



Cisco CIMC XML API メソッドの使用

この章の内容は、次のとおりです。

- [認証方法, 1 ページ](#)
- [クエリーメソッド, 3 ページ](#)

認証方法

認証は CIMC との XML API の交換を可能にします。認証を使用すると、権限を設定し、実行できる操作を制御できます。



(注) このマニュアルのほとんどのコード例では、用語 `<real_cookie>` は 1217377205/85f7ff49-e4ec-42fc-9437-da77a1a2c4bf などの実際の Cookie に置き換えられます。XML API の Cookie は 47 文字の文字列です。これは、セッション情報を維持するために Web ブラウザがローカルに保存する Cookie とは種類が異なります。

ログイン

ログインするために、XML API クライアントは CIMC HTTP（または HTTPS）サーバへの TCP 接続を確立し、`aaaLogin` メソッドを格納している XML ドキュメントをポストします。

次の例では、IP アドレス 192.0.20.72 で CIMC のポート 80 への TCP 接続を確立するために、Telnet ユーティリティを使用しています。使用するパスは `/nuova` です。

```
$ telnet 192.0.20.72 80
POST /nuova HTTP/1.1
USER-Agent: lwp-request/2.06
HOST: 192.0.20.72
Content-Length: 62
Content-Type: application/x-www-form-urlencoded
```

次に、クライアントは `aaaLogin` メソッドを指定し、ユーザ名とパスワードを提供します。

```
<aaaLogin
  inName='admin'
  inPassword='password'>
</aaaLogin>
```



(注) XML API ドキュメントに XML バージョンまたは DOCTYPE の行を含めないでください。
`inName` 属性と `inPassword` 属性はパラメータです。

各 XML API ドキュメントは、実行する操作を表します。要求が XML API ドキュメントとして受け取られると、CIMC は要求を読み取り、メソッドで指定されているアクションを実行します。CIMC は、XML ドキュメント形式のメッセージで応答し、要求の成否を示します。

次に、一般的な成功応答を示します。

```
1 <aaaLogin
2   response="yes"
3   outCookie="<real_cookie>"
4   outRefreshPeriod="600"
5   outPriv="admin">
6 </aaaLogin>
```

応答の各行は、次のように解釈されます。

- 1 ログインに使用するメソッドを指定します。
- 2 これが応答であることを確認します。
- 3 セッションの Cookie を提供します。
- 4 推奨される Cookie のリフレッシュ間隔を指定します。デフォルトのログインセッションの長さは 600 秒です。
- 5 ユーザアカウントに割り当てられる権限レベルを指定します (`admin`、`user`、または `readonly`) 。
- 6 終了タグ。

また、次の例に示すように XML API へのログインに cURL ユーティリティを使用できます。

```
curl -d "<aaaLogin inName='admin' inPassword='password'></aaaLogin>" http://192.0.20.72/nuova
```

HTTPS を有効にした場合は、次の例に示すように、cURL のコマンドで HTTPS を使用する必要があります。

```
curl -d "<aaaLogin inName='admin' inPassword='password'></aaaLogin>" https://192.0.20.72/nuova
```

セッションの更新

セッションは、`aaaLogin` 応答または以前の更新から取得された 47 文字の Cookie を使用して、`aaaRefresh` メソッドで更新されます。

```
<aaaRefresh
  cookie="<real_cookie>"
  inCookie="<real_cookie>"
  inName='admin'
```

```
    inPassword='password'>
  </aaaRefresh>
```

セッションからのログアウト

セッションからログアウトするには、次のメソッドを使用します。

```
<aaaLogout
  cookie="<real_cookie>"
  inCookie="<real_cookie>"
</aaaLogout>
```

失敗応答の例

失敗したログイン：

```
<aaaLogin
  cookie=""
  response="yes"
  errorCode="551"
  invocationResult="unidentified-fail"
  errorDescr="Authentication failed">
</aaaLogin>
```

存在しないオブジェクト（空白が返される場合は、指定した DN オブジェクトが存在しないことを示します）：

```
<configResolveDn
  cookie="<real_cookie>"
  response="yes"
  dn="sys/rack-unit-1/adaptor-9999">
  <outConfig>
  </outConfig>
</configResolveDn>
```

不正な要求：

```
<configConfMo
  cookie="<real_cookie>"
  response="yes"
  dn="sys/rack-unit-1/adaptor-1/ext-eth-0">
  errorCode="103"
  invocationResult="unidentified-fail"
  errorDescr="can't create; object already exists.">
</configConfMo>
```

クエリーメソッド

configResolveChildren の使用

管理情報ツリーの子オブジェクトの解決時に、次のことに注意してください。

configResolveClass の使用

- このメソッドが、名前付きクラスのインスタンスである名前付きオブジェクトのすべての子オブジェクトを取得している。クラス名を省略すると、名前付きオブジェクトのすべての子オブジェクトが返されます。
- `inDn` 属性により、子オブジェクトが取得される名前付きオブジェクトが指定されている（必須）。
- `classId` 属性により、返される子オブジェクト クラスの名前が指定されている（任意）。
- 認証 Cookie（`aaaLogin` または `aaaRefresh` から）が必須である。
- `inHierarchical` 属性（デフォルト値は `false`）が `true` の場合は、結果が階層形式であることが指定されている。
- 列挙値、`classIds`、およびビット マスクが文字列として表示されている。

[configResolveChildren](#) に示す要求または応答の例を参照してください。

configResolveClass の使用

クラスの解決時には、次のことに注意してください。

- 指定したクラス タイプのすべてのオブジェクトが取得されている。
- `classId` が、返されるオブジェクト クラスの名前を指定している（必須）。
- 認証 Cookie（`aaaLogin` または `aaaRefresh` から）が必須である。
- `inHierarchical` 属性（デフォルト値は `false`）が `true` の場合は、結果が階層形式であることが指定されている。
- 列挙値、`classIds`、およびビット マスクが文字列として表示されている。

結果セットは大きくなることがあります。結果セットは正確に定義してください。たとえば、アダプタのリストだけを取得する場合は、クエリーで `classId` の属性値として `adaptorUnit` を使用します。この例は、`adaptorUnit` クラスのすべてのインスタンスについて問い合わせます。

```
<configResolveClass
  cookie="real_cookie"
  inHierarchical="false"
  classId="adaptorUnit"/>
```

[configResolveClass](#) に示す要求または応答の例を参照してください。

configResolveDn の使用

DN を解決するときは、次のことに注意してください。

- DN により指定されたオブジェクトが取得されている。
- 指定された DN が、解決するオブジェクト インスタンスを識別している（必須）。

- 認証 Cookie (aaaLogin または aaaRefresh から) が必須である。
- inHierarchical 属性 (デフォルト値は false) が true の場合は、結果が階層形式であることが指定されている。
- 列挙値、classIds、およびビット マスクが文字列として表示されている。

[configResolveDn](#)に示す要求または応答の例を参照してください。

configResolveParent の使用

オブジェクトの親オブジェクトの解決時に、次のことに注意してください。

- このメソッドが、指定 DN の親オブジェクトを取得している。
- dn 属性が子オブジェクトの DN である (必須)。
- 認証 Cookie (aaaLogin または aaaRefresh から) が必須である。
- inHierarchical 属性 (デフォルト値は false) が true の場合は、結果が階層形式であることが指定されている。
- 列挙値、classIds、およびビット マスクが文字列として表示されている。

[configResolveParent](#)に示す要求または応答の例を参照してください。

