

ISE身份組，通過Rest API建立和修改使用者

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[設定](#)

[1.身份組建立](#)

[2.身份組詳細資訊檢索](#)

[3.使用者建立](#)

[4.使用者詳細資訊檢索](#)

[5.修改使用者詳細資訊](#)

[驗證](#)

簡介

本文檔介紹如何使用可用於身份管理自動化的Rest API建立和修改身份組和使用者。本章中介紹的過程基於示例獨立ISE部署和JSON格式的Rest API Firefox客戶端(REST)。

必要條件

需求

思科建議您瞭解以下主題：

- 思科身分識別服務引擎(ISE)
- REST API
- JSON

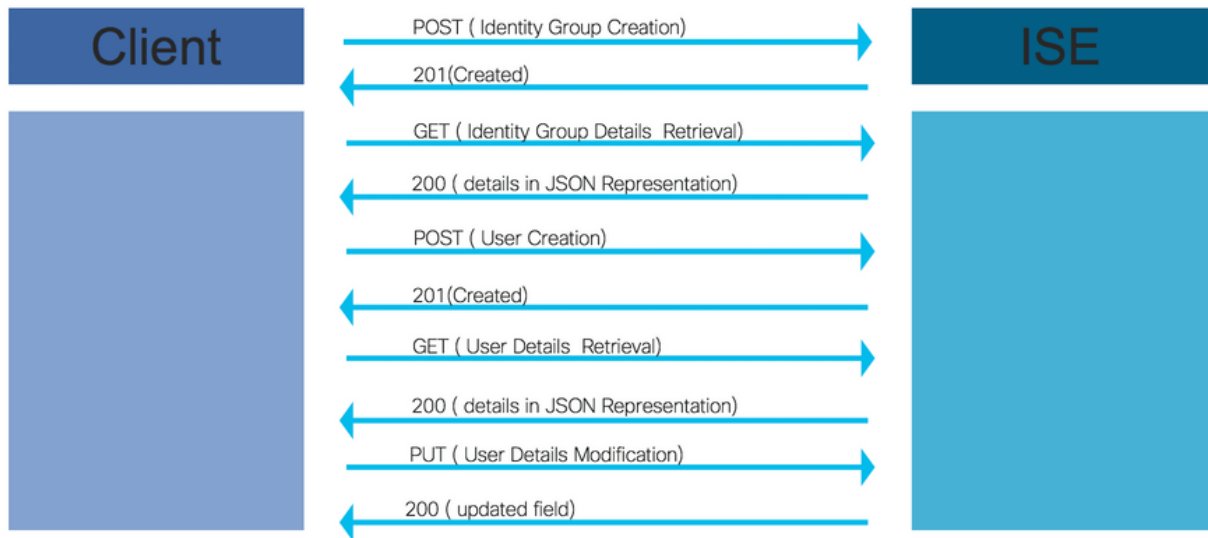
採用元件

本文檔不限於特定的軟體和硬體版本，只是通過REST API提供的示例配置指南。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

設定

API Request and Response flow



以下是通過API進行操作的示例步驟，可用作構建您自己的呼叫的參考。

1.身份組建立

使用POST方法建立身份組。

API呼叫的URL:

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

API呼叫的報頭 :

HTTP 'Content-Type'標頭 : application/json

HTTP 「Accept」 標頭 : application/json

用於建立身份組的JSON代碼

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

範例 :

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. 身份組詳細資訊檢索

使用GET 方法獲取身份組詳細資訊。

API呼叫的URL:

https://<ISE IP>:9060/ers/config/identitygroup?filter=name.CONTAINS。 <身份組名稱>

API呼叫的報頭 :

HTTP 'Content-Type'標頭 : application/json

HTTP 「Accept」 標頭 : application/json

範例 :

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"
        }
      }
    ]
  }
}

```

附註：在此身份組中建立使用者時需要ID（在身份組詳細資訊中接收）。

3.使用者建立

使用POST方法建立使用者。

API呼叫的URL:

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

API呼叫的報頭：

HTTP **Content-Type** Header: application/json

HTTP **Accept** Header: application/json

用於使用者建立的JSON代碼：

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

範例：

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. 使用者詳細資訊檢索

使用GET方法獲取使用者詳細資訊。

API呼叫的URL:

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

附註：此URL可用於過濾使用者。可以使用firstName、lastName、identityGroup、name、description、email、enabled過濾使用者。

建議使用電子郵件ID篩選使用者詳細資訊，因為電子郵件ID對每個使用者是唯一的。

<https://<ISE IP>:9060/ers/config/internaluser?filter=<用於過濾的欄位名稱>.CONTAINS.<用於過濾的欄位值>>

API呼叫的報頭：

HTTP **Content-Type** Header: application/json
HTTP **Accept** Header: application/json

範例：

Request

GET 140:9060/ers/config/internaluser?filter=email.CONTAINS.test@cisco.com Send request

Headers

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

+Add header

Basic auth

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

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"
        }
      }
    ]
  }
}
```

注意:此處接收的ID和Name received是更新使用者的密碼或其他資訊所必需的。href URL將用於更新使用者資訊。

5. 修改使用者詳細資訊

使用PUT Method的幫助修改使用者密碼。

API呼叫的URL:

https://<ISE IP>:9060/ers/config/internaluser/<使用步驟4中描述的進程接收的使用者ID>

以上是使用步驟4中介紹的過程接收的href URL。

API呼叫的報頭：

HTTP **Content-Type** Header: application/json
HTTP **Accept** Header: application/json

使用者憑據修改的JSON代碼：

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

範例：

The screenshot displays a REST client interface for a PUT request. The URL is `140:9060/ers/config/internaluser/be9aee27-22f2-4a77-a8e9-3092ba3b636e`. The request headers are `Content-Type: application/json` and `Accept: application/json`. The basic authentication is set to `ersadmin` with a masked password. The request body is a JSON object:

```
{ "InternalUser": { "id": "be9aee27-22f2-4a77-a8e9-3092ba3b636e", "name": "test123", "password": "Letmein!3", "enablePassword": "Letmein!3" } }
```

. The response is `200 OK` from `https://10.71.244.140:9060/ers/config/internaluser/be9aee27-22f2-4a77-a8e9-3092ba3b636e`.

驗證

要驗證身份組，請導覽至ISE GUI中的管理>身份管理>組>身份組>使用者身份組。

要驗證使用者，請導覽至Administration > Identity Management > Identities > Users in ISE GUI。