカウント名でなければなりません。

関連項目

障害コードによるサーバ障害履歴の読み取り，（49 ページ）
DN によるサーバ障害履歴の読み取り，（47 ページ）
重大度によるサーバ障害履歴の読み取り，（49 ページ）
IP アドレスによるサーバ障害履歴の読み取り，（47 ページ）

重大度によるサーバ障害履歴の読み取り

目的

重大度レベルによってサーバ障害履歴を取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCFaultsHistoryBySeverity/{CIMCFaultsHistoryBySeverityId}
```

実装

CIMCFaultsHistoryBySeverityId 引数は有効な重大度レベルでなければなりません。

関連項目

障害コードによるサーバ障害履歴の読み取り，（49 ページ）
DN によるサーバ障害履歴の読み取り，（47 ページ）
アカウント名によるサーバ障害履歴の読み取り，（48 ページ）
IP アドレスによるサーバ障害履歴の読み取り，（47 ページ）

障害コードによるサーバ障害履歴の読み取り

目的

障害コードによってサーバ障害履歴を取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCFaultsHistoryByCode/{CIMCFaultsHistoryByCodeId}
```

実装

CIMCFaultsHistoryByCodeId 引数は有効な障害コードでなければなりません。

関連項目

重大度によるサーバ障害履歴の読み取り，（49 ページ）
DN によるサーバ障害履歴の読み取り，（47 ページ）
アカウント名によるサーバ障害履歴の読み取り，（48 ページ）
IP アドレスによるサーバ障害履歴の読み取り，（47 ページ）

製品 ID によるサーバの読み取り

目的

製品 ID によってサーバを取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCServerByProductID/{CIMCServerByProductIDId}
```

実装

CIMCServerByProductIDId 引数は、Cisco IMC Supervisor によって管理されているサーバの有効な製品 ID でなければなりません。

関連項目

タグ名によるサーバの読み取り，（42 ページ）
アカウント名によるサーバの読み取り，（50 ページ）
ラック グループによるサーバの読み取り，（53 ページ）
シリアル番号によるサーバの読み取り，（53 ページ）
サーバ IP によるサーバの読み取り，（52 ページ）
UUID によるサーバの読み取り，（51 ページ）
タグ値によるサーバの読み取り，（43 ページ）

アカウント名によるサーバの読み取り

目的

アカウント名によってサーバを取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCServerByAccountName/{CIMCServerByAccountNameId}
```

実装

CIMCServerByAccountNameId 引数は、Cisco IMC Supervisor によって管理されているサーバの有効なアカウント名でなければなりません。

関連項目

タグ名によるサーバの読み取り，（42 ページ）
タグ値によるサーバの読み取り，（43 ページ）
ラック グループによるサーバの読み取り，（53 ページ）
シリアル番号によるサーバの読み取り，（53 ページ）
サーバ IP によるサーバの読み取り，（52 ページ）
UUID によるサーバの読み取り，（51 ページ）
製品 ID によるサーバの読み取り，（50 ページ）

UUID によるサーバの読み取り

目的

UUID によってサーバを取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCServerByUUID/{CIMCServerByUUIDId}
```

実装

CIMCServerByUUIDId 引数は、Cisco IMC Supervisor によって管理されているサーバの有効な UUID でなければなりません。

関連項目

タグ名によるサーバの読み取り，（42 ページ）
タグ値によるサーバの読み取り，（43 ページ）
アカウント名によるサーバの読み取り，（50 ページ）
ラック グループによるサーバの読み取り，（53 ページ）
シリアル番号によるサーバの読み取り，（53 ページ）
サーバ IP によるサーバの読み取り，（52 ページ）
製品 ID によるサーバの読み取り，（50 ページ）

サーバ IP によるサーバの読み取り

目的

IP アドレスによってサーバを取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCServerByServerIP/{CIMCServerByServerIPId}
```

実装

CIMCServerByServerIPId 引数は、Cisco IMC Supervisor によって管理されているサーバの有効な IP アドレスでなければなりません。IP アドレス内のドットはアンダースコアに置き換える必要があります。

関連項目

タグ名によるサーバの読み取り，（42 ページ）
アカウント名によるサーバの読み取り，（50 ページ）
ラック グループによるサーバの読み取り，（53 ページ）
シリアル番号によるサーバの読み取り，（53 ページ）
サーバ IP によるサーバの読み取り，（52 ページ）
UUID によるサーバの読み取り，（51 ページ）
製品 ID によるサーバの読み取り，（50 ページ）

シリアル番号によるサーバの読み取り

目的

シリアル番号によってサーバを取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCServerBySerialNum/{CIMCServerBySerialNumId}
```

実装

CIMCServerBySerialNumId 引数は、Cisco IMC Supervisor によって管理されているサーバの有効なシリアル番号でなければなりません。

関連項目

タグ名によるサーバの読み取り，（42 ページ）
タグ値によるサーバの読み取り，（43 ページ）
アカウント名によるサーバの読み取り，（50 ページ）
ラック グループによるサーバの読み取り，（53 ページ）
サーバ IP によるサーバの読み取り，（52 ページ）
製品 ID によるサーバの読み取り，（50 ページ）
UUID によるサーバの読み取り，（51 ページ）

ラック グループによるサーバの読み取り

目的

ラック グループによってサーバを取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCServerByRackGroup/{CIMCServerByRackGroupId}
```

実装

CIMCServerByRackGroupId 引数は、Cisco IMC Supervisor 内に存在する有効なラック グループでなければなりません。

関連項目

タグ名によるサーバの読み取り，（42 ページ）
タグ値によるサーバの読み取り，（43 ページ）
アカウント名によるサーバの読み取り，（50 ページ）
サーバ IP によるサーバの読み取り，（52 ページ）
シリアル番号によるサーバの読み取り，（53 ページ）
製品 ID によるサーバの読み取り，（50 ページ）
UUID によるサーバの読み取り，（51 ページ）

アカウント名によるサーバ インベントリの読み取り

目的

アカウント名によってサーバ インベントリを取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCServerInventoryByAccountName/{CIMCServerInventoryByAccountNameId}
```

実装

CIMCServerInventoryByAccountNameId 引数は、Cisco IMC Supervisor によって管理されているサーバの有効なアカウント名でなければなりません。

関連項目

サーバ IP によるサーバ インベントリの読み取り，（54 ページ）

サーバ IP によるサーバ インベントリの読み取り

目的

IP アドレスによってサーバ インベントリを取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCServerInventoryByServerIP/{CIMCServerInventoryByServerIPId}
```

実装

CIMCServerInventoryByServerIPId 引数は、Cisco IMC Supervisor によって管理されているサーバの有効な IP アドレスでなければなりません。IP アドレス内のドットはアンダースコアに置き換える必要があります。

関連項目

アカウント名によるサーバ インベントリの読み取り，（54 ページ）

アカウント名によるサーバ使用率の読み取り

目的

アカウント名によってサーバ使用率を取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCServerUtilizationByAccountName/{CIMCServerUtilizationByAccountNameId}
```

実装

CIMCServerUtilizationByAccountNameId 引数は、Cisco IMC Supervisor によって管理されているサーバの有効なアカウント名でなければなりません。

関連項目

サーバ IP によるサーバ使用率の読み取り，（55 ページ）

サーバ IP によるサーバ使用率の読み取り

目的

IP アドレスによってサーバ使用率を取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCServerUtilizationByServerIP/{CIMCServerUtilizationByServerIPId}
```

実装

CIMCServerUtilizationByServerIPId 引数は、Cisco IMC Supervisor によって管理されているサーバの有効な IP アドレスでなければなりません。IP アドレス内のドットはアンダースコアに置き換える必要があります。

関連項目

アカウント名によるサーバ使用率の読み取り，（55 ページ）

アカウント名によるサーバ使用率履歴の読み取り

目的

アカウント名によってサーバ使用率履歴を取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCServerUtilizationHistoryByAccountName/{CIMCServerUtilizationHistoryByAccountNameI
```

実装

CIMCServerUtilizationHistoryByAccountNameId 引数は、Cisco IMC Supervisor によって管理されているサーバの有効なアカウント名でなければなりません。

関連項目

サーバ IP によるサーバ使用率履歴の読み取り，（56 ページ）

サーバ IP によるサーバ使用率履歴の読み取り

目的

IP アドレスによってサーバ使用率履歴を取得します。

前提条件

なし

REST URL

```
/cloupia/api-v2/CIMCServerUtilizationHistoryByServerIP/{CIMCServerUtilizationHistoryByServerIPId}
```

実装

CIMCServerUtilizationHistoryByServerIPId 引数は、Cisco IMC Supervisor によって管理されているサーバの有効な IP アドレスでなければなりません。IP アドレス内のドットはアンダースコアに置き換える必要があります。

関連項目

アカウント名によるサーバ使用率履歴の読み取り，（56 ページ）

ユーザとグループの管理

概要

このカテゴリの例としては、Cisco IMC Supervisor にアクセスするためのユーザとユーザ グループの管理があります。

ユーザ グループの作成

目的

Cisco IMC Supervisor でユーザのグループを作成します。このタスクでは、関連する一連のユーザを表す新しいグループを作成できます。

前提条件

なし

REST URL

```
/cloupia/api-v2/group
```

コンポーネント

CREATE API のパラメータは次のとおりです。

- 文字列 groupName：グループまたは顧客組織の名前。
- 文字列 groupDescription：オプション。グループまたは顧客組織の説明（必要な場合）。
- 文字列 parentGroup：オプション。親グループの名前。
- 文字列 groupCode：オプション。グループの短い名前またはコード名。
- 文字列 groupContact：グループの連絡先。
- 文字列 firstName：オプション。グループ所有者の名。
- 文字列 lastName：オプション。グループ所有者の姓。
- 文字列 phone：オプション。グループ所有者の電話番号。
- 文字列 address：オプション。グループ所有者の住所。
- 文字列 groupSharePolicyId：オプション。このグループのユーザのグループ共有ポリシーの ID。
- ブール値 allowPrivateUsers：オプション。このオプションは、対象リソースへの排他的アクセス権を持つユーザを作成できます。

入力 XML の例

```
<AddGroupConfig>
<groupName></groupName>

<groupDescription></groupDescription>

<parentGroup></parentGroup>

<groupCode></groupCode>

<groupContact></groupContact>

<firstName></firstName>

<lastName></lastName>

<phone></phone>

<address></address>

<groupSharePolicyId>0</groupSharePolicyId>

<allowPrivateUsers>false</allowPrivateUsers>

</AddGroupConfig>
```

実装

ユーザ グループの名前は必須で、一意にする必要があります。連絡先の電子メールは必須です。

関連項目

ユーザ グループの更新 , （59 ページ）
ユーザ グループの削除, （60 ページ）
グループ内のすべてのユーザの有効化, （61 ページ）
グループ内のすべてのユーザの無効化, （62 ページ）

ユーザ グループの更新

目的

このタスクでは、関連する一連のユーザを表す既存のグループをユーザは更新できます。

前提条件

なし

REST URL

```
/cloupia/api-v2/group
```

コンポーネント

UPDATE API のパラメータは次のとおりです。

- 文字列 groupId：グループまたは顧客組織の ID。
- 文字列 groupDescription：オプション。グループまたは顧客組織の説明（必要な場合）。
- 文字列 parentGroup：オプション。親グループの名前。
- 文字列 groupCode：オプション。グループの短い名前またはコード名。
- 文字列 costCenter：オプション。グループのコスト センターです。
- 文字列 groupContact：グループの連絡先。
- 文字列 firstName：オプション。グループ所有者の名。
- 文字列 lastName：オプション。グループ所有者の姓。
- 文字列 phone：オプション。グループ所有者の電話番号。
- 文字列 address：オプション。グループ所有者の住所。
- 文字列 groupSharePolicyId：オプション。このグループのユーザのグループ共有ポリシーの ID。
- ブール値 allowPrivateUsers：オプション。このオプションは、対象リソースへの排他的アクセス権を持つユーザを作成できます。

入力 XML の例

```
<cuicOperationRequest>
<payload>
<![CDATA[
<ModifyGroupConfig>
<groupId></groupId>

<groupDescription></groupDescription>

<parentGroup></parentGroup>

<groupCode></groupCode>

<costCenter></costCenter>

<groupContact></groupContact>

<firstName></firstName>

<lastName></lastName>

<phone></phone>

<address></address>

<groupSharePolicyId>0</groupSharePolicyId>

<allowPrivateUsers>false</allowPrivateUsers>

</ModifyGroupConfig>

]]>
</payload>
</cuicOperationRequest>
```

実装

名前は変更できません。groupId タグは必須で、有効な既存のグループの数値 ID が含まれている必要があります。連絡先の電子メールは必須です。

関連項目

ユーザ グループの作成, （57 ページ）
ユーザ グループの削除, （60 ページ）
グループ内のすべてのユーザの有効化, （61 ページ）
グループ内のすべてのユーザの無効化, （62 ページ）

ユーザ グループの削除

目的

このタスクでは、関連する一連のユーザを表す既存のグループをユーザは削除できます。

前提条件

なし

REST URL

```
/cloupia/api-v2/group
```

コンポーネント

DELETE_USER API のパラメータは次のとおりです。

文字列 groupName：グループまたは顧客組織の名前。

入力 XML の例

```
<cuicOperationRequest>
<operationType>DELETE_GROUP</operationType>
<payload>
<![CDATA[
<DeleteGroupConfig>
<groupID></groupID>
</DeleteGroupConfig>
]]>
</payload>
</cuicOperationRequest>
```

実装

groupId タグは必須で、有効な既存のグループの数値 ID が含まれている必要があります。

関連項目

ユーザ グループの作成, （57 ページ）
ユーザ グループの更新 , （59 ページ）
グループ内のすべてのユーザの有効化, （61 ページ）
グループ内のすべてのユーザの無効化, （62 ページ）

グループ内のすべてのユーザの有効化

目的

このタスクでは、グループに割り当てられているすべてのユーザを有効にできます。

前提条件

なし

REST URL

```
/cloupia/api-v2/group
```

コンポーネント

ENABLE_ALL_USERS_IN_GROUP API のパラメータは次のとおりです。

文字列 groupName：グループまたは顧客組織の名前。

入力 XML の例

```
<cuicOperationRequest>
<operationType>ENABLE_ALL_USERS_IN_GROUP</operationType>
<payload>
<![CDATA[
<EnableAllUsersInGroupConfig>
<groupID></groupID>

</EnableAllUsersInGroupConfig>

]]>
</payload>
</cuicOperationRequest>
```

実装

groupId タグは必須で、有効な既存のグループの数値 ID が含まれている必要があります。

関連項目

ユーザ グループの作成, （57 ページ）
ユーザ グループの更新 , （59 ページ）
ユーザ グループの削除, （60 ページ）
グループ内のすべてのユーザの無効化, （62 ページ）

グループ内のすべてのユーザの無効化

目的

このタスクでは、グループに割り当てられているすべてのユーザを無効にできます。

前提条件

なし

REST URL

```
/cloupia/api-v2/group
```

コンポーネント

DISABLE_ALL_USERS_IN_GROUP API のパラメータは次のとおりです。

文字列 groupName：グループまたは顧客組織の名前。

入力 XML の例

```
<cuicOperationRequest>
<operationType>DISABLE_ALL_USERS_IN_GROUP</operationType>
<payload>
<![CDATA[
<DisableAllUsersInGroupConfig>
<groupID></groupID>

</DisableAllUsersInGroupConfig>

]]>
</payload>
</cuicOperationRequest>
```

実装

groupId タグは必須で、有効な既存のグループの数値 ID が含まれている必要があります。

関連項目

ユーザ グループの作成, （57 ページ）
ユーザ グループの削除, （60 ページ）
ユーザ グループの更新 , （59 ページ）
グループ内のすべてのユーザの有効化, （61 ページ）

ユーザの作成

目的

このタスクでは、新しいユーザを作成できます。

前提条件

なし

REST URL

```
/cloupia/api-v2/user
```

コンポーネント

CREATE API のパラメータは次のとおりです。

- 文字列 userType：ユーザのタイプ。
- 文字列 userGroup：オプション。ユーザのグループ。
- 文字列 mspOrganization：オプション。MSP 組織のユーザ。
- 文字列 loginName：ユーザのログイン名。
- 文字列 password：ユーザのパスワード。
- 文字列 confirmPassword：前のフィールドと同じパスワードを入力します。
- 文字列 userContactEmail：電子メール アドレス。
- 文字列 firstName：オプション。グループ所有者の名。
- 文字列 lastName：オプション。グループ所有者の姓。
- 文字列 phone：オプション。グループ所有者の電話番号。
- 文字列 address：オプション。グループ所有者の住所。

入力 XML の例

```
<cuicOperationRequest>
<payload>
<![CDATA[
<AddUserConfig>
<userType>GroupAdmin</userType>


<!-- Accepts value from the list: userGroupByType-->
<userGroup>1</userGroup>

<mspOrganization></mspOrganization>

<loginName></loginName>


<!-- Accepts value from the list: password-->
<password></password>


<!-- Accepts value from the list: password-->
<confirmPassword></confirmPassword>

<userContactEmail></userContactEmail>

<firstName></firstName>

<lastName></lastName>

<phone></phone>

<address></address>

</AddUserConfig>

]]>
</payload>
</cuicOperationRequest>
```

実装

ログイン名は必須で、一意にする必要があります。パスワードとパスワードの確認は必須で、両方の値が一致しなければなりません。ユーザの連絡先電子メールは必須です。ユーザ タイプは必須で、既存の有効なユーザ ロールにする必要があります。ユーザ グループ ID は、ユーザ タイプがグループ管理者の場合にのみ必須で、既存のユーザ グループの数値 ID を指定する必要があります。

関連項目

ユーザの読み取り, （66 ページ）

ユーザの更新, （66 ページ）

ユーザの削除, （68 ページ）

ユーザの有効化, （69 ページ）

ユーザの無効化, （70 ページ）

ユーザ失効日の更新, （71 ページ）

ユーザ パスワードの更新, （72 ページ）

ユーザの読み取り

目的

このタスクでは、既存のユーザの詳細に対するクエリーを実行できます。userId 引数は、ユーザの有効なログイン名でなければなりません。引数を指定しないと、結果が返されません。

前提条件

なし

REST URL

```
/cloupia/api-v2/user/{userId}
```

実装

userId引数は、ユーザの有効なログイン名でなければなりません。引数を指定しないと、結果が返されません。

関連項目

ユーザの作成, （63 ページ）
ユーザの更新, （66 ページ）
ユーザの削除, （68 ページ）
ユーザの有効化, （69 ページ）
ユーザの無効化, （70 ページ）
ユーザ失効日の更新, （71 ページ）
ユーザ パスワードの更新, （72 ページ）

ユーザの更新

目的

このタスクでは、既存のユーザを更新できます。

前提条件

なし

REST URL

```
/cloupia/api-v2/user
```

コンポーネント

UPDATE USER API のパラメータは次のとおりです。

- 文字列 loginName：ユーザのログイン名。
- 文字列 userType：ユーザのタイプ。
- 文字列 userGroup：オプション。ユーザのグループ。
- 文字列 mspOrganization：オプション。MSP 組織のユーザ。
- 文字列 userContactEmail：電子メール アドレス。
- 文字列 firstName：オプション。グループ所有者の名。
- 文字列 lastName：オプション。グループ所有者の姓。
- 文字列 phone：オプション。グループ所有者の電話番号。
- 文字列 address：オプション。グループ所有者の住所。

入力 XML の例

```
<cuicOperationRequest>
<operationType>UPDATE_USER</operationType>
<payload>
<![CDATA[
<ModifyUserConfig>
<loginName></loginName>

<userType>GroupAdmin</userType>

<userGroup>1</userGroup>

<mspOrganization></mspOrganization>

<userContactEmail></userContactEmail>

<firstName></firstName>

<lastName></lastName>

<phone></phone>

<address></address>

</ModifyUserConfig>

]]>
</payload>
</cuicOperationRequest>
```

実装

ログイン名は必須で、既存の有効なユーザでなければなりません。これは変更できません。ユーザの連絡先電子メールは必須です。ユーザ タイプは必須で、既存の有効なユーザ ロールにする必要があります。ユーザ グループ ID は、ユーザ タイプがグループ管理者の場合にのみ必須で、既存のユーザ グループの数値 ID を指定する必要があります。

関連項目

ユーザの作成, （63 ページ）
ユーザの読み取り, （66 ページ）
ユーザの削除, （68 ページ）
ユーザの有効化, （69 ページ）
ユーザの無効化, （70 ページ）
ユーザ失効日の更新, （71 ページ）
ユーザ パスワードの更新, （72 ページ）

ユーザの削除

目的

このタスクでは、既存のユーザを削除できます。

前提条件

なし

REST URL

```
/cloupia/api-v2/user
```

コンポーネント

DELETE_USER API のパラメータは次のとおりです。

文字列 loginName：ユーザのログイン名。

入力 XML の例

```
<cuicOperationRequest>
<operationType>DELETE_USER</operationType>
<payload>
<![CDATA[
<DeleteUserConfig>
<loginName></loginName>

</DeleteUserConfig>

]]>
</payload>
</cuicOperationRequest>
```

実装

ログイン名は必須で、既存の有効なユーザでなければなりません。

関連項目

ユーザの作成，（63 ページ）
ユーザの読み取り，（66 ページ）
ユーザの更新，（66 ページ）
ユーザの有効化，（69 ページ）
ユーザの無効化，（70 ページ）
ユーザ失効日の更新，（71 ページ）
ユーザ パスワードの更新，（72 ページ）

ユーザの有効化

目的

このタスクでは、アカウントが無効になっている既存のユーザを有効にできます。

前提条件

なし

REST URL

```
/cloupia/api-v2/user
```

コンポーネント

ENABLE_USER API のパラメータは次のとおりです。

文字列 loginName：ユーザのログイン名。

入力 XML の例

```
<cuicOperationRequest>
<operationType>ENABLE_USER</operationType>
<payload>
<![CDATA[
<EnableUserConfig>
<loginName></loginName>

</EnableUserConfig>

]]>
</payload>
</cuicOperationRequest>
```

実装

ログイン名は必須で、既存の有効なユーザでなければなりません。

関連項目

ユーザの作成, （63 ページ）
ユーザの読み取り, （66 ページ）
ユーザの更新, （66 ページ）
ユーザの削除, （68 ページ）
ユーザの無効化, （70 ページ）
ユーザ失効日の更新, （71 ページ）
ユーザ パスワードの更新, （72 ページ）

ユーザの無効化

目的

このタスクでは、有効なアカウントを持つ既存のユーザを無効にできます。

前提条件

なし

REST URL

```
/cloupia/api-v2/user
```

コンポーネント

DISABLE_USER API のパラメータは次のとおりです。

文字列 loginName：ユーザのログイン名。

入力 XML の例

```
<cuicOperationRequest>
<operationType>DISABLE_USER</operationType>
<payload>
<![CDATA[
<DisableUserConfig>
<loginName></loginName>

</DisableUserConfig>

]]>
</payload>
</cuicOperationRequest>
```

実装

ログイン名は必須で、既存の有効なユーザでなければなりません。

関連項目

ユーザの作成，（63 ページ）
ユーザの読み取り，（66 ページ）
ユーザの更新，（66 ページ）
ユーザの削除，（68 ページ）
ユーザの有効化，（69 ページ）
ユーザ失効日の更新，（71 ページ）
ユーザ パスワードの更新，（72 ページ）

ユーザ失効日の更新

目的

このタスクでは、既存のユーザの失効日を更新できます。

前提条件

なし

REST URL

```
/cloupia/api-v2/user
```

コンポーネント

DISABLE_DATE API のパラメータは次のとおりです。

- 文字列 loginName：ユーザのログイン名。
- Long 型 userExpiryDate：対象ユーザに設定する失効日。

入力 XML の例

```
<cuicOperationRequest>
<operationType>DISABLE_DATE</operationType>
<payload>
<![CDATA[
<ConfigureUserExpiryDateConfig>
<loginName></loginName>


<!-- Accepts value from the list: date_time-->
<userExpiryDate>1460449200000</userExpiryDate>

</ConfigureUserExpiryDateConfig>

]]>
</payload>
</cuicOperationRequest>
```

実装

ログイン名は必須で、既存の有効なユーザでなければなりません。失効日は必須で、失効日/時間のタイムスアンプを示す数値として示す必要があります。

関連項目

ユーザの作成, （63 ページ）
ユーザの読み取り, （66 ページ）
ユーザの更新, （66 ページ）
ユーザの削除, （68 ページ）
ユーザの有効化, （69 ページ）
ユーザの無効化, （70 ページ）
ユーザ パスワードの更新, （72 ページ）

ユーザ パスワードの更新

目的

このタスクでは、既存のユーザ パスワードを更新できます。

前提条件

なし

REST URL

```
/cloupia/api-v2/user
```

コンポーネント

UPDATE_USER_PASSWORD API のパラメータは次のとおりです。

- 文字列 loginName：ユーザのログイン名。
- 文字列 password：ユーザのパスワード。
- 文字列 confirmPassword：前のフィールドと同じパスワードを入力します。

入力 XML の例

```
<cuicOperationRequest>
<operationType>UPDATE_USER_PASSWORD</operationType>
<payload>
<![CDATA[
<AddUserConfig>
<loginName></loginName>


<!-- Accepts value from the list: password-->
<password></password>


<!-- Accepts value from the list: password-->
<confirmPassword></confirmPassword>

</AddUserConfig>

]]>
</payload>
</cuicOperationRequest>
```

実装

ログイン名は必須で、既存の有効なユーザでなければなりません。パスワードとパスワードの確認は必須で、両方の値が一致しなければなりません。

関連項目

ユーザの作成, （63 ページ）
ユーザの読み取り, （66 ページ）
ユーザの更新, （66 ページ）
ユーザの削除, （68 ページ）
ユーザの有効化, （69 ページ）
ユーザの無効化, （70 ページ）
ユーザ失効日の更新, （71 ページ）