



外部 RESTful サービス コール

- 「外部 RESTful サービス API コールの呼び出し」 (P.6-1)
- 「GetVersion API コール」 (P.6-1)
- 「内部ユーザの外部 RESTful サービス API コール」 (P.6-2)
- 「エンドポイントの外部 RESTful サービス API コール」 (P.6-9)
- 「エンドポイント ID グループの外部 RESTful サービス API コール」 (P.6-16)
- 「ID グループの外部 RESTful サービス API コール」 (P.6-25)
- 「セキュリティ グループ タグの外部 RESTful サービス API コール」 (P.6-27)
- 「REST API クライアントの使用方法」 (P.6-29)
- 「外部 RESTful サービス Java デモ アプリケーション」 (P.6-38)

外部 RESTful サービス API コールの呼び出し

外部 RESTful サービス API コールを呼び出す前に、次の前提条件を満たす必要があります。

- CLI から外部 RESTful サービスをイネーブルにします。
- 外部 RESTful サービス管理者権限をイネーブルにします。

外部 RESTful サービス API コールを呼び出すために、JAVA、cURL Linux コマンド、Python などの REST クライアントを使用できます。

関連項目

- 「外部 RESTful API サービスのイネーブル化」 (P.5-2)
- 「外部 RESTful サービス許可」 (P.5-15)

GetVersion API コール

GetVersion API コールは、使用可能なすべてのリソース（エンドポイント、内部ユーザ、エンドポイント ID グループ、およびセキュリティ グループ タグ (SGT)）に共通です。これは、必要なリソースのバージョン情報を取得します。

表 6-1 GetVersion の特性

特性	説明
説明	指定したリソースのバージョン情報を取得します
概要	GET/ers/config/<resource-name>/version
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	—
要求メッセージ本文	—
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	VersionInfo タイプのリソース
応答ステータス	200、400、401、403、404、415、500

GetVersion 要求のサンプル

```
Sample Request for Get Version Internal Users API
Method:PUT
URI: https://10.56.13.196:9060/ers/config/internaluser/versioninfo
HTTP Accept header:
application/vnd.com.cisco.ise.identity.internaluser.1.0+xml
```

GetVersion 応答のサンプル

```
HTTP Status: 200 (OK)
Content:
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:versionInfo xmlns:ns2="ers.ise.cisco.com">
  <currentServerVersion>1.0</currentServerVersion>
  <link type="application/xml" href="link" rel="self"/>
  <supportedVersions>0.9,0.8</supportedVersions>
</ns2:versionInfo>
```

内部ユーザの外部 RESTful サービス API コール

内部ユーザの外部 RESTful サービス API コールは、Create、Read、Update、Delete (CRUD) 機能を完全にサポートします。これらの API コールで使用される内部ユーザ ID は Cisco ISE データベースに保存された UUID タイプです。

表 6-2 内部ユーザに対して使用できる外部 RESTful サービス API コール

API コール	HTTP メソッド	URL	内容	クエリー文字列
すべてのユーザの取得	GET	/ers/config/internaluser	—	Page、Size、sortacs または sortdsn、Filter
ユーザの取得	GET	/ers/config/internaluser/{internaluser-id}	—	—
ユーザの作成	POST	/ers/config/internaluser/	internaluser	—
ユーザの更新	PUT	/ers/config/internaluser/{internaluser-id}	internaluser	—
ユーザの削除	DELETE	/ers/config/internaluser/{internaluser-id}	—	—

内部ユーザ スキーマ

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:internaluser name="name" id="id" description="description"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
  <changePassword>true</changePassword>
  <customAttributes>
    <entry>
      <key>key2</key>
      <value>value3</value>
    </entry>
    <entry>
      <key>key1</key>
      <value>value1</value>
    </entry>
  </customAttributes>
  <email>email@domain.com</email>
  <enabled>true</enabled>
  <firstName>firstName</firstName>
  <identityGroups>identityGroups</identityGroups>
  <lastName>lastName</lastName>
  <password>password</password>
</ns3:internaluser>
```

Get All Users API コール

Cisco ISE のすべての内部ユーザを取得するには、Get All Users API コールを使用します。

表 6-3 Get All Users の特性

特性	説明
説明	内部ユーザの集合を取得します
概要	GET /ers/config/internaluser
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	start、size、sortByacn、sortBydcn、filter (フィルタでサポートされるフィールド: name、description、firstName、lastName、identityGroup、email、enabled)
要求メッセージ本文	—
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	SearchResult
応答ステータス	200、400、401、403、404、415、500

すべてのユーザの取得要求のサンプル

```
Method: GET
URI: https://10.56.13.196:9060/ers/config/internaluser
HTTP Accept header:
application/vnd.com.cisco.ise.identity.internaluser.1.0+xml
Request Content:
N/A
```

すべてのユーザの取得応答のサンプル

```
HTTP Status: 200 (OK)
Content:
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult total="2" xmlns:ns2="ers.ise.cisco.com">
  <nextPage type="application/xml" href="link-to-next-page" rel="next"/>
  <previousPage type="application/xml" href="link-to-previous-page" rel="previous"/>
  <resources>
    <resource name="name1" id="id1" description="description1"/>
    <resource name="name2" id="id2" description="description2"/>
  </resources>
</ns2:searchResult>
```

Get User API コール

ユーザ ID を使用して内部ユーザを取得するには、Get User API コールを使用します。

表 6-4 Get User の特性

特性	説明
説明	指定された内部ユーザを取得します
概要	GET /ers/config/internaluser/{internaluser-id}
要求ヘッダー	Accept、Authorization、Host

表 6-4 Get User の特性 (続き)

特性	説明
クエリー文字列	—
要求メッセージ本文	—
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	InternalUser タイプのリソース
応答ステータス	200、400、401、403、404、415、500

ユーザの取得要求のサンプル

```
Method: GET
URI: https://10.56.13.196:9060/ers/config/internaluser/{id}
HTTP Accept header:
application/vnd.com.cisco.ise.identity.internaluser.1.0+xml
Request Content:
N/A
```

ユーザの取得応答のサンプル

```
HTTP Status: 200 (OK)
Content:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:internaluser name="name" id="id" description="description"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
  <changePassword>true</changePassword>
  <customAttributes>
    <entry>
      <key>key2</key>
      <value>value3</value>
    </entry>
    <entry>
      <key>key1</key>
      <value>value1</value>
    </entry>
  </customAttributes>
  <email>email@domain.com</email>
  <enabled>true</enabled>
  <firstName>firstName</firstName>
  <identityGroups>identityGroups</identityGroups>
  <lastName>lastName</lastName>
  <password>password</password>
</ns3:internaluser>
```

Create User API コール

Cisco ISE の内部ユーザを作成するには、Create User API コールを使用します。外部 RESTful サービス API を使用して内部ユーザを作成するのに、パスワードは必須です。

表 6-5 Create User API コールの特性

特性	説明
説明	指定された内部ユーザを作成します
概要	POST /ers/config/internaluser/
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	—
要求メッセージ本文	InternalUser
応答ヘッダー	Content-Length、Content-Type、Location
応答メッセージ本文	InternalUser タイプのリソース
応答ステータス	201、400、401、403、415、500

ユーザの作成要求のサンプル

```
Method: POST
URI: https://10.56.13.196:9060/ers/config/internaluser
HTTP Content-Type header:
application/vnd.com.cisco.ise.identity.internaluser.1.0+xml; charset=utf-8
Request Content:
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:internaluser name="name" id="id" description="description"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
  <changePassword>true</changePassword>
  <customAttributes>
    <entry>
      <key>key2</key>
      <value>value3</value>
    </entry>
    <entry>
      <key>key1</key>
      <value>value1</value>
    </entry>
  </customAttributes>
  <email>email@domain.com</email>
  <enabled>true</enabled>
  <firstName>firstName</firstName>
  <identityGroups>identityGroups</identityGroups>
  <lastName>lastName</lastName>
  <password>password</password>
</ns3:internaluser>
```

ユーザの作成応答のサンプル

```
HTTP Status: 201 (Created)
Content:
N/A
```

Update User API コール

Cisco ISE の内部ユーザを更新するには、Update User API コールを使用します。外部 RESTful サービス API を使用して内部ユーザを更新しているときに、既存のパスワードを変更しない場合は、現在のパスワードを入力する必要はありません。内部ユーザを更新するときに既存のパスワードを変更する場合は、プレーンテキストで指定できます。

表 6-6 Update User の特性

特性	説明
説明	指定された内部ユーザを更新します
概要	PUT /ers/config/internaluser/{internaluser-id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	—
要求メッセージ本文	InternalUser
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	更新されたフィールドのリスト
応答ステータス	200、400、401、403、404、415、500

ユーザの更新要求のサンプル

```
Method: PUT
URI: https://10.56.13.196:9060/ers/config/internaluser/{id}
HTTP Content-Type header:
application/vnd.com.cisco.ise.identity.internaluser.1.0+xml; charset=utf-8
Request Content:
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:internaluser name="name" id="id" description="description"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
  <changePassword>true</changePassword>
  <customAttributes>
    <entry>
      <key>key2</key>
      <value>value3</value>
    </entry>
    <entry>
      <key>key1</key>
      <value>value1</value>
    </entry>
  </customAttributes>
  <email>email@domain.com</email>
  <enabled>true</enabled>
  <firstName>firstName</firstName>
  <identityGroups>identityGroups</identityGroups>
  <lastName>lastName</lastName>
  <password>password</password>
</ns3:internaluser>
```

ユーザの更新応答のサンプル

HTTP Status: 200 (OK)
Content:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:updatedFields xmlns:ns2="ers.ise.cisco.com">
  <updatedField field="field1">
    <newValue>val_new</newValue>
    <oldValue>val_old</oldValue>
  </updatedField>
  <updatedField field="some other field">
    <newValue>val_new</newValue>
    <oldValue>val_old</oldValue>
  </updatedField>
</ns2:updatedFields>
```

Delete User API コール

Cisco ISE から内部ユーザを削除するには、Delete User API コールを使用します。

表 6-7 Delete User の特性

特性	説明
説明	指定された内部ユーザを削除します
概要	DELETE /ers/config/internaluser/{internaluser-id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	—
要求メッセージ本文	—
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	InternalUser タイプのリソース
応答ステータス	204、400、401、403、404、415、500

ユーザの削除要求のサンプル

```
Method: DELETE
URI: https://10.56.13.196:9060/ers/config/internaluser/{id}
HTTP Accept header:
application/vnd.com.cisco.ise.identity.internaluser.1.0+xml
Request Content:
N/A
```

ユーザの削除応答のサンプル

HTTP Status: 204 (No Content)
Content:
N/A

エンドポイントの外部 RESTful サービス API コール

これらの API コールで使用される内部ユーザ ID は Cisco ISE データベースに保存された UUID タイプです。



(注)

これらの例は、DB データへの参照があるため、そのまま使用するためのものではありません。例を基本テンプレートとして処理し、サーバに送信する前に編集する必要があります。

表 6-8 エンドポイントに対して使用できる外部 RESTful サービス API コール

オペレーション	メソッド	URL	内容	クエリー文字列
すべてのエンドポイントの取得	GET	/ers/config/endpoint	—	page、size、sortacs または sortdsn、filter
エンドポイントの取得	GET	/ers/config/endpoint/{endpoint-id}	—	—
エンドポイントの作成	POST	/ers/config/endpoint/	endpoint	—
エンドポイントの更新	PUT	/ers/config/endpoint/{endpoint-id}	endpoint	—
エンドポイントの削除	DELETE	/ers/config/endpoint/{endpoint-id}	—	—
エンドポイントの登録	PUT	/ers/config/endpoint/register	endpoint	—
エンドポイントの登録解除	PUT	/ers/config/endpoint/{endpoint-id}/deregister	—	—

エンドポイントのスキーマ

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:endpoint name="name" id="id" description="description"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
  <groupId>groupId</groupId>
  <identityStore>identityStore</identityStore>
  <identityStoreId>identityStoreId</identityStoreId>
  <mac>00:01:02:03:04:05</mac>
  <portalUser>portalUser</portalUser>
  <profileId>profileId</profileId>
  <staticGroupAssignment>true</staticGroupAssignment>
  <staticProfileAssignment>false</staticProfileAssignment>
</ns3:endpoint>
```

Get All Endpoints API コール

Get All Endpoints API コールを使用して、エンドポイントのリストを取得します。

表 6-9 Get All Endpoints の特性

特性	説明
説明	エンドポイントの集合を取得します
概要	GET ers/config/endpoint
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	start、size、sortByacn、sortBydcn、filter (フィルタでサポートされるフィールド: mac、portalUser、profileId、staticGroupAssignment、staticProfileAssignment)
要求メッセージ本文	—
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	SearchResult
応答ステータス	200、400、401、403、404、415、500

すべてのエンドポイントの取得要求のサンプル

```
Method: GET
URI: https://10.56.13.196:9060/ers/config/endpoint
HTTP Accept header:
application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
Request Content:
N/A
```

すべてのエンドポイントの取得応答のサンプル

```
HTTP Status: 200 (OK)
Content:
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult total="2" xmlns:ns2="ers.ise.cisco.com">
  <nextPage type="application/xml" href="link-to-next-page" rel="next"/>
  <previousPage type="application/xml" href="link-to-previous-page" rel="previous"/>
  <resources>
    <resource name="name1" id="id1" description="description1"/>
    <resource name="name2" id="id2" description="description2"/>
  </resources>
</ns2:searchResult>
```



(注)

サポートされていないため、エンドポイントに、完全なオブジェクトに対するハイパーリンクはありません。名前の説明と ID だけが表示されます。

Get Endpoint API コール

ユーザ ID を使用してエンドポイントを取得するには、Get Endpoint API コールを使用します。

表 6-10 Get Endpoint の特性

特性	説明
説明	指定されたエンドポイントを取得します
概要	GET /ers/config/endpoint/{endpoint-id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	—
要求メッセージ本文	—
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	InternalUser タイプのリソース
応答ステータス	200、400、401、403、404、415、500

エンドポイントの取得要求のサンプル

```
Method: GET
URI: https://10.56.13.196:9060/ers/config/endpoint/{id}
HTTP Accept header:
application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
Request Content:
N/A
```

エンドポイントの取得応答のサンプル

```
HTTP Status: 200 (OK)
Content:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:endpoint name="name" id="id" description="description" xmlns:ns2="ers.ise.cisco.com"
xmlns:ns3="identity.ers.ise.cisco.com">
  <groupId>groupId</groupId>
  <identityStore>identityStore</identityStore>
  <identityStoreId>identityStoreId</identityStoreId>
  <mac>00:01:02:03:04:05</mac>
  <portalUser>portalUser</portalUser>
  <profileId>profileId</profileId>
  <staticGroupAssignment>true</staticGroupAssignment>
  <staticProfileAssignment>>false</staticProfileAssignment>
</ns3:endpoint>
```

Create Endpoint API コール

Cisco ISE のエンドポイントを作成するには、Create Endpoint API コールを使用します。

表 6-11 Create Endpoint の特性

特性	説明
説明	指定されたエンドポイントを作成します
概要	POST /ers/config/endpoint

表 6-11 Create Endpoint の特性 (続き)

特性	説明
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	—
要求メッセージ本文	エンドポイント
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	InternalUser タイプのリソース
応答ステータス	200、400、401、403、404、415、500

エンドポイントの作成要求のサンプル

```
Method: POST
URI: https://10.56.13.196:9060/ers/config/endpoint
HTTP Content-Type header:
application/vnd.com.cisco.ise.identity.endpoint.1.0+xml; charset=utf-8
Request Content:
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:endpoint name="name" id="id" description="description"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
  <groupId>groupId</groupId>
  <identityStore>identityStore</identityStore>
  <identityStoreId>identityStoreId</identityStoreId>
  <mac>00:01:02:03:04:05</mac>
  <portalUser>portalUser</portalUser>
  <profileId>profileId</profileId>
  <staticGroupAssignment>true</staticGroupAssignment>
  <staticProfileAssignment>>false</staticProfileAssignment>
</ns3:endpoint>
```

エンドポイントの作成応答のサンプル

```
HTTP Status: 201 (Created)
Content:
N/A
```

Update Endpoint API コール

Cisco ISE のエンドポイントを更新するには、Update Endpoint API コールを使用します。

表 6-12 Update Endpoint の特性

特性	説明
説明	指定されたエンドポイントを更新します
概要	PUT /ers/config/endpoint/{endpoint-id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	—
要求メッセージ本文	エンドポイント
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	更新されたフィールドのリスト
応答ステータス	200、400、401、403、404、415、500

エンドポイントの更新要求のサンプル

```
Method: PUT
URI: https://10.56.13.196:9060/ers/config/endpoint/{id}
HTTP Content-Type header:
application/vnd.com.cisco.ise.identity.endpoint.1.0+xml; charset=utf-8
Request Content:
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:endpoint name="name" id="id" description="description" xmlns:ns2="ers.ise.cisco.com"
xmlns:ns3="identity.ers.ise.cisco.com">
  <groupId>groupId</groupId>
  <identityStore>identityStore</identityStore>
  <identityStoreId>identityStoreId</identityStoreId>
  <mac>00:01:02:03:04:05</mac>
  <portalUser>portalUser</portalUser>
  <profileId>profileId</profileId>
  <staticGroupAssignment>true</staticGroupAssignment>
  <staticProfileAssignment>>false</staticProfileAssignment>
</ns3:endpoint>
```

エンドポイントの更新応答のサンプル

```
HTTP Status: 200 (OK)
Content:
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:updatedFields xmlns:ns2="ers.ise.cisco.com">
  <updatedField field="field1">
    <newValue>val_new</newValue>
    <oldValue>val_old</oldValue>
  </updatedField>
  <updatedField field="some other field">
    <newValue>val_new</newValue>
    <oldValue>val_old</oldValue>
  </updatedField>
</ns2:updatedFields>
```

Delete Endpoint API コール

Cisco ISE のエンドポイントを削除するには、Delete Endpoint API コールを使用します。

表 6-13 Delete Endpoint の特性

特性	説明
説明	指定されたエンドポイントを削除します
概要	DELETE /ers/config/endpoint/{endpoint-id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	—
要求メッセージ本文	—
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	—
応答ステータス	200、400、401、403、404、415、500

エンドポイントの削除要求のサンプル

```
Method: DELETE
URI: https://10.56.13.196:9060/ers/config/endpoint/{id}
HTTP Accept header:
application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
Request Content:
N/A
```

エンドポイントの削除応答のサンプル

```
HTTP Status: 204 (No Content)
Content:
N/A
```

Register Endpoint API コール

Cisco ISE のエンドポイントを登録するには、Register Endpoint API コールを使用します。エンドポイントがすでに存在する場合は、登録されます。存在しない場合は、作成され、登録されます。いずれのシナリオでも、リターン ステータスは、no content を意味する 204 になります。GUI 登録フローと同様に、エンドポイントは登録済みデバイス グループに静的に割り当てられ、ポータル ユーザおよび ID ストアはコンテンツで指定されているとおりに設定されます。

表 6-14 Register Endpoint の特性

特性	説明
説明	指定されたエンドポイントを登録します
概要	PUT /ers/config/endpoint/register
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	—
要求メッセージ本文	endpoint
応答ヘッダー	Content-Length、Content-Type

表 6-14 Register Endpoint の特性 (続き)

特性	説明
応答メッセージ本文	更新されたフィールドのリスト
応答ステータス	202、400、401、403、404、415、500

エンドポイントの登録要求のサンプル

```
Method: PUT
URI: https://10.56.13.196:9060/ers/config/endpoint/register
HTTP Content-Type header:
application/vnd.com.cisco.ise.identity.endpoint.1.0+xml; charset=utf-8
Request Content:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:endpoint name="name" id="id" description="description" xmlns:ns2="ers.ise.cisco.com"
xmlns:ns3="identity.ers.ise.cisco.com">
  <groupId>groupId</groupId>
  <identityStore>identityStore</identityStore>
  <identityStoreId>identityStoreId</identityStoreId>
  <mac>00:01:02:03:04:05</mac>
  <portalUser>portalUser</portalUser>
  <profileId>profileId</profileId>
  <staticGroupAssignment>true</staticGroupAssignment>
  <staticProfileAssignment>false</staticProfileAssignment>
</ns3:endpoint>
```

エンドポイントの登録応答のサンプル

```
HTTP Status: null
Content:
N/A
```

Deregister Endpoint API コール

Cisco ISE のエンドポイントを登録解除するには、Deregister Endpoint API コールを使用します。

表 6-15 Deregister Endpoint の特性

特性	説明
説明	指定されたエンドポイントを登録解除します
概要	PUT /ers/config/endpoint/{endpoint-id}/deregister
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	—
要求メッセージ本文	—
応答メッセージ本文	—
応答ステータス	202、400、401、403、404、415、500

エンドポイントの登録解除要求のサンプル

```
Method: PUT
URI: https://10.56.13.196:9060/ers/config/endpoint/{id}/deregister
HTTP Content-Type header:
application/vnd.com.cisco.ise.identity.endpoint.1.0+xml; charset=utf-8
Request Content:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:endpoint name="name" id="id" description="description" xmlns:ns2="ers.ise.cisco.com"
xmlns:ns3="identity.ers.ise.cisco.com">
  <groupId>groupId</groupId>
  <identityStore>identityStore</identityStore>
  <identityStoreId>identityStoreId</identityStoreId>
  <mac>00:01:02:03:04:05</mac>
  <portalUser>portalUser</portalUser>
  <profileId>profileId</profileId>
  <staticGroupAssignment>true</staticGroupAssignment>
  <staticProfileAssignment>false</staticProfileAssignment>
</ns3:endpoint>
```

エンドポイントの登録解除応答のサンプル

```
HTTP Status: null
Content:
N/A
```

エンドポイント ID グループの外部 RESTful サービス API コール

これらの API コールで使用される内部ユーザ ID は Cisco ISE データベースに保存された UUID タイプです。



(注)

後述の項に示されている例は、DB データへの参照があるため、そのまま使用するためのものではありません。例を基本テンプレートとして処理し、サーバに送信する前に編集する必要があります。

表 6-16 エンドポイント ID グループに対して使用できる外部 RESTful サービス API コール

オペレーション	メソッド	URL	内容	クエリー文字列
すべてのエンドポイントグループの取得	GET	/ers/config/endpointgroup	—	page、size、sortacs または sortdsn、filter
エンドポイントグループの取得	GET	/ers/config/endpointgroup/{endpointgroup-id}	—	—

表 6-16 エンドポイント ID グループに対して使用できる外部 RESTful サービス API コール (続き)

オペレーション	メソッド	URL	内容	クエリー文字列
エンドポイント グループの作成	POST	/ers/config/endpointgroup/	Endpointgroup	—
エンドポイント グループの更新	PUT	/ers/config/endpointgroup/{endpointgroup-id}	Endpointgroup	—
エンドポイント グループの削除	DELETE	/ers/config/endpointgroup/{endpointgroup-id}	—	—

エンドポイント ID グループのスキーマ

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:endpointgroup name="name" id="id" description="description"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
  <systemDefined>true</systemDefined>
</ns3:endpointgroup>
```

Get All Endpoint Identity Groups API コール

Get All Endpoint Identity Groups API コールを使用して、エンドポイント ID グループの集合を取得します。

表 6-17 Get All Endpoint Identity Groups の特性

特性	説明
説明	エンドポイント グループの集合を取得します
概要	GET /ers/config/endpointgroup
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	start、size、sortByacn、sortBydcn、filter
要求メッセージ本文	—
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	SearchResult
応答ステータス	202、400、401、403、404、415、500

すべてのエンドポイント ID グループの取得要求のサンプル

```
Method: GET
URI: https://10.56.13.196:9060/ers/config/endpointgroup
HTTP Accept header:
application/vnd.com.cisco.ise.identity.endpointgroup.1.0+xml
Request Content:
N/A
```

すべてのエンドポイント ID グループの取得応答のサンプル

HTTP Status: 200 (OK)
Content:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult total="2" xmlns:ns2="ers.ise.cisco.com">
  <nextPage type="application/xml" href="link-to-next-page" rel="next"/>
  <previousPage type="application/xml" href="link-to-previous-page" rel="previous"/>
  <resources>
    <resource name="name1" id="id1" description="description1"/>
    <resource name="name2" id="id2" description="description2"/>
  </resources>
</ns2:searchResult>
```



(注)

エンドポイントの完全なオブジェクトに対するハイパーリンクはありません。名前、説明、および ID だけが表示されます。

Get Endpoint Identity Group API コール

Get Endpoint Identity Group API コールを使用して、指定されたエンドポイント ID グループを取得します。

表 6-18 Get Endpoint Identity Group の特性

特性	説明
説明	指定されたエンドポイント グループを取得します
概要	GET /ers/config/endpoint/{endpointgroup-id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	—
要求メッセージ本文	—
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	Endpoint タイプのリソース
応答ステータス	200、400、401、403、404、415、500

エンドポイント ID グループの取得要求のサンプル

Method: GET
URI: https://10.56.13.196:9060/ers/config/endpointgroup/{id}
HTTP Accept header:
application/vnd.com.cisco.ise.identity.endpointgroup.1.0+xml
Request Content:
N/A

エンドポイント ID グループの取得応答のサンプル

HTTP Status: 200 (OK)

Content:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:endpointgroup name="name" id="id" description="description"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
  <systemDefined>true</systemDefined>
</ns3:endpointgroup>
}
```

Create Endpoint Identity Group API コール

Create Endpoint Identity Group API コールを使用して、指定されたエンドポイント ID グループを作成します。

表 6-19 Create Endpoint Identity Group の特性

特性	説明
説明	指定されたエンドポイント グループを作成します
概要	POST /ers/config/endpoint/
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	—
要求メッセージ本文	Endpoint Group
応答ヘッダー	Content-Length、Content-Type、Location
応答メッセージ本文	Endpoint Group タイプのリソース
応答ステータス	201、400、401、403、415、500

エンドポイント ID グループの作成要求のサンプル

```
Method: POST
URI: https://10.56.13.196:9060/ers/config/endpointgroup
HTTP Content-Type header:
application/vnd.com.cisco.ise.identity.endpointgroup.1.0+xml; charset=utf-8
Request Content:
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:endpointgroup name="name" id="id" description="description"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
  <systemDefined>true</systemDefined>
</ns3:endpointgroup>
```

エンドポイント ID グループの作成応答のサンプル

```
HTTP Status: 201 (Created)
Content:
N/A
```

Update Endpoint Identity Group API コール

Update Endpoint Identity Group API コールを使用して、指定されたエンドポイント ID グループを更新します。

表 6-20 Update Endpoint Identity Group の特性

特性	説明
説明	指定されたエンドポイント グループを更新します
概要	PUT /ers/config/endpoint/{endpointgroup-id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	—
要求メッセージ本文	Endpoint Group
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	更新されたフィールドのリスト
応答ステータス	200、400、401、403、404、415、500

エンドポイント ID グループの更新要求のサンプル

```
Method: PUT
URI: https://10.56.13.196:9060/ers/config/endpointgroup/{id}
HTTP Content-Type header:
application/vnd.com.cisco.ise.identity.endpointgroup.1.0+xml; charset=utf-8
Request Content:
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:endpointgroup name="name" id="id" description="description"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
  <systemDefined>true</systemDefined>
</ns3:endpointgroup>
```

エンドポイント ID グループの更新応答のサンプル

```
HTTP Status: 200 (OK)
Content:
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:updatedFields xmlns:ns2="ers.ise.cisco.com">
  <updatedField field="field1">
    <newValue>val_new</newValue>
    <oldValue>val_old</oldValue>
  </updatedField>
  <updatedField field="some other field">
    <newValue>val_new</newValue>
    <oldValue>val_old</oldValue>
  </updatedField>
</ns2:updatedFields>
}
```

Delete Endpoint Identity Group API コール

Delete Endpoint Identity Group API コールを使用して、指定されたエンドポイント ID グループを削除します。

表 6-21 Delete Endpoint Identity Group の特性

特性	説明
説明	指定されたエンドポイント グループを削除します
概要	DELETE /ers/config/endpointgroup/{endpointgroup-id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	—
要求メッセージ本文	—
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	—
応答ステータス	204、400、401、403、404、415、500

エンドポイント ID グループの削除要求のサンプル

```
Method: DELETE
URI: https://10.56.13.196:9060/ers/config/endpointgroup/{id}
HTTP Accept header:
application/vnd.com.cisco.ise.identity.endpointgroup.1.0+xml
Request Content:
N/A
```

エンドポイント ID グループの削除応答のサンプル

```
HTTP Status: 204 (No Content)
Content:
N/A
```

Profiler プロファイルの外部 RESTful サービス API コール

Profiler プロファイル API コールを使用すると、プロファイルを検索することができます。Profiler プロファイルは、エンドポイントが分類されるか、または静的に割り当てられるプロファイルです。Profiler プロファイルは、プロファイラ ポリシーとも呼ばれます。プロファイル ID はエンドポイントの属性です。エンドポイントは名前でもフィルタリングできます。



(注) 次の項に示されている例は、DB データへの参照があるため、そのまま使用するためのものではありません。例を基本テンプレートとして処理し、サーバに送信する前に編集する必要があります。

表 6-22 Profiler プロファイルに対して使用できる外部 RESTful サービス API コール

オペレーション	メソッド	URL	内容	クエリー文字列
すべてのプロファイルの取得	GET	/ers/config/profilerprofile	—	Start サイズ (Size) sortacs または sortdsn filter
プロファイルの取得 (ID で取得)	GET	/ers/config/profilerprofile/{id}	—	—
プロファイルの作成	POST	ers/config/profilerprofile	—	—

Profiler プロファイルのスキーマ

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:profilerprofile name="name" id="id" description="description"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com"/>
```

Get All Profiles API コール

Get All Profiles API コールを使用して、Profiler プロファイルのリストを取得します。

表 6-23 Get All Profiles の特性

特性	説明
説明	Profiler プロファイルの集合を取得します
概要	GET /ers/config/profilerprofile
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	start、size、sortbyacn、sortbydcn、filter (フィルタでサポートされるフィールド: name)
要求メッセージ本文	—
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	SearchResult
応答ステータス	200、400、401、403、404、415、500

すべてのプロファイルの取得要求のサンプル

```
Method: GET
URI: https://10.56.13.196:9060/ers/config/profilerprofile
HTTP Accept header:
application/vnd.com.cisco.ise.identity.profilerprofile.1.0+xml
Request Content:
N/A
```

すべてのプロファイルの取得応答のサンプル

HTTP Status: 200 (OK)

Content:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult total="2" xmlns:ns2="ers.ise.cisco.com">
  <nextPage type="application/xml" href="link-to-next-page" rel="next"/>
  <previousPage type="application/xml" href="link-to-previous-page" rel="previous"/>
  <resources>
    <resource name="name1" id="id1" description="description1"/>
    <resource name="name2" id="id2" description="description2"/>
  </resources>
</ns2:searchResult>
```



(注)

このリリースでサポートされていないため、Profiler プロファイルの完全なオブジェクトに対するハイパーリンクはありません。名前の説明と ID だけが表示されます。

Get Profile API コール

Get Profile API コールを使用して、特定の Profiler プロファイルを取得します。

表 6-24 Get Profile の特性

特性	説明
説明	指定された Profiler プロファイルを取得します。
概要	GET ers/config/profilerprofile/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	start、size、sortbyacn、sortbydcn、filter (フィルタでサポートされるフィールド: name)
要求メッセージ本文	—
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	SearchResult
応答ステータス	200、400、401、403、404、415、500

プロファイルの取得要求のサンプル

Method: GET

URI: https://10.56.13.196:9060/ers/config/profilerprofile/{id}

HTTP Accept header:

application/vnd.com.cisco.ise.identity.profilerprofile.1.0+xml

Request Content:

N/A

プロファイルの取得応答のサンプル

HTTP Status: 200 (OK)

Content:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:profilerprofile name="name" id="id" description="description"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com"/>
```

Create Profile API コール

Create Profile API コールを使用して、Profiler プロファイルを作成します。

表 6-25 Create Profile の特性

特性	説明
説明	Profiler プロファイルを作成します
概要	POST ers/config/profilerprofile
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	start、size、sortbyacn、sortbydcn、filter (フィルタでサポートされるフィールド: name)
要求メッセージ本文	—
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	—
応答ステータス	200、400、401、403、404、415、500

プロファイルの作成要求のサンプル

Method: POST

URI: https://10.56.13.196:9060/ers/config/profilerprofile

HTTP Content-Type header:

application/vnd.com.cisco.ise.identity.profilerprofile.1.0+xml; charset=utf-8

Request Content:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:profilerprofile name="name" id="id" description="description"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com"/>
```


プロフィールの作成応答のサンプル

```
HTTP Status: 201 (Created)
Content:
N/A
```

ID グループの外部 RESTful サービス API コール

外部 RESTful サービスの ID グループ API コールを使用すると、ID グループを検索することができます。



(注)

後述の項に示されている例は、DB データへの参照があるため、そのまま使用するためのものではありません。例を基本テンプレートとして処理し、サーバに送信する前に編集する必要があります。

表 6-26 ID グループに対して使用できる外部 RESTful サービス API コール

オペレーション	メソッド	URL	内容	クエリー文字列
ID グループの取得 (ID で取得)	GET	/ers/config/identitygroup/{id}	—	—
ID グループのリスト	GET	/ers/config/identitygroup	—	page、size、sortacs または sortdsn、filter
IdentityGroup リソースのバージョン情報の取得	GET	/ers/config/identitygroup/version	—	—

ID グループのスキーマ

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:identitygroup name="name" id="id" description="description"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
  <parent>parent</parent>
</ns3:identitygroup>
```

Get All Identity Groups API コール

Get All Identity Groups API コールを使用して、Cisco ISE の ID グループを ID で取得します。

表 6-27 Get All Identity Groups の特性

特性	説明
説明	ID で ID グループ リソースを取得します。
概要	GET /ers/config/identitygroup/{id}
要求ヘッダー	Accept、Authorization、Host

表 6-27 Get All Identity Groups の特性 (続き)

特性	説明
クエリー文字列	start、size、sortbyacn、sortbydcn、filter (フィルタでサポートされるフィールド: name、description)
要求メッセージ本文	—
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	—
応答ステータス	200、400、401、403、404、415、500

すべての ID グループの取得要求のサンプル

```
Method: GET
URI: https://10.56.13.196:9060/ers/config/identitygroup/{id}
HTTP Accept header:
application/vnd.com.cisco.ise.identity.identitygroup.1.0+xml
Request Content:
N/A
```

すべての ID グループの取得応答のサンプル

```
HTTP Status: 200 (OK)
Content:
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:identitygroup name="name" id="id" description="description"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
  <parent>parent</parent>
</ns3:identitygroup>
```

Get Identity Group API コール

Get Identity Group API コールを使用して、Cisco ISE の特定の ID グループを取得します。

表 6-28 Get Identity Group の特性

特性	説明
説明	指定された ID グループのリソースを取得します
概要	GET /ers/config/identitygroup
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	start、size、sortbyacn、sortbydcn、filter (フィルタでサポートされるフィールド: name、description)
要求メッセージ本文	—

表 6-28 Get Identity Group の特性 (続き)

特性	説明
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	SearchResult
応答ステータス	200、400、401、403、404、415、500

ID グループの取得要求のサンプル

```
Method: GET
URI: https://10.56.13.196:9060/ers/config/identitygroup/{id}
HTTP Accept header:
application/vnd.com.cisco.ise.identity.identitygroup.1.0+xml
Request Content:
N/A
```

ID グループの取得応答のサンプル

```
HTTP Status: 200 (OK)
Content:
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:identitygroup name="name" id="id" description="description"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
  <parent>parent</parent>
</ns3:identitygroup>
```

セキュリティ グループ タグの外部 RESTful サービス API コール

セキュリティ グループ タグ (SGT) API コールを使用すると、SGT を検索することができます。これらの API コールで使用される内部ユーザ ID は Cisco ISE データベースに保存された UUID タイプです。



(注)

後述の項に示されている例は、DB データへの参照があるため、そのまま使用するためのものではありません。例を基本テンプレートとして処理し、サーバに送信する前に編集する必要があります。

表 6-29 SGT に対して使用できる外部 RESTful サービス API コール

オペレーション	メソッド	URL	内容	クエリー文字列
すべての SGT の取得	GET	/ers/config/sgt	—	page、size、sortacs または sortdsn、filter
SGT の取得 (ID で取得)	GET	/ers/config/sgt/{sgt-id}	—	—

Get All SGTs API コール

Get All SGTs API コールを使用して、SGT リソースの集合を取得します。

表 6-30 Get All SGTs の特性

特性	説明
説明	SGT リソースの集合を取得します
概要	GET /ers/config/sgt
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	start、size、sortbyacn、sortbydcn、filter (フィルタでサポートされるフィールド : name)
要求メッセージ本文	—
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	SearchResult
応答ステータス	200、400、401、403、404、415、500

すべての SGT の取得要求のサンプル

```
Method: GET
URI: https://10.56.13.196:9060/ers/config/sgt
HTTP Accept header:
application/vnd.com.cisco.ise.sga.sgt.1.0+xml
Request Content:
N/A
```

すべての SGT の取得応答のサンプル

```
HTTP Status: 200 (OK)
Content:
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult total="2" xmlns:ns2="ers.ise.cisco.com">
  <nextPage type="application/xml" href="link-to-next-page" rel="next"/>
  <previousPage type="application/xml" href="link-to-previous-page" rel="previous"/>
  <resources>
    <resource name="name1" id="id1" description="description1"/>
    <resource name="name2" id="id2" description="description2"/>
  </resources>
</ns2:searchResult>
```

Get SGT API コール

Get SGT API コールを使用して、特定の SGI リソースを取得します。

表 6-31 Get SGT の特性

特性	説明
説明	指定された SGT を取得します
概要	GET /ers/config/sgt/{sgt-id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	—
要求メッセージ本文	—
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	InternalUser タイプのリソース
応答ステータス	200、400、401、403、404、415、500

SGT の取得要求のサンプル

```
Method: GET
URI: https://10.56.13.196:9060/ers/config/sgt/{id}
HTTP Accept header:
application/vnd.com.cisco.ise.sga.sgt.1.0+xml
Request Content:
N/A
```

SGT の取得応答のサンプル

```
HTTP Status: 200 (OK)
Content:
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:sgt name="name" id="id" description="description" xmlns:ns2="ers.ise.cisco.com"
xmlns:ns3="sga.ers.ise.cisco.com">
  <generationId>generationId</generationId>
  <isTagFromRange>isTagFromRange</isTagFromRange>
  <value>1</value>
</ns3:sgt>
}
```

REST API クライアントの使用方法

外部 RESTful サービス API を使用してアプリケーションを構築およびテストするには、Google Chrome の POSTMAN プラグインなど、業界標準の REST API クライアントを使用できます。

REST アーキテクチャと原則に従って設計されているので、POSTMAN Chrome プラグインは、使いやすくて REST クライアントです。POSTMAN Chrome プラグインは、要求の収集を保存できるようにし、ブラウザ環境で作業をしているときに便利で、軽量の特定のタスクを含む多くの機能を提供します。外部 RESTful サービス API をテストするために POSTMAN プラグインを使用することもできます。

POSTMAN では、Google Chrome Web ブラウザを使用して標準の HTTP および HTTPS 要求および応答を送信および取得できるようにします。Cisco ISE リソースに対する操作を実行するために、次の標準 HTTP メソッドを使用できます。

- GET
- POST
- PUT
- DELETE

外部 RESTful サービス API はさまざまな API コールでこれらの HTTP 要求を使用できるようにし、次に、HTTP 要求は Cisco ISE サーバに対する操作を実行できるようにします。これらの HTTP 要求が使用される操作の包括的リストについては、「外部 RESTful サービス コール」(P.6-1) を参照してください。



(注)

POSTMAN プラグインをダウンロードするには、<https://chrome.google.com/webstore/detail/postman-rest-client/fdmmgilgnpjigdojppjoooidkmcocm?hl=en> にアクセスしてください。POSTMAN プラグインの使用法の詳細については、<https://github.com/a85/POSTMan-Chrome-Extension/wiki> を参照してください。

GET を使用した外部 RESTful サービス API コールの実行

GET メソッドは、指定されたリソースの表現を要求します。GET を使用した要求は、データを取得するだけで、他の効果はありません。

GET 要求をリソース URI に送信して、検索操作を実行することができます。デフォルトで、結果はデフォルト サイズが 20 の最初のページ (ページ インデックス = 0) です。フィルタ、ソート、および ページング パラメータを URI に追加して、検索操作の結果を制御できます。検索は、次の結果タイプを返します。

```
SearchResult: MediaType [MediaType:
[application/vnd.com.cisco.ise.ers.searchresult.1.0+xml]
```

結果として生成されたリソースは、名前、ID、説明、およびリソースの完全な表現へのリンクを含む基本リソースの集合です。



(注)

外部 RESTful サービス API コールは、この項で説明されていない外部 RESTful サービス API のその他のコンポーネントに加え、GET HTTP メソッドを使用します。特性、要求、および応答など、さまざまな外部 RESTful サービス API コンポーネントの詳細については、「外部 RESTful サービス コール」を参照してください。

Get HTTPS メソッドを使用する外部 RESTful サービス API コールの要求本文には、次の 3 つの構築ブロックが含まれます。

- URI
- Accept ヘッダー
- Authorization ヘッダー

URI

Get メソッドはユニフォーム リソース識別子 (URI) を Cisco ISE サーバに送信し、HTTP 返信は未処理の結果のデータです。一般的な URI は、次の形式に従う必要があります。

- `https://<Cisco ISE Server address:<port>/<namespace>/config/<Cisco ISE Resource Name>`

ここで、`<Cisco ISE Server Address>` は Cisco ISE サーバのサーバアドレスを示し、`<port>` はポート 9060 を示し、`<namespace>` は Cisco ISE リソースが属する名前空間を示し、`<Cisco ISE Resource Name>` は Cisco ISE リソースの名前を示します。

次に、interaluser Cisco ISE リソースのデータを要求する URI の例を示します。

- `https://10.56.13.196:9060/ers/config/internaluser.`



(注) URI は、要求本文ではなく、URL にすぎません。この URL は、GET メソッドを使用してサーバに送信されます。

Accept ヘッダー

Accept ヘッダーは次の形式に従う必要があります。

- `application/vnd.com.cisco.ise.<resource-namespace>.<resource-type>.<major version>.<minor version>+xml`

ここで、`<resource-namespace>` は Cisco ISE リソースが属する名前空間を示し、`<resource-type>` は Cisco ISE リソースのタイプを示し、`<major-version>` は Cisco ISE 導入のメジャーバージョン番号を示し、`<minor-version>` は導入のマイナーバージョン番号を示します。

次に、一般的な Accept ヘッダーの例を示します。

- `application/vnd.com.cisco.ise.identity.internaluser.1.0+xml`

Authorization ヘッダー

Authorization ヘッダーには、GET 要求に組み込まれている暗号化認証キーが含まれます。認可のクレデンシャルを指定した後、暗号化キーを生成する必要があります。その後、暗号化キーは要求の本体に埋め込まれます。



(注) 暗号化キーの生成の詳細については、「[POSTMAN を使用した GET 要求の実行](#)」(P.6-31) を参照してください。

POSTMAN を使用した GET 要求の実行

ステップ 1 Google Chrome ブラウザで POSTMAN プラグインを開きます。

ステップ 2 左側のペインのオプションを使用して、新しい収集を作成します。



(注) POSTMAN プラグインの使用法の詳細については、<https://github.com/a85/POSTMan-Chrome-Extension/wiki> を参照してください。

ステップ 3 ドロップダウンメニューから、[GET] を選択します。

ステップ 4 URL アドレス バーに、URI を入力します。

URI で、通信しようとする Cisco ISE サーバ、およびアクセスしようとする Cisco ISE リソースを指定します。URI の形式の詳細については、「[URI](#)」(P.6-30) を参照してください。

ステップ 5 [基本認証 (Basic Auth)] タブをクリックします。

ユーザアクセス クレデンシャルを指定できるオプションが表示されます。

ステップ 6 アクセス クレデンシヤルを [ユーザ名 (Username)] および [パスワード (Password)] フィールドに指定し、[ヘッダーの更新 (Refresh Headers)] をクリックします。

POSTMAN は暗号化キーを含む Authorization ヘッダーを表示します。

ステップ 7 次の値を指定して、Accept ヘッダーを追加します。
application/vnd.com.cisco.ise.ers.<namespace>.<ise resource>.1.0+xml



(注) Accept ヘッダーの詳細については、「[Accept ヘッダー](#)」(P.6-31) を参照してください。

ステップ 8 [送信 (Send)] をクリックします。

POSTMAN プラグインは、要求が成功していることを示す 200 OK ステータスの応答を表示します。要求は、URI で指定したリソースの詳細も返します。

POST を使用した外部 RESTful サービス API コールの実行

Post メソッドは、要求に含まれているエンティティを、URI で識別される Web リソースの新しい下位として、サーバが承認することを要求します。

POST 要求をリソース URI に送信して、リソースを作成することができます。要求するコンテンツは、リソースの XML 表現を保持し、スキーマに従って正しい形式にする必要があります。要求ヘッダーは、暗号化された認可、およびリソースのコンテンツとそのバージョンを定義するコンテンツタイプを保持します。



(注) 外部サービス RESTful API コールは、このセクションで説明されていない外部 RESTful API サービスの他のコンポーネントに加え、HTTP POST 方式を使用します。特性、要求、および応答など、さまざまなコンポーネントの詳細については、「[外部 RESTful サービス コール](#)」を参照してください。

POST HTTP メソッドを使用する外部 RESTful サービス API コールの要求本文には、次の 3 つの構築ブロックが含まれます。

- [URI](#)
- [Content-Type ヘッダー](#)
- [Authorization ヘッダー](#)

URI

POST メソッドはユニフォーム リソース識別子 (URI) を Cisco ISE サーバに送信します。一般的な URI は、次の形式に従う必要があります。

- *https://<Cisco ISE Server address:<port>/<namespace>/config/<Cisco ISE Resource Name>*

ここで、<Cisco ISE Server Address> は Cisco ISE サーバのサーバアドレスを示し、<port> はポート 9060 を示し、<namespace> は Cisco ISE リソースが属する名前空間を示し、<Cisco ISE Resource Name> は Cisco ISE リソースの名前を示します。

次に、internaluser Cisco ISE リソースのデータを要求する URI の例を示します。

- *https://10.56.13.196:9060/ers/config/internaluser.*



(注) URI は、要求本文ではなく、URL にすぎません。この URL は、POST メソッドを使用してサーバに送信されます。

Content-Type ヘッダー

Content-Type ヘッダーは次の形式に従う必要があります。

- `application/vnd.com.cisco.ise.<resource-namespace>.<resource-type>.<major version>.<minor version>+xml`

ここで、`<resource-namespace>` は Cisco ISE リソースが属する名前空間を示し、`<resource-type>` は Cisco ISE リソースのタイプを示し、`<major-version>` は Cisco ISE 導入のメジャーバージョン番号を示し、`<minor-version>` は導入のマイナーバージョン番号を示します。

次に、一般的な Content-Type ヘッダーの例を示します。

- `application/vnd.com.cisco.ise.identity.internaluser.1.0+xml`

Authorization ヘッダー

Authorization ヘッダーには、POST 要求に組み込まれている暗号化認証キーが含まれます。認可のクレデンシャルを指定した後、暗号化キーを生成する必要があります。その後、暗号化キーは要求の本体に埋め込まれます。



(注) 暗号化キーの生成の詳細については、「[POSTMAN を使用した POST 要求の実行](#)」(P.6-33) を参照してください。

POSTMAN を使用した POST 要求の実行

ステップ 1 Google Chrome ブラウザで POSTMAN プラグインを開きます。

ステップ 2 左側のペインのオプションを使用して、新しい収集を作成します。



(注) POSTMAN プラグインの使用法の詳細については、<https://github.com/a85/POSTMan-Chrome-Extension/wiki> を参照してください。

ステップ 3 ドロップダウンメニューから、[POST] を選択します。

ステップ 4 URL アドレス バーに、URI を入力します。

URI で、通信しようとする Cisco ISE サーバ、およびアクセスしようとする Cisco ISE リソースを指定します。URI の形式の詳細については、「[URI](#)」(P.6-32) を参照してください。

ステップ 5 [基本認証 (Basic Auth)] タブをクリックします。

ユーザアクセス クレデンシャルを指定できるオプションが表示されます。

ステップ 6 アクセス クレデンシャルを [ユーザ名 (Username)] および [パスワード (Password)] フィールドに指定し、[ヘッダーの更新 (Refresh Headers)] をクリックします。

POSTMAN は暗号化キーを含む Authorization ヘッダーを表示します。

ステップ 7 次の値を指定して、Content-Type ヘッダーを追加します。

`application/vnd.com.cisco.ise.ers.<namespace>.<ise resource>.1.0+xml`



(注) Content-Type ヘッダーの詳細については、「[Content-Type ヘッダー](#)」(P.6-33)を参照してください。

ステップ 8 [未処理 (raw)] の横に表示されるドロップダウンメニューから、[XML] を選択します。

ステップ 9 [未処理 (raw)] をクリックします。

ステップ 10 POSTMAN プラグインは、POST 要求の本文を指定できる編集ペインを開きます。

ステップ 11 編集ペインで POST 要求のメッセージ本文を入力します。



(注) メッセージ本文には、Cisco ISE サーバに作成しようとする Cisco ISE リソースに対応する詳細を含める必要があります。たとえば、内部ユーザを作成しているときには、内部ユーザの名前や説明、パスワードなどの詳細を指定する必要があります。POST 要求を使用する外部 RESTful サービス API コールのメッセージ本文、および指定する必要がある Cisco ISE リソースの詳細については、「[外部 RESTful サービス コール](#)」を参照してください。

ステップ 12 [送信 (Send)] をクリックします。

POSTMAN プラグインは、要求が成功していることを示す 201 CREATED ステータスの応答を表示します。追加したリソースが GUI に表示されているかどうかを確認するために、Cisco ISE にログインできます。

PUT を使用した外部 RESTful サービス API コールの実行

PUT メソッドは、含められたエンティティを指定された URI の下に保存するように要求します。URI がすでに存在しているリソースを指す場合は、変更されます。URI が既存のリソースを指さない場合は、サーバはその URI でリソースを作成できます。

PUT 要求をリソース URI に送信して、リソースを更新することができます。要求するコンテンツは、リソースの XML 表現を保持し、スキーマに従って正しい形式にする必要があります。要求ヘッダーは、暗号化された認可、およびリソースのコンテンツとそのバージョンを定義するコンテンツタイプを保持します。



(注) 外部 RESTful サービス API コールは、この項で説明されていない外部 RESTful サービス API のその他のコンポーネントに加え、PUT HTTP メソッドを使用します。特性、要求、および応答など、さまざまなコンポーネントの詳細については、「[外部 RESTful サービス コール](#)」を参照してください。

PUT HTTP メソッドを使用する外部 RESTful サービス API コールの要求本文には、次の 3 つの構築ブロックが含まれます。

- [URI](#)
- [Content-Type ヘッダー](#)
- [Authorization ヘッダー](#)

URI

PUT メソッドはユニフォーム リソース識別子 (URI) を Cisco ISE サーバに送信します。一般的な URI は、次の形式に従う必要があります。

- `https://<Cisco ISE Server address:<port>/<namespace>/config/<Cisco ISE Resource Name>`

ここで、`<Cisco ISE Server Address>` は Cisco ISE サーバのサーバアドレスを示し、`<port>` はポート 9060 を示し、`<namespace>` は Cisco ISE リソースが属する名前空間を示し、`<Cisco ISE Resource Name>` は Cisco ISE リソースの名前を示します。

次に、interaluser Cisco ISE リソースのデータを要求する URI の例を示します。

- `https://10.56.13.196:9060/ers/config/internaluser.`



(注) URI は、要求本文ではなく、URL にすぎません。この URL は、PUT メソッドを使用してサーバに送信されます。

Content-Type ヘッダー

Content-Type ヘッダーは次の形式に従う必要があります。

- `application/vnd.com.cisco.ise.<resource-namespace>.<resource-type>.<major version>.<minor version>+xml`

ここで、`<resource-namespace>` は Cisco ISE リソースが属する名前空間を示し、`<resource-type>` は Cisco ISE リソースのタイプを示し、`<major-version>` は Cisco ISE 導入のメジャーバージョン番号を示し、`<minor-version>` は導入のマイナーバージョン番号を示します。

次に、一般的な Content-Type ヘッダーの例を示します。

- `application/vnd.com.cisco.ise.identity.internaluser:1.0+xml`

Authorization ヘッダー

Authorization ヘッダーには、PUT 要求に組み込まれている暗号化認証キーが含まれます。認可のクレデンシャルを指定した後、暗号化キーを生成する必要があります。その後、暗号化キーは要求の本体に埋め込まれます。



(注) 暗号化キーの生成の詳細については、「[POSTMAN を使用した PUT 要求の実行](#)」(P.6-35) を参照してください。

POSTMAN を使用した PUT 要求の実行

- ステップ 1 Google Chrome ブラウザで POSTMAN プラグインを開きます。
- ステップ 2 左側のペインのオプションを使用して、新しい収集を作成します。



(注) POSTMAN プラグインの使用法の詳細については、<https://github.com/a85/POSTMan-Chrome-Extension/wiki> を参照してください。

- ステップ 3 ドロップダウンメニューから、[PUT] を選択します。

- ステップ 4** URL アドレス バーに、URI を入力します。
- URI で、通信しようとする Cisco ISE サーバ、およびアクセスしようとする Cisco ISE リソースを指定します。URI の形式の詳細については、「[URI](#)」(P.6-35) を参照してください。
- ステップ 5** [基本認証 (Basic Auth)] タブをクリックします。
- ユーザ アクセス クレデンシャルを指定できるオプションが表示されます。
- ステップ 6** アクセス クレデンシャルを [ユーザ名 (Username)] および [パスワード (Password)] フィールドに指定し、[ヘッダーの更新 (Refresh Headers)] をクリックします。
- POSTMAN は暗号化キーを含む Authorization ヘッダーを表示します。
- ステップ 7** 次の値を指定して、Content-Type ヘッダーを追加します。
- ```
application/vnd.com.cisco.ise.ers.<namespace>.<ise resource>.1.0+xml
```



(注) Content-Type ヘッダーの詳細については、「[Content-Type ヘッダー](#)」(P.6-35) を参照してください。

- ステップ 8** [未処理 (raw)] の横に表示されるドロップダウンメニューから、[XML] を選択します。
- ステップ 9** [未処理 (raw)] をクリックします。
- ステップ 10** POSTMAN プラグインは、POST 要求の本文を指定できる編集ペインを開きます。
- ステップ 11** 編集ペインで POST 要求のメッセージ本文を入力します。



(注) メッセージ本文には、Cisco ISE サーバに更新しようとする Cisco ISE リソースに対応する詳細を含める必要があります。たとえば、内部ユーザを更新しているときには、内部ユーザの名前や説明、パスワードなどの詳細を指定する必要があります。PUT 要求を使用する外部 RESTful サービス API コールのメッセージ本文、および指定する必要がある Cisco ISE リソースの詳細の詳細については、「[外部 RESTful サービス コール](#)」を参照してください。

- ステップ 12** [送信 (Send)] をクリックします。
- POSTMAN プラグインは、要求が成功していることを示す 201 CREATED ステータスの応答を表示します。追加したリソースが GUI に表示されているかどうかを確認するために、Cisco ISE にログインできます。

## DELETE を使用した外部 RESTful サービス API コールの実行

DELETE メソッドは、指定されたリソースを削除します。

DELETE 要求をリソース URI に送信して、リソースを削除することができます。要求ヘッダーは、暗号化された認可、およびリソースのコンテンツとそのバージョンを定義するコンテンツタイプを保持します。



(注) 外部 RESTful サービス API コールは、この項で説明されていない外部 RESTful サービス API のその他のコンポーネントに加え、DELETE HTTP メソッドを使用します。特性、要求、および応答など、さまざまなコンポーネントの詳細については、「[外部 RESTful サービス コール](#)」を参照してください。

DELETE HTTP メソッドを使用する外部 RESTful サービス API コールの要求本文には、次の 3 つの構築ブロックが含まれます。

- [URI](#)
- [Accept ヘッダー](#)
- [Authorization ヘッダー](#)

## URI

DELETE メソッドはユニフォーム リソース識別子 (URI) を Cisco ISE サーバに送信します。一般的な URI は、次の形式に従う必要があります。

- `https://<Cisco ISE Server address:<port>/<namespace>/config/<Cisco ISE Resource Name>`

ここで、`<Cisco ISE Server Address>` は Cisco ISE サーバのサーバアドレスを示し、`<port>` はポート 9060 を示し、`<namespace>` は Cisco ISE リソースが属する名前空間を示し、`<Cisco ISE Resource Name>` は Cisco ISE リソースの名前を示します。

次に、internaluser Cisco ISE リソースのデータを要求する URI の例を示します。

- `https://10.56.13.196:9060/ers/config/internaluser.`



(注) URI は、要求本文ではなく、URL にすぎません。この URL は、GET メソッドを使用してサーバに送信されます。

## Accept ヘッダー

Accept ヘッダーは次の形式に従う必要があります。

- `application/vnd.com.cisco.ise.<resource-namespace>.<resource-type>.<major version>.<minor version>+xml`

ここで、`<resource-namespace>` は Cisco ISE リソースが属する名前空間を示し、`<resource-type>` は Cisco ISE リソースのタイプを示し、`<major-version>` は Cisco ISE 導入のメジャーバージョン番号を示し、`<minor-version>` は導入のマイナーバージョン番号を示します。

次に、一般的な Accept ヘッダーの例を示します。

- `application/vnd.com.cisco.ise.identity.internaluser.1.0+xml`

## Authorization ヘッダー

Authorization ヘッダーには、DELETE 要求に組み込まれている暗号化認証キーが含まれます。認可のクレデンシャルを指定した後、暗号化キーを生成する必要があります。その後、暗号化キーは要求の本体に埋め込まれます。



(注) 暗号化キーの生成の詳細については、「[POSTMAN を使用した DELETE 要求の実行](#)」(P.6-37) を参照してください。

## POSTMAN を使用した DELETE 要求の実行

**ステップ 1** Google Chrome ブラウザで POSTMAN プラグインを開きます。

**ステップ 2** 左側のペインのオプションを使用して、新しい収集を作成します。



(注) POSTMAN プラグインの使用法の詳細については、<https://github.com/a85/POSTMan-Chrome-Extension/wiki> を参照してください。

**ステップ 3** ドロップダウンメニューから、[DELETE] を選択します。

**ステップ 4** URL アドレスバーに、URI を入力します。

URI で、通信しようとする Cisco ISE サーバ、およびアクセスしようとする Cisco ISE リソースを指定します。URI の形式の詳細については、「URI」(P.6-37) を参照してください。

**ステップ 5** [基本認証 (Basic Auth)] タブをクリックします。

ユーザアクセス クレデンシャルを指定できるオプションが表示されます。

**ステップ 6** アクセス クレデンシャルを [ユーザ名 (Username)] および [パスワード (Password)] フィールドに指定し、[ヘッダーの更新 (Refresh Headers)] をクリックします。

POSTMAN は暗号化キーを含む Authorization ヘッダーを表示します。

**ステップ 7** 次の値を指定して、Accept ヘッダーを追加します。

`application/vnd.com.cisco.ise.ers.<namespace>.<ise resource>.1.0+xml`



(注) Accept ヘッダーの詳細については、「Accept ヘッダー」(P.6-37) を参照してください。

**ステップ 8** [送信 (Send)] をクリックします。

POSTMAN プラグインは、要求が成功していることを示す 200 OK ステータスの応答を表示します。追加したリソースが GUI に表示されているかどうかを確認するために、Cisco ISE にログインできません。

## 外部 RESTful サービス Java デモ アプリケーション

外部 RESTful サービス Java デモ アプリケーションは、単純な外部 RESTful サービス クライアントコードで、外部 RESTful サービス API コールを使用して内部ユーザおよびエンドポイントを設定します。この Java クライアントコードの主な Cisco ISE 依存関係は、`ers-schema.jar` ファイルです。デモ Java アプリケーションは、`Apache DefaultHttpClient` に基づく汎用 HTTP クライアントおよび 2 つのメインクラスから構成されます。これらのクラスは、順番に REST コールを実行します。

ソースコードおよび必要なライブラリは、デモ アプリケーションの `zip` ファイルで参照として使用可能です。外部 RESTful サービス Java デモ アプリケーションおよび `ers-schema.jar` ファイルは、外部 RESTful サービス オンライン SDK からダウンロードできます。

### 関連項目

- 「外部 RESTful サービス Java デモ アプリケーションのダウンロード」(P.6-39)

## 外部 RESTful サービス Java デモ アプリケーションのダウンロード

- 
- ステップ 1** ブラウザのアドレス バーに外部 RESTful サービス SDK の URL を入力します。たとえば、`https://<ise hostname or ip address>:9060/ers/sdk` と入力します。ここで、`<ise hostname or ip address>` は Cisco ISE ホストの IP アドレスを示します。
- ステップ 2** ユーザ名と、Cisco ISE の初期セットアップで指定して設定した大文字と小文字が区別されるパスワードを入力します。
- ステップ 3** [ログイン (Login)] をクリックするか、Enter を押します。
- ステップ 4** 外部 RESTful サービス オンライン SDK のページのナビゲーション ペインから、[開発者リソース (Developer Resources)] > [ダウンロード (Downloads)] を選択します。
- ステップ 5** [スキーマ Jar (ers-schema-1.2.0-896.jar) (Schema Jar (ers-schema-1.2.0-896.jar))] をクリックします。  
*ers-schema-1.0-892.jar* ファイルがローカル マシンにダウンロードされます。
- ステップ 6** [ERS Java デモ アプリケーション (Java ERS Demo Application)] をクリックします。  
*ers-demo-app.zip* ファイルがローカル マシンにダウンロードされます。
- 

### 終了後



- (注) 外部 RESTful サービス Java デモ アプリケーションを使用するために必要なすべての情報は、デモ アプリケーション zip ファイル (*ers-demo-app.zip*) にある *readme.txt* ファイルを参照してください。
-

