

ISE Identity-Group, creazione e modifica di utenti tramite l'API Rest

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Configurazione](#)

[1. Creazione del gruppo di identità](#)

[2. Recupero dei dettagli del gruppo di identità](#)

[3. Creazione utente](#)

[4. Recupero dei dettagli utente](#)

[5. Modifica dettagli utente](#)

[Verifica](#)

Introduzione

In questo documento viene descritto come creare e modificare gruppi di identità e utenti utilizzando l'API Rest che può essere utilizzata per l'automazione della gestione delle identità. La procedura descritta in questo capitolo si basa sull'implementazione ISE standalone di esempio e sul client Firefox (RESTED) dell'API REST in formato JSON.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

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

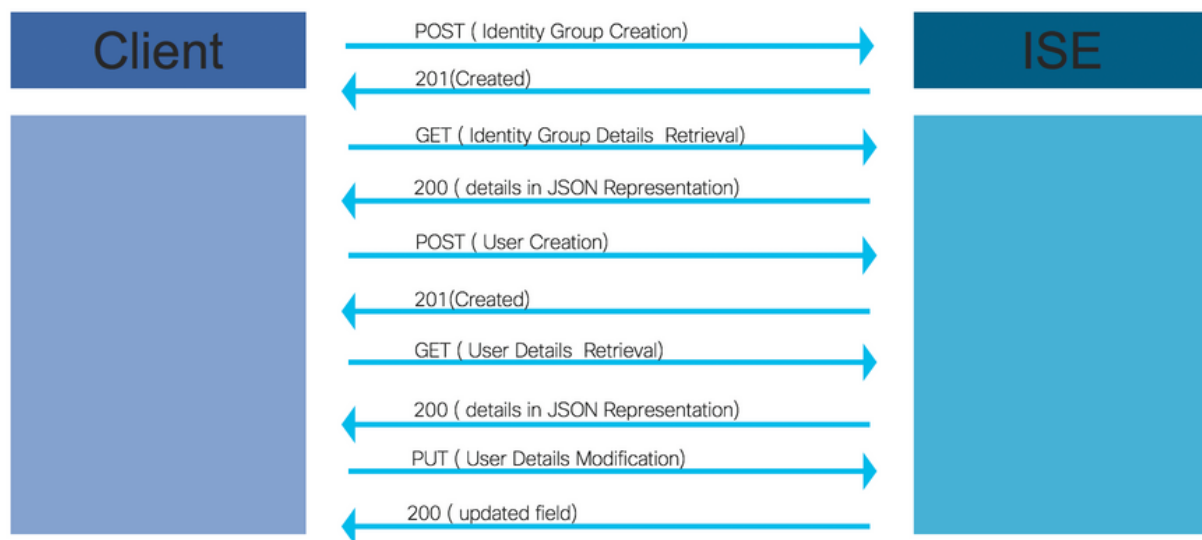
Componenti usati

Il presente documento non è limitato a versioni software e hardware specifiche ed è solo un esempio di guida alla configurazione tramite l'API REST.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Configurazione

API Request and Response flow



Di seguito vengono riportati alcuni passaggi di esempio per le operazioni tramite API che possono essere utilizzate come riferimento per creare chiamate personalizzate.

1. Creazione del gruppo di identità

Creare un gruppo di identità con l'aiuto del metodo **POST**.

URL per chiamata API:

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

Intestazione per chiamata API:

Intestazione 'Content-Type' application/json

HTTP:

Intestazione 'Accept' HTTP: application/json

Codice JSON per la creazione del gruppo di identità

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

Esempio:

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. Recupero dei dettagli del gruppo di identità

Recuperare i dettagli del gruppo di identità con l'aiuto del metodo **GET**.

URL per chiamata API:

https://<ISE IP>:9060/ers/config/identitygroup?filter=name.CONTAINS.<Nome gruppo di identità >

Intestazione per chiamata API:

Intestazione 'Content-Type' application/json

HTTP: application/json

Intestazione 'Accept' HTTP: application/json

Esempio:

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"
        }
      }
    ]
  }
}
```

Nota: Per creare utenti in questo gruppo di identità è necessario l'ID (ricevuto nei dettagli del gruppo di identità).

3. Creazione utente

Creare un utente con l'aiuto del metodo **POST**.

URL per chiamata API:

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

Intestazione per chiamata API:

Intestazione **tipo di contenuto** application/json
 HTTP:

Intestazione **accettazione** application/json
 HTTP:

Codice JSON per la creazione dell'utente:

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

Esempio:

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

- Request:** Method: POST, URL: `https://10.71.244.140:9060/ers/config/internaluser/`, Button: Send request
- Headers:**
 - Content-Type: application/json
 - Accept: application/json
- Basic auth:** Username: ersadmin, Password: [masked], Show password? checkbox is unchecked.
- Request body:** Type: Custom, Content:

```
{
  "InternalUser": {
    "name": "test123",
    "email": "test@cisco.com",
    "enabled": true,
    "password": "Letmein!1",
    "firstName": "test",
    "lastName": "test",
    "changePassword": false,
    "identityGroups": "b6ae7220-4289-11ea-a840-cee7c3fe1b0d",
  }
}
```
- Response (80.364s) - https://10.71.244.140:9060/ers/config/internaluser/:** Status: 201 Created

4. Recupero dei dettagli utente

Recuperare i dettagli dell'utente con l'aiuto del metodo **GET**.

URL per chiamata API:

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

Nota: Questo URL può essere utilizzato per filtrare gli utenti. L'utente può essere filtrato utilizzando `firstName`, `lastName`, `identityGroup`, `name`, `description`, `email`, `enabled`.

Questa opzione è consigliata per filtrare i dettagli dell'utente con l>ID e-mail, in quanto l>ID e-mail è univoco per ogni utente.

`https://<ISE IP>:9060/ers/config/internaluser?filter=<nome del campo utilizzato per il filtro>.CONTAINS.<Valore del campo per il filtro>`

Intestazione per chiamata API:

Intestazione **tipo di contenuto** application/json
HTTP:
Intestazione **accettazione** application/json
HTTP:

Esempio:

The screenshot shows a REST client interface. At the top, the 'Request' section is active. 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 an '+Add header' button. The 'Basic auth' section is also expanded, showing the username 'ersadmin' and a masked password field with a 'Show password?' checkbox. Below the request section, the 'Response' section shows a status of '200 OK' and the URL. The response body is a JSON object:

```
{
  "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"
        }
      }
    ]
  }
}
```

Nota: ID e nomi ricevuti sono necessari per aggiornare la password o altre informazioni per un utente. href URL verrà utilizzato per aggiornare le informazioni utente.

5. Modifica dettagli utente

Modificare la password utente con l'aiuto **del** metodo **PUT**.

URL per chiamata API:

https://<ISE IP>:9060/ers/config/internaluser/<ID utente ricevuto tramite il processo descritto nel passaggio 4>

L'URL di cui sopra è href ricevuto utilizzando il processo descritto nel passaggio 4.

Intestazione per chiamata API:

Intestazione **tipo di contenuto** application/json
HTTP:
Intestazione **accettazione** application/json
HTTP:

Codice JSON per la modifica delle credenziali utente:

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

Esempio:

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

- Request:** Method: PUT, URL: 140:9060/ers/config/internaluser/be9aee27-22f2-4a77-a8e9-3092ba3b636e, Send request button.
- Headers:** Content-Type: application/json, Accept: application/json. Includes an "Add header" button.
- Basic auth:** Username: ersadmin, Password: [masked]. Includes a "Show password?" checkbox.
- Request body:** Type: Custom. JSON body:

```
{  "InternalUser": {    "id": "be9aee27-22f2-4a77-a8e9-3092ba3b636e",    "name": "test123",    "password": "Letmein!3",    "enablePassword": "Letmein!3"  }}
```
- Response (2.177s):** Status: 200 OK, URL: https://10.71.244.140:9060/ers/config/internaluser/be9aee27-22f2-4a77-a8e9-3092ba3b636e

Verifica

Per verificare i gruppi di identità, passare a **Amministrazione > Gestione delle identità > Gruppi > Gruppi di identità > Gruppi di identità utente** nella GUI ISE.

Per verificare gli utenti, selezionare **Amministrazione > Gestione delle identità > Identità > Utenti** nella GUI ISE.