

ISE Identity Group, création et modification d'utilisateurs via l'API de repos

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Components Used](#)

[Configuration](#)

[1. Création du groupe d'identités](#)

[2. Récupération des détails du groupe d'identités](#)

[3. Création d'utilisateurs](#)

[4. Récupération des détails utilisateur](#)

[5. Modification des détails utilisateur](#)

[Vérification](#)

Introduction

Ce document décrit comment créer et modifier des groupes d'identité et des utilisateurs à l'aide de l'API Rest qui peut être utilisée pour l'automatisation de la gestion des identités. La procédure décrite dans ce chapitre est basée sur un exemple de déploiement ISE autonome et de client Firefox Rest API (RESTED) au format JSON.

Conditions préalables

Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

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

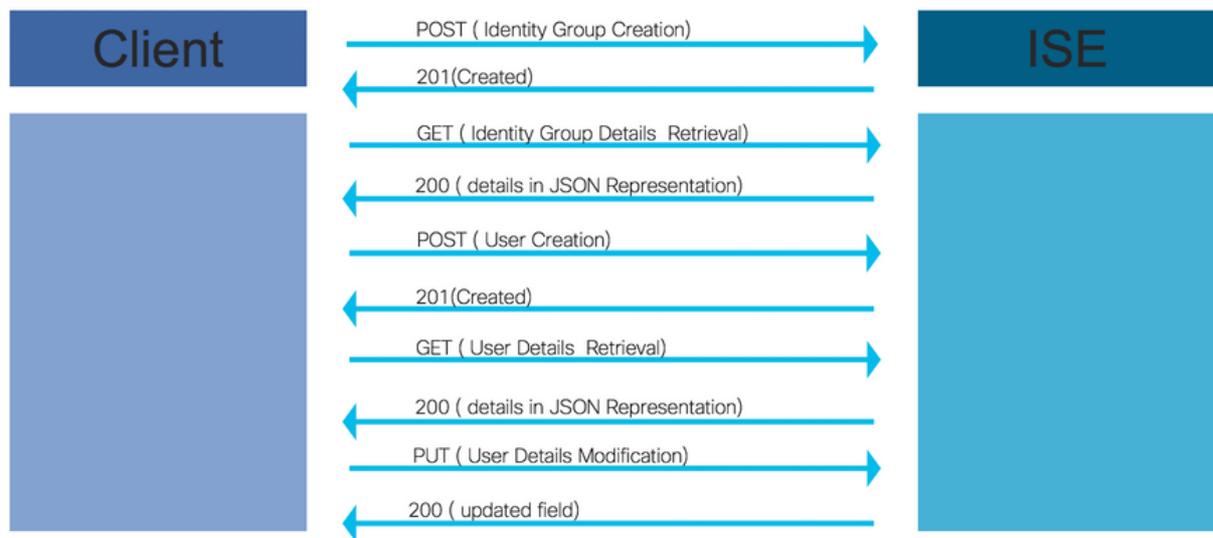
Components Used

Ce document n'est pas limité à des versions logicielles et matérielles spécifiques et n'est qu'un exemple de guide de configuration via l'API REST.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

Configuration

API Request and Response flow



Il s'agit des étapes types pour les opérations via l'API qui peuvent être utilisées comme référence pour générer vos propres appels.

1. Création du groupe d'identités

Créez un groupe d'identités à l'aide de la méthode **POST**.

URL de l'appel API :

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

En-tête de l'appel API :

En-tête HTTP 'Content-Type' : application/json

En-tête HTTP 'Accepter' : application/json

Code JSON pour la création de groupe d'identités

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

Exemple :

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. Récupération des détails du groupe d'identités

Récupère les détails du groupe d'identités à l'aide de la méthode **GET**.

URL de l'appel API :

https://<IP ISE>:9060/ers/config/identitygroup?filter=name.CONTAINS.<Nom du groupe d'identités >

En-tête de l'appel API :

En-tête HTTP 'Content-Type' : application/json
En-tête HTTP 'Accepter' : application/json

Exemple :

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"
        }
      }
    ]
  }
}
```

Note: L'ID (reçu dans les détails du groupe d'identités) est requis pour créer des utilisateurs dans ce groupe d'identités.

3. Création d'utilisateurs

Créez un utilisateur à l'aide de la méthode **POST**.

URL de l'appel API :

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

En-tête de l'appel API :

En-tête HTTP **Content-Type** : application/json
 En-tête **d'acceptation** HTTP : application/json

Code JSON pour la création d'utilisateurs :

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

Exemple :

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": "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/

201 Created

Headers >

4. Récupération des détails utilisateur

Récupérez les détails de l'utilisateur à l'aide de la méthode **GET**.

URL de l'appel API :

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

Note: Cette URL peut être utilisée pour filtrer les utilisateurs. L'utilisateur peut être filtré à l'aide de firstName, lastName, identityGroup, name, description, email, enabled.

Il est recommandé de filtrer les détails de l'utilisateur avec l'ID d'e-mail car l'ID d'e-mail est unique pour chaque utilisateur.

https://<IP ISE>:9060/ers/config/internaluser?filter=<nom du champ utilisé pour le filtrage>.CONTAINS.<Valeur du champ pour le filtrage>

En-tête de l'appel API :

En-tête HTTP **Content-Type** : application/json
En-tête d'**acceptation** HTTP : application/json

Exemple :

Request ↗ ⚙ +

GET 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"
        }
      }
    ]
  }
}
```

Remarque : IDandNamereceived est requis pour mettre à jour le mot de passe ou d'autres informations pour un utilisateur.hrefURL sera utilisé pour mettre à jour les informations utilisateur.

5. Modification des détails utilisateur

Modifiez le mot de passe utilisateur à l'aide de la méthode **PUT**.

URL de l'appel API :

https://<IP ISE>:9060/ers/config/internaluser/<ID utilisateur reçu à l'aide du processus décrit à l'étape 4>

L'URL href ci-dessus est reçue à l'aide du processus décrit à l'étape 4.

En-tête de l'appel API :

En-tête HTTP **Content-Type** : application/json
En-tête **d'acceptation** HTTP : application/json

Code JSON pour la modification des informations d'identification utilisateur :

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

Exemple :

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

- 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.
- Basic auth:** Username: ersadmin, Password: [masked], Show password? checkbox.
- Request body:** Type: Custom, JSON payload:

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

Vérification

Pour vérifier les groupes d'identités, accédez à **Administration > Identity Management > Groups > Identity Groups > User Identity Groups** dans l'interface utilisateur graphique ISE.

Pour vérifier que les utilisateurs naviguent vers **Administration > Identity Management > Identities > Users** in ISE GUI.