# ISE Identity-Group, User Creation and Modification through Rest API

## Contents

Introduction
Prerequisites
Requirements
Components Used
Configure
1. Identity Group Creation
2. Identity Group Details Retrieval
3. User Creation
4. User Details Retrieval
5. User Details Modification
Verify

## Introduction

This document describes how to create and modify identity groups and users using Rest API which can be used for identity management automation. The procedure described in this chapter is based on sample standalone ISE deployment and Rest API Firefox Client (RESTED) in JSON format.

## Prerequisites

### Requirements

Cisco recommends that you have knowledge of these topics:

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

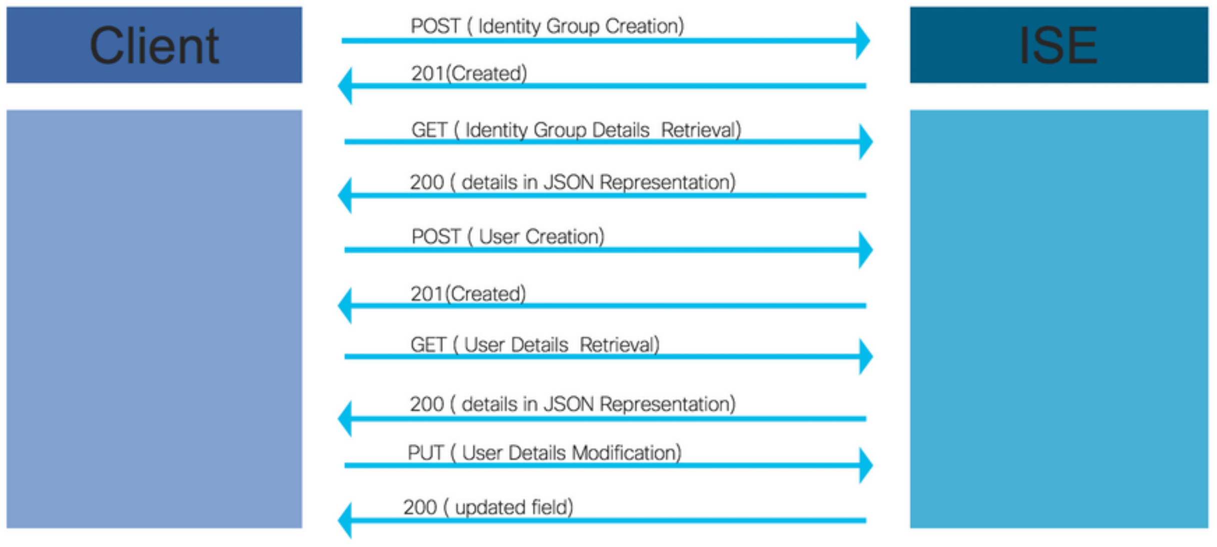
### Components Used

This document is not restricted to specific software and hardware versions and is only a sample configuration guide through REST API.

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. If your network is live, ensure that you understand the potential impact of any command.

## Configure

API Request and Response flow


Client
ISE
POST ( Identity Group Creation)
201(Created)
GET ( Identity Group Details Retrieval)
200 ( details in JSON Representation)
POST ( User Creation)
201(Created)
GET ( User Details Retrieval)
200 ( details in JSON Representation)
PUT ( User Details Modification)
200 ( updated field)

These are the sample steps for operations through API which can be used as a reference to build your own calls.

## 1. Identity Group Creation

Create an Identity Group with the help of the **POST** method.

URL for API Call:

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

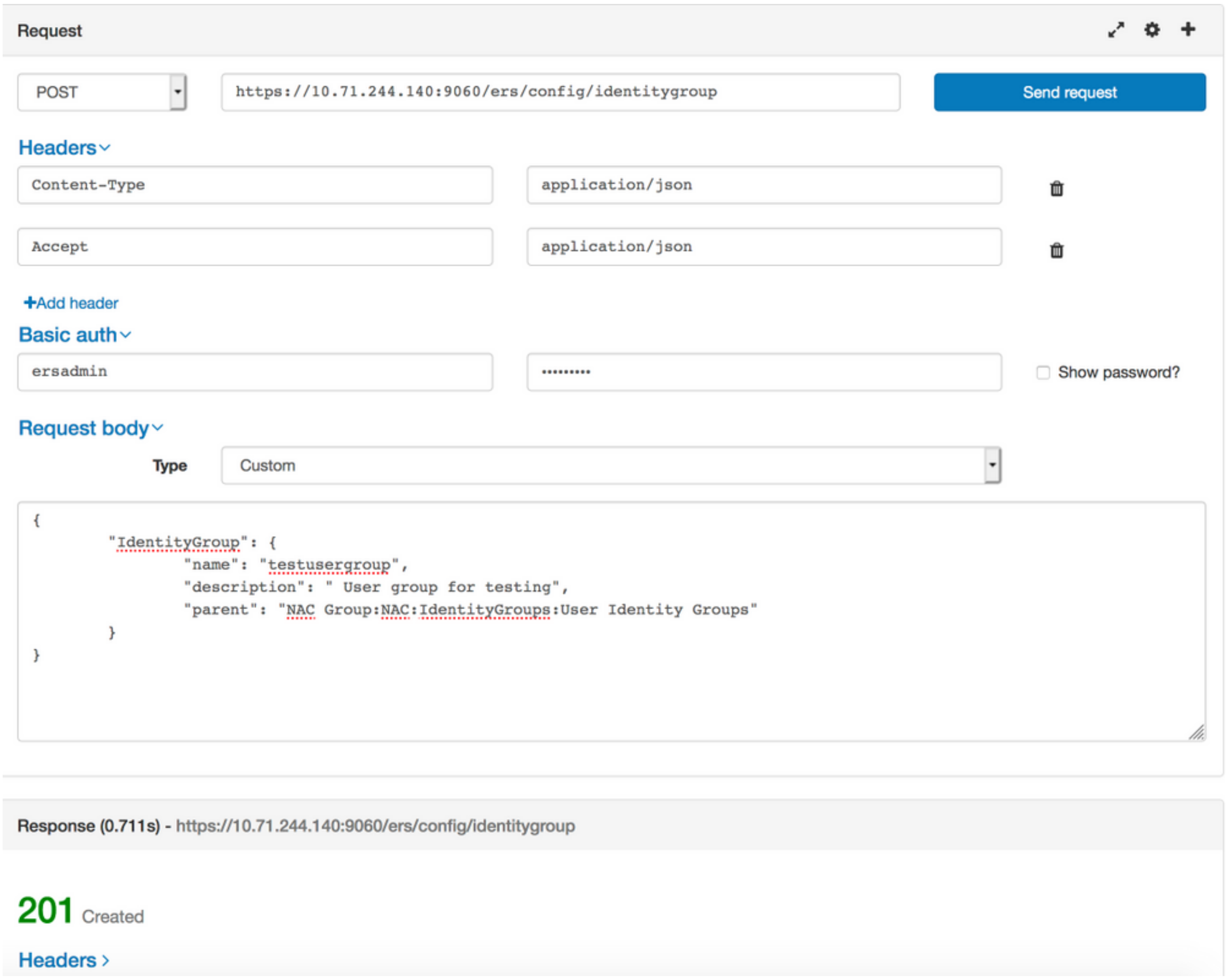
The header for API Call:

| | |
|---|---|
| HTTP 'Content-Type' Header: | application/json |
| HTTP 'Accept' Header: | application/json |

JSON code for Identity Group Creation

```
{ "IdentityGroup": { "name": "<Identity Group Name>", "description": "<Indentity Group
Description>", "parent": "NAC Group:NAC:IdentityGroups:User Identity Groups" } }
```

Example:



```
https://10.71.244.140:9060/ers/config/identitygroup
```

Headers:

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

Basic auth: ersadmin / •••••••••

```
{
        "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

## 2. Identity Group Details Retrieval

Fetch the Identity Group details with the help of**GET** method.

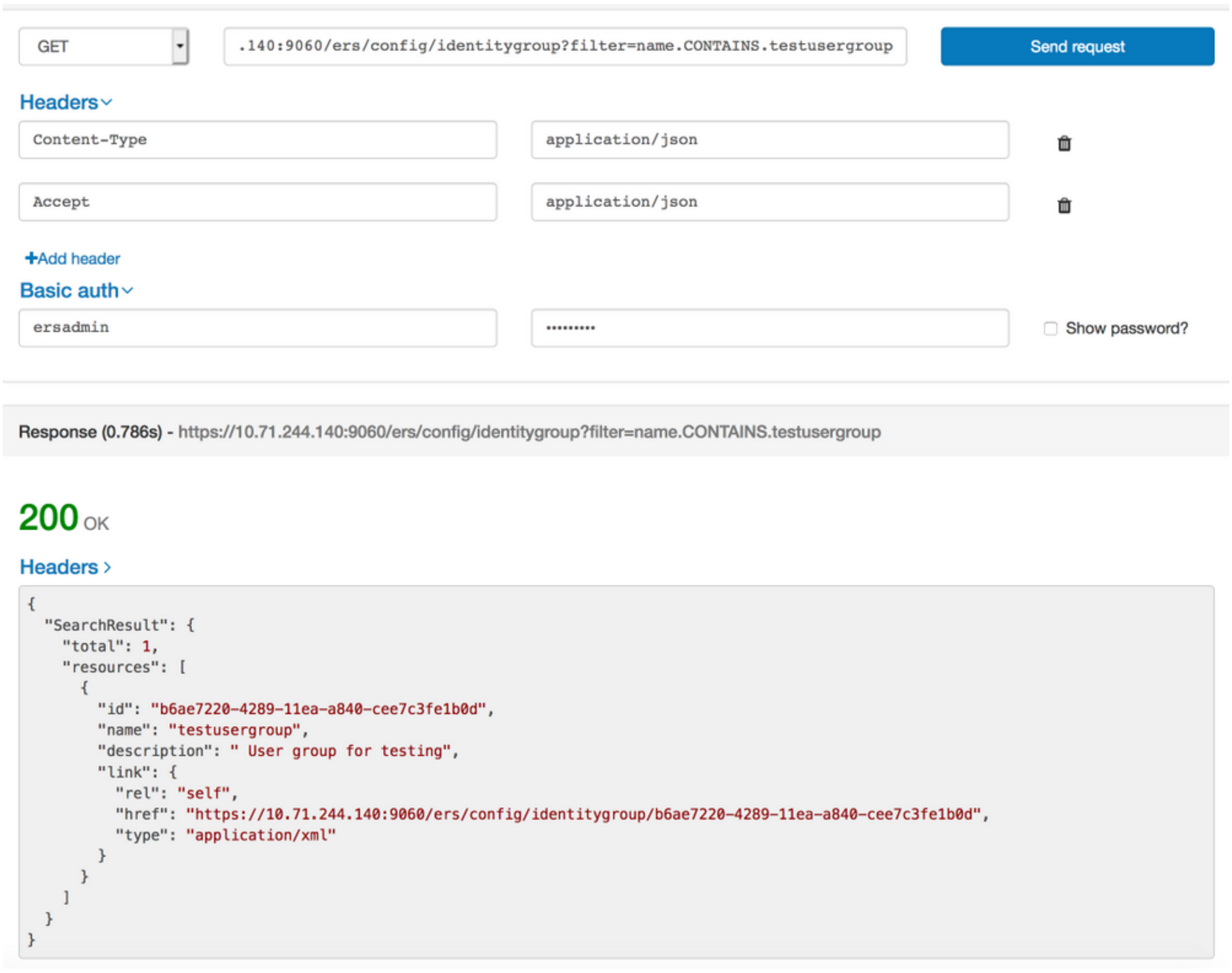
URL for API Call:

**https://<ISE IP>:9060/ers/config/identitygroup?filter=name.CONTAINS.<Identity Group Name >**

The header for API Call:

| | |
|---|---|
| HTTP 'Content-Type' Header: | application/json |
| HTTP 'Accept' Header: | application/json |

Example:



```
.140:9060/ers/config/identitygroup?filter=name.CONTAINS.testusergroup
Content-Type    application/json
Accept          application/json
ersadmin        •••••••••
```

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

200 OK

```
{
  "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**: ID (received in Identity Group details) is required to create users in this Identity Group.

## 3. User Creation

Create a User with the help of**POST** Method.

URL for API Call:

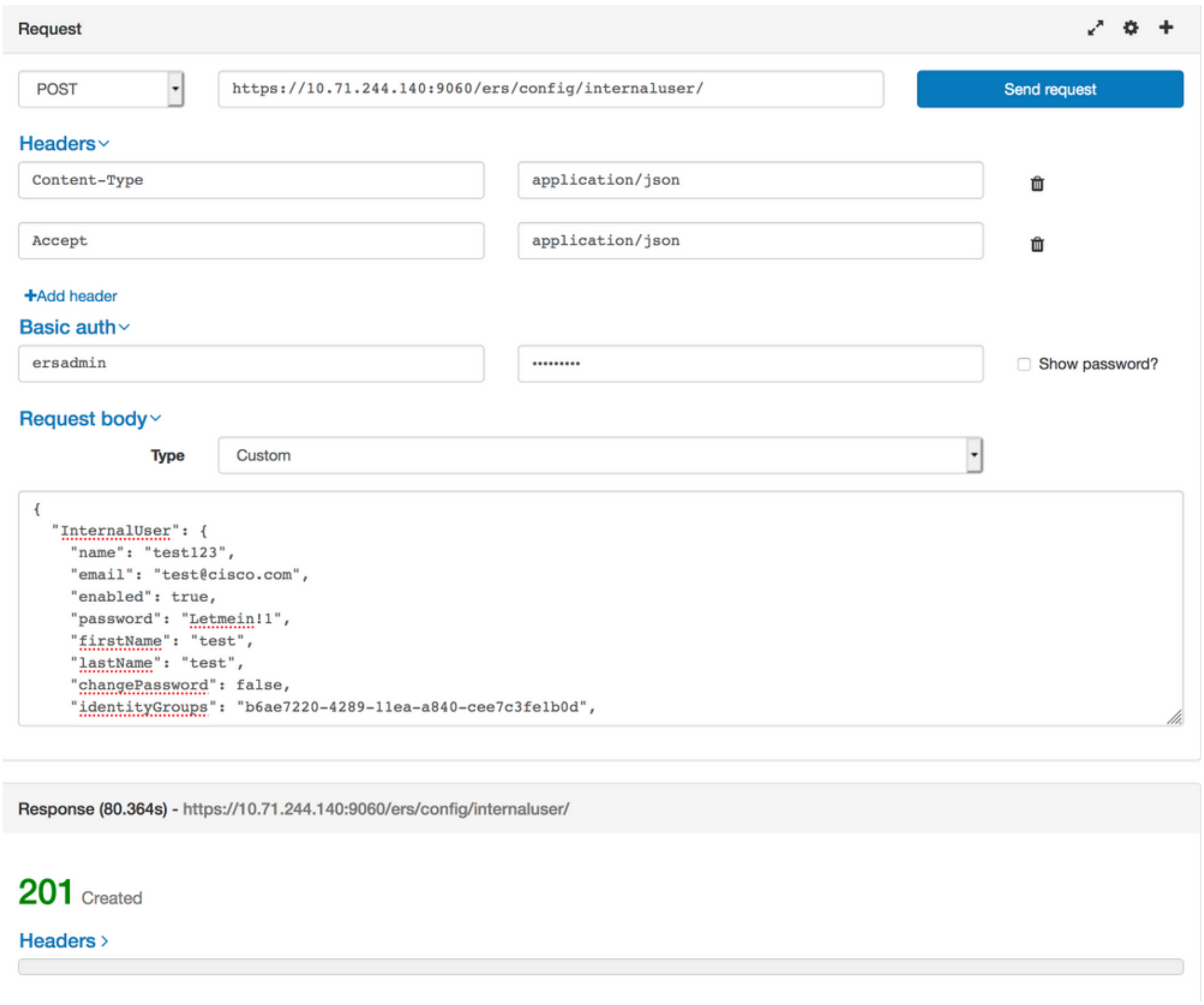
**https://<ISE IP>:9060/ers/config/internaluser/**

The header for API Call:

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

JSON code for User Creation:

```
{ "InternalUser": { "name": "<username>", "email": "<emailid of user>", "enabled": true,
"password": "<password>", "firstName": "<first name of user>", "lastName": "<Last name of
user>", "changePassword": false, "identityGroups": "<ID of Identity Group, which user will be
part of and can be received as described in step 2>", "expiryDateEnabled": false,
"enablePassword": "<Enable Password>" } }
```
Example:



POST https://10.71.244.140:9060/ers/config/internaluser/

Headers:
Content-Type: application/json
Accept: application/json

Basic auth: ersadmin / •••••••••

Type: Custom

```
{
  "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. User Details Retrieval

Fetch User details with the help of **GET** Method.

URL for API Call:

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

> **Note**: This URL Can be used to filter Users. User can be filtered using firstName, lastName, identityGroup, name, description, email, enabled.

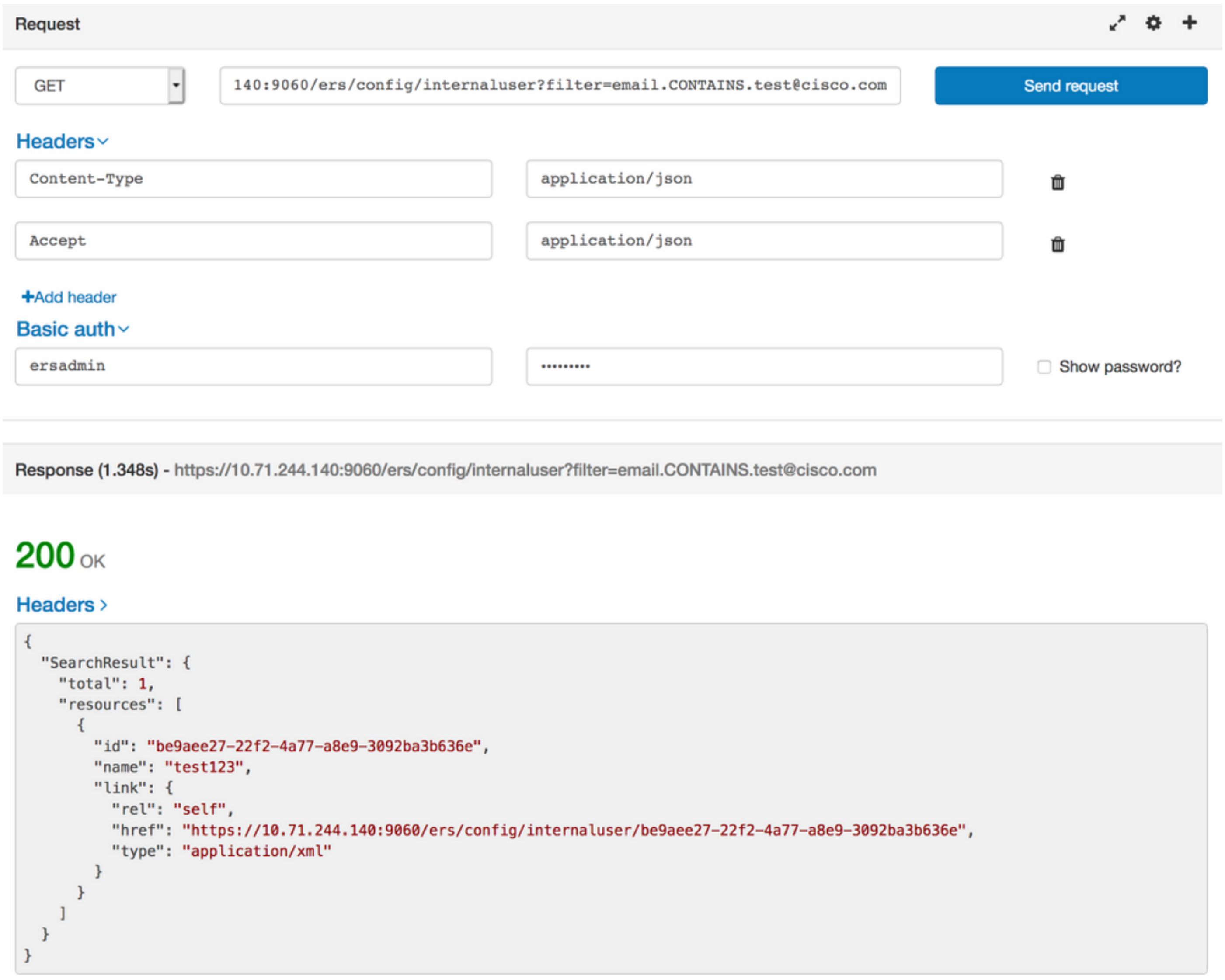
This is recommended to filter user details with email ID as email id is unique for each user.

**https://<ISE IP>:9060/ers/config/internaluser?filter=<name of the field used for filtering>.CONTAINS.<Value of the field for filtering>**

The header for API Call:

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

Example:



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 ••••••••• 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"
        }
      }
    ]
  }
}
```

**Note**:**ID**and**Name**received here is required to update the password or other information for a user.**href**URL will be used to update user information.

## 5. User Details Modification

Modify user password with the help of**PUT** Method.

URL for API Call:

**https://<ISE IP>:9060/ers/config/internaluser/<User ID received using  process described in step 4>**

Above is href URL received using the process described in Step 4.

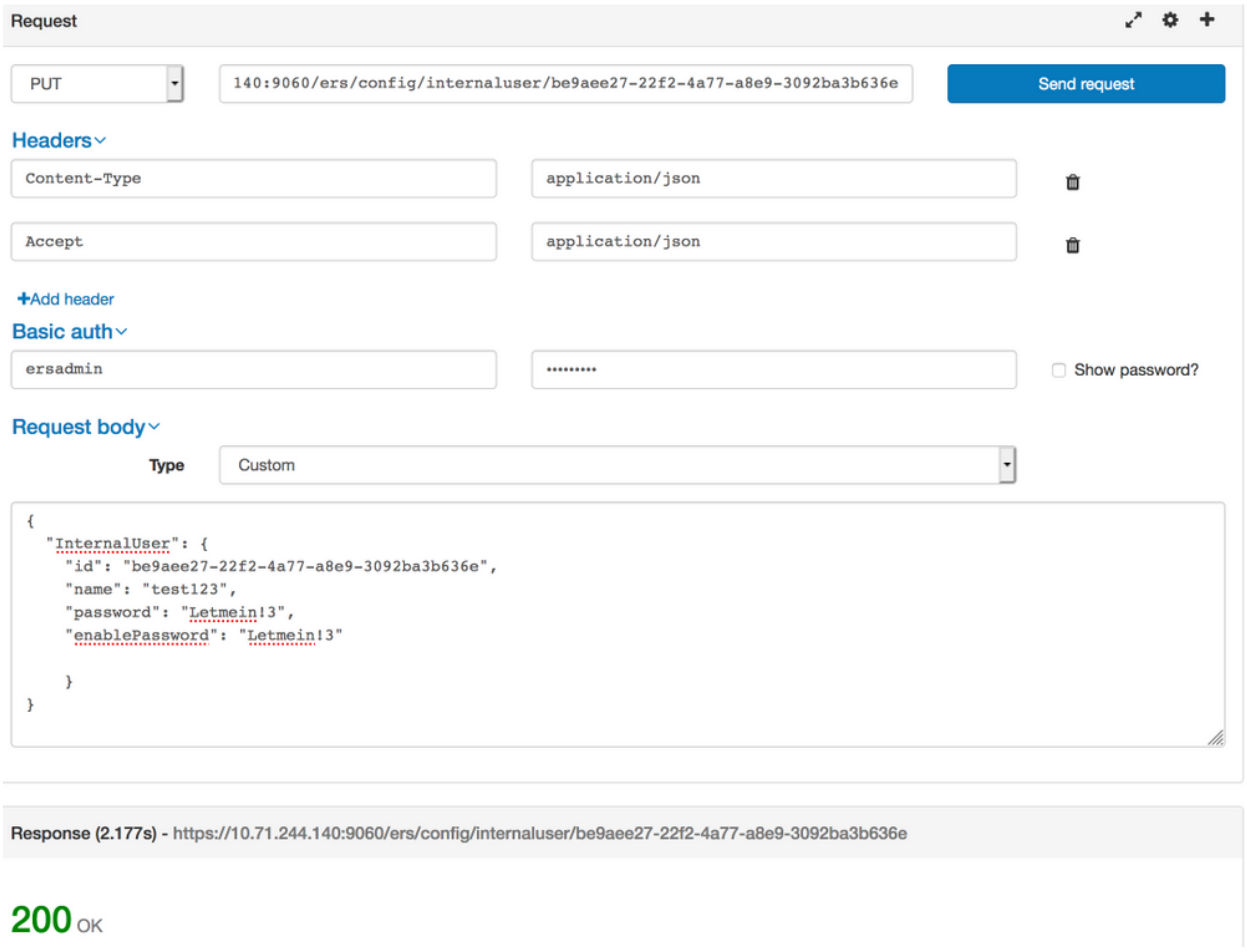
The header for API Call:

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

JSON code for User credential Modification:

```
{ "InternalUser": { "id": "<ID of user which needs password change>", "name": " <user name which
needs password change>", "password": "<New Password>", "enablePassword": "<New Enable Password>"
} }
```

Example:



140:9060/ers/config/internaluser/be9aee27-22f2-4a77-a8e9-3092ba3b636e

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

Type: Custom

```
{
  "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

200 OK

## Verify

To Verify identity groups navigate to**Administration > Identity Management > Groups > Identity Groups > User Identity Groups**in ISE GUI.

To Verify users navigate to**Administration > Identity Management > Identities > Users**in ISE GUI.