

# ISE IDグループ、Rest APIによるユーザの作成と変更

## 内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[設定](#)

[1.アイデンティティグループの作成](#)

[2.アイデンティティグループの詳細取得](#)

[3.ユーザの作成](#)

[4.ユーザー詳細の取得](#)

[5.ユーザーの詳細変更](#)

[確認](#)

## 概要

このドキュメントでは、アイデンティティ管理の自動化に使用できるRest APIを使用して、アイデンティティグループとユーザを作成および変更する方法について説明します。この章で説明する手順は、サンプルのスタンドアロンISE展開とJSON形式のRest API Firefox Client(SIST)に基づいています。

## 前提条件

### 要件

次の項目に関する知識があることが推奨されます。

- Cisco Identity Services Engine ( ISE )
- REST API
- JSON

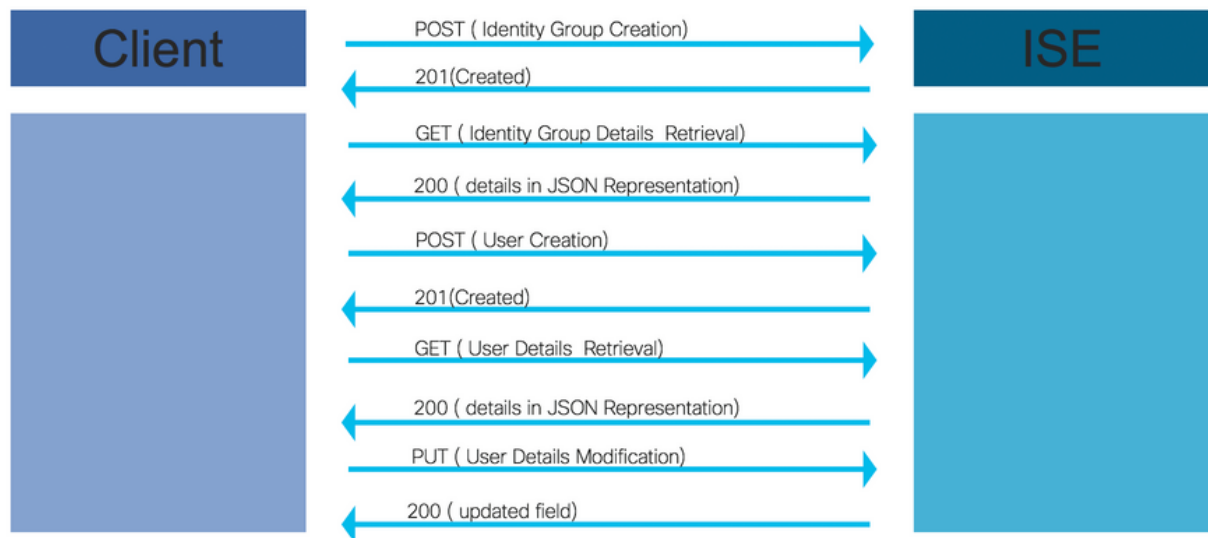
### 使用するコンポーネント

このドキュメントは、特定のソフトウェアおよびハードウェアのバージョンに限定されるものではなく、REST APIを使用した設定例のガイドにすぎません。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

# 設定

## API Request and Response flow



以下に、APIを使用した操作のサンプルステップを示します。このステップは、独自のコールを作成するための参照として使用できます。

### 1. アイデンティティグループの作成

POSTメソッドを使用してIDグループを作成してください。

APIコールのURL:

**<https://<ISE IP>:9060/ers/config/identitygroup>**

APIコールのヘッダー:

HTTP 'Content-Type'ヘッダー: application/json  
HTTP 'Accept'ヘッダー: application/json

IDグループ作成用のJSONコード

```
{ "IdentityGroup": { "name": "  
例:
```

Request ↗ ⚙ +

POST  Send request

**Headers** ▾

|              |                  |   |
|--------------|------------------|---|
| Content-Type | application/json | 🗑 |
| Accept       | application/json | 🗑 |

[+Add header](#)

**Basic auth** ▾

|          |       |   |
|----------|-------|---|
| ersadmin | ..... | <input type="checkbox"/> Show password? |
|----------|-------|---|

**Request body** ▾

Type

```
{
  "IdentityGroup": {
    "name": "testusergroup",
    "description": " User group for testing",
    "parent": "NAC Group:NAC:IdentityGroups:User Identity Groups"
  }
}
```

Response (0.711s) - https://10.71.244.140:9060/ers/config/identitygroup

**201** Created

[Headers >](#)

## 2. アイデンティティグループの詳細取得

GETメソッドを使用して、アイデンティティグループの詳細を取得します。

APIコールのURL:

https://<ISE IP>:9060/ers/config/identitygroup?filter=name.CONTAINS.<IDグループ名>

APIコールのヘッダー :

HTTP 'Content-Type'ヘッダー : application/json

HTTP 'Accept'ヘッダー : application/json

例 :

GET

**Headers** ▾

|              |                  |   |
|--------------|------------------|---|
| Content-Type | application/json | 🗑 |
| Accept       | application/json | 🗑 |

[+Add header](#)

**Basic auth** ▾

|          |       |   |
|----------|-------|---|
| ersadmin | ..... | <input type="checkbox"/> Show password? |
|----------|-------|---|

---

Response (0.786s) - https://10.71.244.140:9060/ers/config/identitygroup?filter=name.CONTAINS.testusergroup

**200** OK

**Headers** >

```
{
  "SearchResult": {
    "total": 1,
    "resources": [
      {
        "id": "b6ae7220-4289-11ea-a840-cee7c3fe1b0d",
        "name": "testusergroup",
        "description": " User group for testing",
        "link": {
          "rel": "self",
          "href": "https://10.71.244.140:9060/ers/config/identitygroup/b6ae7220-4289-11ea-a840-cee7c3fe1b0d",
          "type": "application/xml"
        }
      }
    ]
  }
}
```

注：このアイデンティティグループにユーザを作成するには、ID (アイデンティティグループの詳細で受信) が必要です。

### 3. ユーザの作成

POSTメソッドを使用してユーザーを作成します。

APIコールのURL:

https://<ISE IP>:9060/ers/config/internaluser/

APIコールのヘッダー:

HTTP Content-Typeヘッダー : application/json  
 :  
 HTTP受け入れヘッダー : application/json

ユーザ作成用のJSONコード:

```
{ "InternalUser": { "name": "
```

例:

Request

POST  [Send request](#)

Headers

|              |                  |  |
|--------------|------------------|--|
| Content-Type | application/json |  |
| Accept       | application/json |  |

[+Add header](#)

Basic auth

|          |       |   |
|----------|-------|---|
| ersadmin | ..... | <input type="checkbox"/> Show password? |
|----------|-------|---|

Request body

Type

```
{
  "InternalUser": {
    "name": "test123",
    "email": "test@cisco.com",
    "enabled": true,
    "password": "Letmein11",
    "firstName": "test",
    "lastName": "test",
    "changePassword": false,
    "identityGroups": "b6ae7220-4289-11ea-a840-cee7c3fe1b0d",
  }
}
```

Response (80.364s) - https://10.71.244.140:9060/ers/config/internaluser/

**201** Created

[Headers >](#)

## 4. ユーザー詳細の取得

GETメソッドを使用してユーザーの詳細を取得します。

APIコールのURL:

**https://<ISE IP>:9060/ers/config/internaluser**

注：このURLは、ユーザをフィルタリングするために使用できます。ユーザは、firstName、lastName、identityGroup、name、description、email、enabledを使用してフィルタリングできます。

電子メールIDはユーザごとに一意であるため、電子メールIDを使用してユーザの詳細をフィルタリングすることをお勧めします。

**https://<ISE IP>:9060/ers/config/internaluser?filter=<フィルタリングに使用するフィールドの名前>.CONTAINS.<フィルタリング用フィールドの値>**

APIコールのヘッダー：

**HTTP Content-Type**ヘッダー application/json

:  
HTTP受け入れヘッダー : application/json

例 :

The screenshot shows a REST client interface. At the top, the method is set to 'GET' and the URL is '140:9060/ers/config/internaluser?filter=email.CONTAINS.test@cisco.com'. A 'Send request' button is visible. Below the URL, the 'Headers' section is expanded, showing 'Content-Type' and 'Accept' both set to 'application/json'. There is also a '+Add header' button. The 'Basic auth' section is expanded, showing the username 'ersadmin' and a masked password '.....'. A checkbox for 'Show password?' is present. Below the request configuration, the response is shown as 'Response (1.348s) - https://10.71.244.140:9060/ers/config/internaluser?filter=email.CONTAINS.test@cisco.com'.

200 OK

Headers >

```
{
  "SearchResult": {
    "total": 1,
    "resources": [
      {
        "id": "be9aee27-22f2-4a77-a8e9-3092ba3b636e",
        "name": "test123",
        "link": {
          "rel": "self",
          "href": "https://10.71.244.140:9060/ers/config/internaluser/be9aee27-22f2-4a77-a8e9-3092ba3b636e",
          "type": "application/xml"
        }
      }
    ]
  }
}
```

注:ユーザのパスワードやその他の情報を更新するには、ここで受信したIDおよびNameが必要です。hrefURLはユーザ情報の更新に使用されます。

## 5. ユーザーの詳細変更

PUTメソッドを使用して、ユーザーパスワードを変更します。

APIコールのURL:

https://<ISE IP>:9060/ers/config/internaluser/<ステップ4で説明したプロセスを使用してユーザIDを受信>

上記は、ステップ4で説明したプロセスを使用して受信したhref URLです。

APIコールのヘッダー :

HTTP Content-Typeヘッダー application/json

:  
HTTP受け入れヘッダー : application/json

ユーザー資格情報の変更用のJSONコード :

```
{ "InternalUser": { "id": "
```

例 :

The screenshot shows a REST client interface with the following sections:

- Request:** Method is PUT, URL is 140:9060/ers/config/internaluser/be9aee27-22f2-4a77-a8e9-3092ba3b636e. A "Send request" button is present.
- Headers:** Content-Type is application/json, Accept is application/json. There is a "+Add header" link.
- Basic auth:** Username is ersadmin, password is masked with dots. A "Show password?" checkbox is present.
- Request body:** Type is Custom. The body contains a JSON object: 

```
{  
  "InternalUser": {  
    "id": "be9aee27-22f2-4a77-a8e9-3092ba3b636e",  
    "name": "test123",  
    "password": "Letmein!3",  
    "enablePassword": "Letmein!3"  
  }  
}
```
- Response:** Status is 200 OK, URL is https://10.71.244.140:9060/ers/config/internaluser/be9aee27-22f2-4a77-a8e9-3092ba3b636e.

## 確認

IDグループを確認するには、ISE GUIで[Administration] > [Identity Management] > [Groups] > [Identity Groups] > [User Identity Group]に移動します。

ユーザを確認するには、ISE GUIで[Administration] > [Identity Management] > [Identities] > [Usersin]に移動します。