



External RESTful Services API Operations

- [Overview, page 7-1](#)
- [Prerequisites for Using the External RESTful Services API Calls, page 7-1](#)
- [GetVersion, page 7-2](#)
- [External RESTful Services APIs for Internal Users, page 7-2](#)
- [External RESTful Services APIs for Endpoints, page 7-8](#)
- [External RESTful Services APIs for Endpoint Certificates, page 7-16](#)
- [External RESTful Services APIs for Endpoint Identity Groups, page 7-18](#)
- [External RESTful Services APIs for Identity Groups, page 7-22](#)
- [External RESTful Services APIs for Guest Users, page 7-24](#)
- [External RESTful Services APIs for Portals, page 7-43](#)
- [External RESTful Services APIs for Profiles, page 7-46](#)
- [External RESTful Services APIs for Network Devices, page 7-47](#)
- [External RESTful Services APIs for Network Device Groups, page 7-52](#)
- [External RESTful Services APIs for SGTs, page 7-55](#)
- [REST API Client, page 7-57](#)

Overview

This chapter provides examples of the External RESTful Services API calls, and describes how to use them. Instructions are provided for issuing the External RESTful Services API calls, as well as examples of API output schema files and sample data returned.

Prerequisites for Using the External RESTful Services API Calls

You must fulfill the following prerequisites before invoking an External RESTful Services API call:

- You must have enabled External RESTful Services from the GUI.
- You must have External RESTful Services Admin privileges.

You can use any REST client like JAVA, curl linux command, python or any other client to invoke External RESTful Services API calls.

Related Topics

- [Enabling External RESTful Services APIs from the GUI, page 5-2](#)
- [External RESTful Services API Authentication and Authorization, page 5-2](#)

GetVersion

The GetVersion operation is common to all available resources. It fetches the version information of the required resource. The following table lists the main characteristics of this operation:

Table 7-1 Main Characteristics of GetVersion Operation

Description	Retrieve the version information of the specified resource
Synopsis	GET/ers/config/<resource-name>/versioninfo
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	Version information
Response Status	200, 400, 401,403,404,415,500

Sample Request for GetVersion Operation

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/<resource-type>/versioninfo
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.<resource-namespace>.1.0+xml
```

Sample Response for GetVersion Operation

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.<resource-namespace>.versioninfo.1.0+xml
Content-Length: 122
{
  <?xml version="1.0" encoding="UTF-8"?>
  <ns2:versionInfo xmlns:ns2="ers.ise.cisco.com">
    <currentServerVersion>1.2</currentServerVersion>
    <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/<resource-type>/versioninfo" rel="self"/>
    <supportedVersions>1.0,1.1,1.2,1.3</supportedVersion>
  </ns2:versionInfo>
}
```

External RESTful Services APIs for Internal Users

The External RESTful Services APIs for Internal users support full CRUD functionality. The following table lists the External RESTful Services APIs that are available for internal users:

Table 7-2 External RESTful Services APIs Available For Internal Users

Operation	HTTP Method	URL	Content	QueryString
Get All Users	GET	/ers/config/internaluser	n/a	Page, Size, sortacs or sortdsn, Filter
Get User	GET	/ers/config/internaluser/{internal user-id ¹ }	n/a	
Create User	POST	/ers/config/internaluser/	internaluser	
Update User	PUT	/ers/config/internaluser/{internal user-id}	internaluser	
Delete User	DELETE	/ers/config/internaluser/{internal user-id}	n/a	
Get InternalUser Resource Version Info	GET	/ers/config/internaluser/version	n/a	

1. Internal user ID is the UUID type as stored in the Cisco ISE database.

Retrieve All Internal Users

You can use this API call to retrieve all the internal users present in Cisco ISE. The following table lists the main characteristics of this API call:

Table 7-3 Main Characteristics of Retrieve All Internal Users API Call

Description	Retrieve collection of internal users
Synopsis	GET /ers/config/internaluser
Request Headers	Accept, Authorization, Host
QueryString	page, size, sortbyacn, sortbydcn, filter
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	SearchResult
Response Status	200, 400, 401, 403, 404, 415, 429, 500

Sample Request for Retrieve All Internal Users API

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/internaluser?page=0&size=20&sortacs=name
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.internaluser.1.0+xml
```

Sample Response for Retrieve All Internal Users API

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.searchresult.1.0+xml
```

```

Content-Length: 16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult total="2" xmlns:ns2="ers.ise.cisco.com">
  <resources>
    <ns2:resource description="description1" name="name1" id="id1"/>
    <ns2:resource description="description2" name="name2" id="id2"/>
  </resources>
</ns2:searchResult>
}

```

Get Internal Users by ID

You can use this API call to get an internal user by the ID in Cisco ISE. The following table lists the main characteristics of this API call:

Table 7-4 Main Characteristics of Read Internal Users API Call

Description	Retrieve the specified internal user
Synopsis	GET /ers/config/internaluser/{id}
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	Resource of type InternalUser
Response Status	200, 400, 401,403, 404, 415, 429, 500

Sample Request for Read Internal Users API

```

GET https://<ISE-ADMIN-NODE>:9060/ers/config/internaluser/333
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.internaluser.1.0+xml

```

Sample Response for Read Internal Users API

```

HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.identity.internaluser.1.0+xml
Content-Length: 16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:internaluser description="description" name="name" id="id"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
  <changePassword>true</changePassword>
  <customAttributes>
    <entry>
      <key>key2</key>
      <value>value3</value>
    </entry>
    <entry>
      <key>key1</key>
      <value>value1</value>
    </entry>
  </customAttributes>

```

```

    </customAttributes>
    <email>email@example.com</email>
    <enabled>true</enabled>
    <firstName>John</firstName>
    <identityGroups>identityGroups</identityGroups>
    <lastName>Doe</lastName>
    <password>12345</password>
  </ns3:internaluser>
}

```

Create Internal Users

You can use this API call to create internal users in Cisco ISE. Password is mandatory for creating internal users using External RESTful Services APIs. The following table lists the main characteristics of this API call:

Table 7-5 Main Characteristics of Create Internal Users API Call

Description	Create the specified internal user
Synopsis	POST /ers/config/internaluser/
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	InternalUser
Response Headers	Content-Length, Content-Type, Location
Response Message Body	Resource of type InternalUser
Response Status	201, 400, 401, 403, 415, 429, 500

Sample Request for Create Internal Users API

```

POST https://<ISE-ADMIN-NODE>:9060/ers/config/internaluser/
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.identity.internaluser.1.0+xml
{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <ns3:internaluser description="description" name="name" id="id"
  xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
    <changePassword>true</changePassword>
    <customAttributes>
      <entry>
        <key>key2</key>
        <value>value3</value>
      </entry>
      <entry>
        <key>key1</key>
        <value>value1</value>
      </entry>
    </customAttributes>
    <email>email@example.com</email>
    <enabled>true</enabled>
    <firstName>John</firstName>
    <identityGroups>identityGroups</identityGroups>
    <lastName>Doe</lastName>
    <password>12345</password>
  </ns3:internaluser>
}

```

```
</ns3:internaluser>
}
```

Sample Response for Create Internal Users API

```
HTTP/1.1 201 OK (see the location header for the new user's ID)
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.identity.internaluser.1.0+xml
Location: https://<ISE-ADMIN-NODE>/ers/config/internaluser/444
```

Update Internal Users

You can use this API call to update internal users in Cisco ISE. You must set the password as '*****', if the password is not getting changed while updating the internal users using the External RESTful Services APIs. The following table lists the main characteristics of this API call:

Table 7-6 Main Characteristics of Update Internal Users API Call

Description	Update the specified internal user
Synopsis	PUT /ers/config/internaluser/{id}
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	InternalUser
Response Headers	Content-Length, Content-Type
Response Message Body	List of updated fields
Response Status	200, 400, 401, 403, 404, 415, 429, 500

Sample Request for Update Internal Users API

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/internaluser/333
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.identity.internaluser.1.0+xml
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:internaluser description="description" name="name" id="333"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
  <changePassword>true</changePassword>
  <customAttributes>
    <entry>
      <key>key2</key>
      <value>value3</value>
    </entry>
    <entry>
      <key>key1</key>
      <value>value1</value>
    </entry>
  </customAttributes>
  <email>email@example.com</email>
  <enabled>true</enabled>
  <firstName>John</firstName>
  <identityGroups>IdentityGroups</identityGroups>
  <lastName>Doe</lastName>
  <password>*****</password>
```

```
</ns3:internaluser>
}
```

Sample Response for Update Internal Users API

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.updatedfields.1.0+xml
Content-Length: 529
{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <ns2:updatedFields xmlns:ns2="ers.ise.cisco.com">
    <updatedField field="lastname">
      <newValue>Doe</newValue>
      <oldValue>name</oldValue>
    </updatedField>
    <updatedField field="identityGroups">
      <newValue>IdentityGroups</newValue>
      <oldValue>zzz</oldValue>
    </updatedField>
  </ns2:updatedFields>
}
```

Delete Internal Users

You can use this API call to delete internal users from Cisco ISE. The following table lists the main characteristics of this API call:

Table 7-7 Main Characteristics of Delete Internal Users API Call

Description	Delete the specified internal user
Synopsis	DELETE /ers/config/internaluser/{internaluser-id}
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	Resource of type InternalUser
Response Status	204, 400, 401, 403, 404, 415, 429, 500

Sample Request for Delete Internal Users API

```
DELETE https://<ISE-ADMIN-NODE>:9060/ers/config/internaluser/333
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.internaluser.1.0+xml
```

Sample Response for Delete Internal Users API

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
```

External RESTful Services APIs for Endpoints

The following table lists the External RESTful Services APIs for end points:

Table 7-8 External RESTful Services APIs Available for Endpoints

Operation	Method	URL	Content	QueryString
Get All Endpoints	GET	/ers/config/endpoint	n/a	page, size, sortacs or sortdsn, filter
Get Endpoint	GET	/ers/config/endpoint/{id ¹ }	n/a	
Create Endpoint	POST	/ers/config/endpoint/	endpoint	
Update Endpoint	PUT	/ers/config/endpoint/{id}	endpoint	
Delete Endpoint	DELETE	/ers/config/endpoint/{id}	n/a	
Register Endpoint	PUT ²	/ers/config/endpoint/register	endpoint	
Deregister Endpoint	PUT	/ers/config/endpoint/{id}/deregister	n/a	
Get Endpoint Resource Version Info	GET	/ers/config/endpoint/version	n/a	

1. Endpoint ID is the UUID type as stored in the Cisco ISE database.
2. If the endpoint already exists, it will be registered. If it does not exist, it will be first created and then registered. In both the scenarios, the return status will be 204.

Get All Endpoints

The Get All API for Endpoints works only for retrieving endpoints associated to the user specified in the filter. The following table lists the main characteristics of this API call:

Table 7-9 Main Characteristics of Get All Endpoints API Call

Description	Retrieve collection of endpoints associated to the specified internal user
Synopsis	GET /ers/config/endpoint/
Request Headers	Accept, Authorization, Host
QueryString	page, size, sortbyacn, sortbydcn, filter
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	SearchResult
Response Status	200, 400, 401, 403, 404, 415, 429, 500

Sample Request for Get All Endpoints API

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint?filter=userid.EQ.123
```



```
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
```

Sample Response for Get All Endpoints API

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.searchresult.1.0+xml
Content-Length: 16347
{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <ns2:searchResult total="2" xmlns:ns2="ers.ise.cisco.com">
  <resources>
  <resource name=name1 id="id1">
    <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/id1" rel="self"/>
  </resource>
  <resource name="name2" id="id2">
    <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/id2" rel="self"/>
  </resource>
  </resources>
  </ns2:searchResult>
  }
```

Get Endpoints by ID

You can use this API call to get an endpoint by the ID in Cisco ISE. The following table lists the main characteristics of this API call:

Table 7-10 Main Characteristics of Read Endpoints API Call

Description	Retrieve the specified endpoint
Synopsis	GET /ers/config/endpoint/{endpoint-id}
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	Resource of type InternalUser
Response Status	200, 400, 401,403, 404, 415, 429, 500

Sample Request for Read Endpoints API

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/333
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
```

Sample Response for Read Endpoints API

```
HTTP/1.1 200 OK
```

```

Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
Content-Length: 16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:endpoint description="description" name="name" id="id" xmlns:ns2="ers.ise.cisco.com"
xmlns:ns3="identity.ers.ise.cisco.com">
  <group>group</group>
  <groupId>groupId</groupId>
  <identityStore>identityStore</identityStore>
  <identityStoreId>identityStoreId</identityStoreId>
  <mac>00:01:02:03:04:05</mac>
  <portalUser>portalUser</portalUser>
  <profile>profile</profile>
  <profileId>profileId</profileId>
  <staticGroupAssignment>true</staticGroupAssignment>
  <staticProfileAssignment>false</staticProfileAssignment>
</ns3:endpoint>
}

```

Create Endpoints

You can use this API call to create endpoints in Cisco ISE. The following table lists the main characteristics of this API call:

Table 7-11 Main Characteristics of Create Endpoints API Call

Description	Create the specified endpoint
Synopsis	POST /ers/config/endpoint/
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	Resource of type InternalUser
Response Status	200, 400, 401,403, 404, 415, 429, 500

Note the following while creating an endpoint:

- If you are creating an endpoint with Group static assignment, *groupid* is mandatory.
- If you are creating an endpoint with Profile static assignment, *profileid* is mandatory.
- If you are creating an endpoint with both Group and Profile static assignment, both *groupid* and *profileid* are required.

If these attributes are not included in the request, endpoint may not be created properly.

Sample Request for Create Endpoints API

```

POST https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

```

```

<ns3:endpoint description="description" name="name" id="id" xmlns:ns2="ers.ise.cisco.com"
xmlns:ns3="identity.ers.ise.cisco.com">
  <group>group</group>
  <groupId>groupId</groupId>
  <identityStore>identityStore</identityStore>
  <identityStoreId>identityStoreId</identityStoreId>
  <mac>00:01:02:03:04:05</mac>
  <portalUser>portalUser</portalUser>
  <profile>profile</profile>
  <profileId>profileId</profileId>
  <staticGroupAssignment>true</staticGroupAssignment>
  <staticProfileAssignment>false</staticProfileAssignment>
</ns3:endpoint>
}

```

Sample Response for Create Endpoints API

```

HTTP/1.1 201 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
Location: https://cisco.com/ers/config/endpoint/444

```

Update Endpoints

You can use this API call to update endpoints in Cisco ISE. The following table lists the main characteristics of this API call:

Table 7-12 Main Characteristics of Update Endpoints API Call

Description	Update the specified endpoint
Synopsis	PUT /ers/config/endpoint/{endpoint-id}
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	Resource of type InternalUser
Response Status	200, 400, 401, 403, 404, 415, 429, 500

Sample Request for Update Endpoints API

```

PUT https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/333
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <ns2:endpoint xmlns:ns2="identity.ers.ise.cisco.com" description="updated"
name="Endpoint">
    <staticGroupAssignment>false</staticGroupAssignment>
    <staticProfileAssignment>false</staticProfileAssignment>
  </ns2:endpoint>
}

```

Sample Response for Update Endpoints API

```

HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.updatedfields.1.0+xml
Content-Length: 529
{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <ns2:updatedFields xmlns:ns2="ers.ise.cisco.com">
    <updatedField field="staticGroupAssignment">
      <newValue>>false</newValue>
      <oldValue>>true</oldValue>
    </updatedField>
    <updatedField field="staticProfileAssignment">
      <newValue>>false</newValue>
      <oldValue>>true</oldValue>
    </updatedField>
  </ns2:updatedFields>
}

```

Delete Endpoints

You can use this API call to update endpoints in Cisco ISE. The following table lists the main characteristics of this API call:

Table 7-13 Main Characteristics of Delete Endpoints API Call

Description	Delete the specified endpoint
Synopsis	DELETE /ers/config/endpoint/{id}
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	Resource of type InternalUser
Response Status	200, 400, 401,403, 404, 415, 429, 500

Sample Request for Delete Endpoints API

```

DELETE https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/333
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.endpoint.1.0+xml

```

Sample Response for Delete Endpoints API

```

HTTP/1.1 204 No Content
Date: Thu, 12 Jul 2012 23:59:59 GMT

```

Register Endpoints

You can use this API call to register endpoints in Cisco ISE. The endpoint is created if it doesn't already exist. Similar to the GUI registration flow, the endpoint is statically assigned to the Registered Devices group and portal user and identity store will be set as specified in the content.

The following table lists the main characteristics of this API call:

Table 7-14 Main Characteristics of Register Endpoints API Call

Description	Register the specified endpoint
Synopsis	PUT /ers/config/endpoint/register
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	endpoint
Response Headers	Content-Length, Content-Type
Response Message Body	List of updated fields
Response Status	202, 400, 401, 403, 404, 415, 429, 500

Sample Request for Register Endpoints API

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/register
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <ns3:endpoint description="description" name="name" id="id" xmlns:ns2="ers.ise.cisco.com"
  xmlns:ns3="identity.ers.ise.cisco.com">
    <group>group</group>
    <groupId>groupId</groupId>
    <identityStore>identityStore</identityStore>
    <identityStoreId>identityStoreId</identityStoreId>
    <mac>00:01:02:03:04:05</mac>
    <portalUser>portalUser</portalUser>
    <profile>profile</profile>
    <profileId>profileId</profileId>
    <staticGroupAssignment>true</staticGroupAssignment>
    <staticProfileAssignment>false</staticProfileAssignment>
  </ns3:endpoint>
}
```

Sample Response for Register Endpoints API

```
HTTP/1.1 204 No Content
Date: Thu, 12 Jul 2012 23:59:59 GMT
```

Deregister Endpoints

You can use this API call to deregister endpoints in Cisco ISE. No content expected in the result. The following table lists the main characteristics of this API call:

Table 7-15 Main Characteristics of Deregister Endpoints API Call

Description	Deregister the specified endpoint
Synopsis	PUT /ers/config/endpoint/{id}/deregister
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	N/A
Response Message Body	N/A
Response Status	202, 400, 401, 403, 404, 415, 429, 500

Sample Request for Deregister Endpoint API Call

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/123/deregister
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
```

Sample Response for Deregister Endpoint API Call

```
HTTP/1.1 204 No Content
Date: Thu, 12 Jul 2012 23:59:59 GMT
```

Start Bulk Execution for Endpoints

A bulk execution allows you to send up to 500 CRUD operations of the same type in a single request.

If the request is valid, the server returns the status code 202 (ACCEPTED) and a unique bulk identifier in the LOCATION response header that you can use to track the bulk status using the Get Bulk Status operation.

Only one bulk is allowed to run at a time. If a bulk request was posted while another bulk is still running, the server will return with a response status 503 (Service Unavailable) with a corresponding descriptive message asking the client to try again later.

Table 7-16 Start Bulk Execution for Endpoints Main Characteristics

Description	Start Execute
Synopsis	PUT /ers/config/endpoint/bulk
Request Headers	Accept, Authorization, Host
QueryString	n/a
Request Message Body	BulkRequest
Response Headers	Content-Length, Content-Type
Response Message Body	n/a
Response Status	202, 400, 401, 403, 404, 415, 500

Sample Request for Start Bulk Execution for Endpoints API Call

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/bulk HTTP/1.1
Host: {some-ise-node-ip}
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxx Content-Type:
application/vnd.com.cisco.ise.ers.endpointbulkrequest.1.0+xml
{
  <ns3:endpointBulkRequest
  resourceMediaType = "vnd.com.cisco.ise.ers.identity.endpoint.1.0+xml" operationType =
  "create"
  xmlns:ns2 = "ers.ise.cisco.com"
  xmlns:ns3 = "identity.ers.ise.cisco.com">
  <resourcesList> <resource
  xsi:type = "ns3:ersEndPoint"
  description = "created by bulk request"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"> <mac>11:22:33:44:55:66</mac>
  <staticGroupAssignment>false</staticGroupAssignment>
  <staticProfileAssignment>false</staticProfileAssignment>
  </resource>
  . . .
  <resource
  xsi:type = "ns3:ersEndPoint"
  description = "created by bulk request"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"> <group>Profiled</group>
  <groupId>804f5350-7808-11e2-bdd0-0050568e01f0</groupId> <identityStore></identityStore>
  <identityStoreId></identityStoreId> <mac>11:22:33:44:55:77</mac>
  <portalUser></portalUser>
  <profile>Apple-iPod</profile> <profileId>b8128870-7808-11e2-bdd0-0050568e01f0</profileId>
  <staticGroupAssignment>true</staticGroupAssignment>
  <staticProfileAssignment>true</staticProfileAssignment>
  </resource> </resourcesList>
  </ns3:endpointBulkRequest>
}
```

Sample Response for Start Bulk Execution for Endpoints API Call

```
HTTP/1.1 202 ACCEPTED
Date: Thu, 12 Jul 2012 23:59:59 GMT
Location: https://ise-node-ip:9060/ers/config/endpoint/123443545334
```

Get Bulk Status for Endpoints

If a bulk execution request is valid and no other bulk already in progress, the server returns a unique bulk identifier in the LOCATION response header. Use this ID to track the bulk status. The status report will be available for at least 2 hours after the operation's start time.

Table 7-17 Get Bulk Status Main Characteristics

Description	Monitor the specified bulk execution progress
Synopsis	GET /ers/config/endpoint/bulk/{bulkid}
Request Headers	Accept, Authorization, Host
QueryString	n/a
Request Message Body	n/a
Response Headers	Content-Length, Content-Type

Table 7-17 Get Bulk Status Main Characteristics

Response Message Body	BulkStatus
Response Status	200, 400, 401, 403, 404, 415, 500

Get Bulk Status for Endpoints Example

Request

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/bulk/53454354534 HTTP/1.1
Host: {some-ise-node-ip}
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.bulkStatus.1.0+xml
```

Response

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.bulkStatus.1.0+xml Content-Length: 16347
{
  <ns2:bulkStatus
    xmlns:ns2 = "ers.ise.cisco.com"
    successCount = "750"
    startTime = "Thu Mar 07 17:17:35 IST 2013"
    resourcesCount = "750"
    operationType = "create"
    mediaType =
      "vnd.com.cisco.ise.ers.identity.endpoint.1.0+xml"
    failCount = "0"
    executionStatus = "COMPLETED"
    bulkId = "1362669455284">
    <resourcesStatus>
      <resourceStatus status = "SUCCUESS"
        description = "created by bulk request"
        id = "23d068d0-873a-11e2-bad4-00215edbb2a8" />
      . . . . .
      <resourceStatus
        status = "SUCCUESS"
        description = "created by bulk request"
        id = "23cfa580-873a-11e2-bad4-00215edbb2a8"/>
    </resourcesStatus>
  </ns2:bulkStatus>
}
```

External RESTful Services APIs for Endpoint Certificates

The following table lists the External RESTful Services APIs for endpoint certificates:

Table 7-18 ERS APIs Available for Endpoint Certificates

Operation	Method	URL	Content	QueryString
Create Endpoint Certificate	POST	/ers/config/endpointcert/certRequest/	Octet Stream representing a ZIP file	
Get Certificate Resource Version Info	GET	/ers/config/endpointcert/version	n/a	

Create Endpoint Certificate

The following table lists the main characteristics of the Create Endpoint Certificate API call:

Table 7-19 Main Characteristics of Create Endpoint Identity Groups API Call

Description	Create an endpoint certificate
Synopsis	POST /ers/config/endpointcert/certRequest/
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	Endpoint certificate
Response Headers	Content-Length, Content-Type, Location
Response Message Body	Octet Stream representing a ZIP file
Response Status	201, 400, 401, 403, 415, 429, 500

Sample Request for Create Endpoint Certificate API Call

```
PUT https://<ISE-ADMIN-NODE>/ers/config/endpointcert/certRequest
HTTP Content-Type header:
application/vnd.com.cisco.ise.ca.endpointcert.1.0+xml; charset=utf-8

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:endpointcert xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="ca.ers.ise.cisco.com">
  <certTemplateName>Certificate_Template_Name</certTemplateName>
  <certificateRequest>
    <entry>
      <key>san</key>
      <value>11-22-33-44-55-66</value>
    </entry>
    <entry>
      <key>cn</key>
      <value>userName [or] machineName</value>
    </entry>
  </certificateRequest>
  <format>PKCS8 [or] PKCS8_CHAIN [or] PKCS12 [or] PKCS12_CHAIN</format>
  <password>password</password>
</ns3:endpointcert>
```

Sample Response for Create Endpoint Certificate API Call

HTTP Status: 200 (OK)

Content:

[Response is returned as an Octet Stream representing a ZIP file.]

External RESTful Services APIs for Endpoint Identity Groups

The following table lists the External RESTful Services APIs for endpoint identity groups:

Table 7-20 ERS APIs Available for Endpoint Identity Groups

Operation	Method	URL	Content	QueryString
Get All Endpoints Groups	GET	/ers/config/endpointgroup	n/a	page, size, sortasc or sortdsn, filter
Get Endpoint Group	GET	/ers/config/endpointgroup/{id ¹ }	n/a	
Create Endpoint Group	POST	/ers/config/endpointgroup/	Endpointgroup	
Update Endpoint Group	PUT	/ers/config/endpointgroup/{id}	Endpointgroup	
Delete Endpoint Group	DELETE	/ers/config/endpointgroup/{id}	n/a	
Get IdentityGroup Resource Version Info	GET	/ers/config/ endpointgroup /version	n/a	

1. Endpoint Group ID is the UUID type as stored in the Cisco ISE database.

Get All Endpoint Identity Groups

The following table lists the main characteristics of the Get All Endpoint Identity Groups API call:

Table 7-21 Main Characteristics of Get All Endpoint Identity Groups API Call

Description	Retrieve a collection of endpoint groups
Synopsis	GET /ers/config/endpoint
Request Headers	Accept, Authorization, Host
QueryString	page, size, sortasc, sortdsc, filter
Request Message Body	n/a
Response Headers	Content-Length, Content-Type
Response Message Body	List of updated fields
Response Status	202, 400, 401, 403, 404, 415, 429, 500

Sample Request for Get All Endpoint Identity Groups API Call

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/endpointgroup
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.endpointgroup.1.0+xml
```

Sample Response for Get All Endpoint Identity Groups API Call

```

HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.searchresult.1.0+xml
Content-Length: 16347
{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <ns2:searchResult total="2" xmlns:ns2="ers.ise.cisco.com">
  <resources>
    <resource name="name1" id="id1" description="description1">
      <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/endpointgroup/id1" rel="self"/>
    </resource>
    <resource name="name2" id="id2" description="description2">
      <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/endpointgroup/id2" rel="self"/>
    </resource>
  </resources>
</ns2:searchResult>
}

```

Get Endpoint Identity Groups by ID

The following table lists the main characteristics of the Get Endpoint Identity Groups by ID API call:

Table 7-22 Main Characteristics of Read Endpoint Identity Groups API Call

Description	Retrieve the specified endpoint group
Synopsis	GET /ers/config/endpoint/{id}
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	Resource of type Endpoint
Response Status	200, 400, 401,403, 404, 415, 429, 500

Sample Request for Read Endpoint Identity Groups API Call

```

GET https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/333
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.endpointgroup.1.0+xml

```

Sample Response for Read Endpoint Identity Groups API Call

```

HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.identity.endpointgroup.1.0+xml
Content-Length: 16347
{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <ns3:endpointgroup description="description" name="name" id="id"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">

```

```

    <systemDefined>true</systemDefined>
  </ns3:endpointgroup>
}

```

Create Endpoint Identity Groups

The following table lists the main characteristics of the Create Endpoint Identity Groups API call:

Table 7-23 Main Characteristics of Create Endpoint Identity Groups API Call

Description	Create the specified endpoint group
Synopsis	POST /ers/config/endpoint/
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	Endpoint Group
Response Headers	Content-Length, Content-Type, Location
Response Message Body	Endpoint Group
Response Status	201, 400, 401, 403, 415, 429, 500

Sample Request for Create Endpoint Identity Groups API Call

```

POST https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.identity.endpointgroup.1.0+xml
{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <ns3:endpointgroup description="description" name="name" id="id"
  xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
    <systemDefined>true</systemDefined>
  </ns3:endpointgroup>
}

```

Sample Response for Create Endpoint Identity Groups API Call

```

HTTP/1.1 201 OK (see the location header for the new endpoint ID)
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type:
Location: https://cisco.com/ers/config/endpointgroup/444

```

Update Endpoint Identity Groups

The following table lists the main characteristics of the Update Endpoint Identity Groups API call:

Table 7-24 Main Characteristics of Update Endpoint Identity Groups API Call

Description	Update the specified endpoint group
Synopsis	PUT /ers/config/endpoint/{id}

Table 7-24 Main Characteristics of Update Endpoint Identity Groups API Call (continued)

Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	Endpoint Group
Response Headers	Content-Length, Content-Type
Response Message Body	List of updated fields
Response Status	200, 400, 401, 403, 404, 415, 429, 500

Sample Request for Update Endpoint Identity Groups API Call

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/ endpoint /333
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.identity.endpointgroup.1.0+xml
{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <ns2:endpointgroup xmlns:ns2="identity.ers.ise.cisco.com" description="updated" id="0"
  name="Group">
    <systemDefined>false</systemDefined>
  </ns2:endpointgroup>
}
```

Sample Response for Update Endpoint Identity Groups API Call

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.updatedfields.1.0+xml
Content-Length: 529
{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <ns2:updatedFields xmlns:ns2="ers.ise.cisco.com">
    <updatedField field="description">
      <newValue>updated</newValue>
      <oldValue>Group</oldValue>
    </updatedField>
  </ns2:updatedFields>
}
```

Delete Endpoint Identity Groups

The following table lists the main characteristics of the Delete Endpoint Identity API call:

Table 7-25 Main Characteristics of Delete Endpoint Identity Groups API Call

Description	Delete the specified endpoint group
Synopsis	DELETE /ers/config/endpointgroup/{id}
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	N/A

Table 7-25 Main Characteristics of Delete Endpoint Identity Groups API Call (continued)

Response Headers	Content-Length, Content-Type
Response Message Body	NA
Response Status	204, 400, 401, 403, 404, 415, 429, 500

Sample Request for Delete Endpoint Identity Groups API Call

```
DELETE https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/333
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.endpointgroup.1.0+xml
```

Sample Response for Delete Endpoint Identity Groups API Call

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
```

External RESTful Services APIs for Identity Groups

The following table lists the External RESTful Services APIs for Identity Groups:

Table 7-26 APIs Available for Identity Groups

Operation	Method	URL	Content	QueryString
Get All Identity Groups	GET	/ers/config/identitygroup	n/a	page, size, sortacs or sortdsn, filter
Get Identity Group by ID	GET	ers/config/identitygroup/{id}	n/a	
Get IdentityGroup Resource Version Info	GET	/ers/config/identitygroup/version	n/a	

Retrieve All Identity Groups

You can use this API call to retrieve all identity groups in Cisco ISE. The following table lists the main characteristics of this API call:

Table 7-27 Main Characteristics of Retrieve All Identity Groups API Call

Description	Retrieve a collection of identity group resources
Synopsis	GET /ers/config/identitygroup
Request Headers	Accept, Authorization, Host
QueryString	page, size, sortbyacn, sortbydcn, filter
Request Message Body	N/A
Response Headers	Content-Length, Content-Type

Table 7-27 Main Characteristics of Retrieve All Identity Groups API Call (continued)

Response Message Body	SearchResult
Response Status	200, 400, 401, 403, 404, 415, 429, 500

Sample Request for Retrieve All Identity Group API Call

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/identitygroup?page=0&size=20&sortacs=name
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.identitygroup.1.0+xml
```

Sample Response for Retrieve All Identity Group API Call

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.searchresult.1.0+xml
Content-Length: 16347
{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <ns2:searchResult total="2" xmlns:ns2="ers.ise.cisco.com">
    <resources>
      <resource name="name1" id="id1" description="description1">
        <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/identitygroup/id1" rel="self"/>
      </resource>
      <resource name="name2" id="id2" description="description2">
        <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/identitygroup/id2" rel="self"/>
      </resource>
    </resources>
  </ns2:searchResult>
}
```

Get Identity Groups by ID

The following table lists the main characteristics of the Get Identity Groups by ID API call:

Table 7-28 Main Characteristics of Read Endpoint Identity Groups API Call

Description	Retrieve the specified identity group
Synopsis	GET /ers/config/identitygroup/{id}
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	Resource of type Endpoint
Response Status	200, 400, 401, 403, 404, 415, 429, 500

Sample Request for Get Identity Group by ID API Call

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/identitygroup/333
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.identitygroup.1.0+xml
```

Sample Response for Get Identity Group by ID API Call

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:identitygroup name="name" id="id" description="description"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
  <parent>parent</parent>
</ns3:identitygroup>
```

External RESTful Services APIs for Guest Users

The following table lists the External RESTful Services APIs for Guest Users:

Table 7-29 Supported Scenarios

Operation	Method	URL	Content
Get Specific Guest User	GET	/ers/config/guestuser/{id}	n/a
Get All Guest Users	GET	/ers/config/guestuser/	n/a
Create Guest User	POST	/ers/config/guestuser /	guest user information (XML)
Update Guest User	PUT	/ers/config/guestuser/{id}	partial or full guest user information (XML)
Delete Guest User	DELETE	/ers/config/guestuser/{id}	n/a
Suspend Guest User	PUT	/ers/config/guestuser/suspend/{id}	reason
Reinstate Guest User	PUT	/ers/config/guestuser/reinstate/{id}	n/a
Send Email	PUT	/ers/config/guestuser/email/{id}/portalId/{portalId}	senderEmail
Send SMS	PUT	/ers/config/guestuser/sms/{id}/portalId/{portalId}	n/a
Approve Guest User	PUT	/ers/config/guestuser/approve/{id}	n/a
Deny Guest User	PUT	/ers/config/guestuser/deny/{id}	n/a
Reset Password	PUT	/ers/config/guestuser/resetpassword/{id}	n/a
Start Bulk Execution	PUT	/ers/config/ guestuser/bulk	BulkRequest
Get Bulk Status	GET	/ers/config/ guestuser/bulk/{bulkId}	n/a
Change Sponsor's Password	PUT	/ers/config/guestuser/changeSponsorPassword/{portalId}	operationAdditionalData

Table 7-29 Supported Scenarios

Operation	Method	URL	Content
Get All Portals	GET	/ers/config/portal	n/a
Get Portal by ID	GET	/ers/config/portal/{id}	n/a
Get Guest API Info	GET	/ers/config/guestuser/versioninfo	n/a

Content Type and Accept Headers

Each guest account request needs to set the following:

- Content Type as: `application/vnd.com.cisco.ise.identity.guestuser.2.0+xml`
- Accept: `application/vnd.com.cisco.ise.identity.guestuser.2.0+xml`

Each bulk execution request needs to set the following:

- Content Type as: `application/vnd.com.cisco.ise.identity.guestuserbulkrequest.1.0+xml`
- Accept: `application/vnd.com.cisco.ise.identity.guestuserbulkrequest.1.0+xml`

Get a Guest User

You can use the GET operation to retrieve specific guest users from the ISE database using either the guest's username or database record ID.

Table 7-30 Get a Guest User Main Characteristics

Description	Retrieve the specified Guest User
Synopsis	GET /ers/config/guestuser/{id}
Request Headers	Accept, Authorization, Host
QueryString	n/a
Request Message Body	n/a
Response Headers	Content-Length, Content-Type
Response Message Body	Resource of type GuestUser
Response Status	200, 400, 401, 403, 404, 415, 500

Get a Guest User Examples

- [Get a Guest User by ID Example, page 7-26](#)
- [Filter by Usernames that Start with “ilucky” Example, page 7-26](#)
- [Filter by Username that Starts with “ilucky” and Last Name that Starts with “J” Example, page 7-27](#)
- [Filter By the First Name “John” and Sort By Username Example, page 7-28](#)
- [Guest User Request and Response Using curl Example, page 7-29](#)

Get a Guest User by ID Example

Request

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/3333
```

```
Content-Type - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Accept - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Authorization - Basic xxxxxxxxxxxxxxxxxxxxxxx
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<ns3:guestuser xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com"
name="fzvhervocnz" id="af827350-1f0f-11e4-b961-005056103001">
  <link type="application/xml" href="https://10.0.10.130:9060/ers/config/guestuser/3333"
rel="self"/>
  <customFields/>
  <guestAccessInfo>
    <fromDate>08/08/2014 08:21</fromDate>
    <toDate>08/09/2014 08:21</toDate>
    <validDays>1</validDays>
  </guestAccessInfo>
  <guestInfo>
    <company>Cisco</company>
    <creationTime>08/08/2014 08:21</creationTime>
    <emailAddress>doe@example.com</emailAddress>
    <enabled>true</enabled>
    <firstName>John</firstName>
    <lastName>Doe</lastName>
    <notificationLanguage>English</notificationLanguage>
    <password>12345</password>
    <phoneNumber>9999998877</phoneNumber>
    <smsServiceProvider>ATT</smsServiceProvider>
    <userName>Guest1</userName>
  </guestInfo>
  <guestType>Daily (default)</guestType>
  <personBeingVisited>sponsor@example.com</personBeingVisited>
  <reasonForVisit>Interview</reasonForVisit>
  <sponsorUserName>SponsoredUser1</sponsorUserName>
  <status>Awaiting Initial Login</status>
</ns3:guestuser>
```

Related Topics

See [Guest Passwords, page 6-6](#) for details on password visibility in the API.

Filter by Usernames that Start with “ilucky” Example

Request

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/?filter=username.STARTSW.ilucky
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult
  xmlns:ns2="ers.ise.cisco.com" total="8">
  <resources>
    <resource name="ilucky101" id="a0957160-6224-11e3-9bc2-000c2932c73c">
```

```

        <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/a0957160-6224-11e3-9bc2-000c2932c73
c" rel="self"/>
    </resource>
    <resource name="ilucky102" id="e14f4460-6224-11e3-9bc2-000c2932c73c">
        <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/e14f4460-6224-11e3-9bc2-000c2932c73
c" rel="self"/>
    </resource>
    <resource name="ilucky201" id="123581f0-6227-11e3-9bc2-000c2932c73c">
        <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/123581f0-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
    </resource>
    <resource name="ilucky301" id="154f9330-6227-11e3-9bc2-000c2932c73c">
        <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/154f9330-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
    </resource>
    <resource name="ilucky401" id="172c6980-6227-11e3-9bc2-000c2932c73c">
        <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/172c6980-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
    </resource>
    <resource name="ilucky501" id="19631fa0-6227-11e3-9bc2-000c2932c73c">
        <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/19631fa0-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
    </resource>
    <resource name="ilucky601" id="1b44b0e0-6227-11e3-9bc2-000c2932c73c">
        <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/1b44b0e0-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
    </resource>
    <resource name="ilucky602" id="2e1ac600-6227-11e3-9bc2-000c2932c73c">
        <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/2e1ac600-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
    </resource>
</resources>
</ns2:searchResult>

```

Filter by Username that Starts with "ilucky" and Last Name that Starts with "J" Example

Request

```

GET
https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/?filter=userName.STARTSW.ilucky&filter=
lastName.STARTSW.j

```

Response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult
  xmlns:ns2="ers.ise.cisco.com" total="8">
  <resources>
    <resource name="ilucky101" id="a0957160-6224-11e3-9bc2-000c2932c73c">
      <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/a0957160-6224-11e3-9bc2-000c2932c73
c" rel="self"/>
    </resource>
    <resource name="ilucky102" id="e14f4460-6224-11e3-9bc2-000c2932c73c">

```

```

        <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/e14f4460-6224-11e3-9bc2-000c2932c73
c" rel="self"/>
    </resource>
    <resource name="ilucky201" id="123581f0-6227-11e3-9bc2-000c2932c73c">
        <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/123581f0-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
    </resource>
    <resource name="ilucky301" id="154f9330-6227-11e3-9bc2-000c2932c73c">
        <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/154f9330-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
    </resource>
    <resource name="ilucky401" id="172c6980-6227-11e3-9bc2-000c2932c73c">
        <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/172c6980-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
    </resource>
    <resource name="ilucky501" id="19631fa0-6227-11e3-9bc2-000c2932c73c">
        <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/19631fa0-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
    </resource>
    <resource name="ilucky601" id="1b44b0e0-6227-11e3-9bc2-000c2932c73c">
        <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/1b44b0e0-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
    </resource>
    <resource name="ilucky602" id="2e1ac600-6227-11e3-9bc2-000c2932c73c">
        <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/2e1ac600-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
    </resource>
</resources>
</ns2:searchResult>

```

Filter By the First Name “John” and Sort By Username Example

Request

```

GET
https://<ISE-Admin-node>:9060/ers/config/guestuser/?page=0&size=10&sortdsc=name&filter=fir
stName.eq.john

```

Response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult
  xmlns:ns2="ers.ise.cisco.com" total="2">
  <resources>
    <resource name="jdoe0002" id="886f5b40-5ece-11e3-8faf-000c29c56fc6">
      <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/886f5b40-5ece-11e3-8faf-000c29c56fc
6" rel="self"/>
    </resource>
    <resource name="jdoe0001" id="79e5a5a0-5df9-11e3-84f5-000c29c56fc6">
      <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/79e5a5a0-5df9-11e3-84f5-000c29c56fc
6" rel="self"/>
    </resource>
  </resources>
</ns2:searchResult>

```

Guest User Request and Response Using curl Example

The following example describes a request for getting a guest user by ID sent to ISE and its response using `curl` Linux command.

curl Command

```
$ curl -v -k -H 'ACCEPT:application/vnd.com.cisco.ise.identity.guestuser.2.0+xml'
https://username:password@<ISE-ADMIN-NODE>:9060/ers/config/guestuser/user1
* About to connect() to <ISE-ADMIN-NODE> port 9060
*   Trying 111.11.11.111... * connected
* Connected to <ISE-ADMIN-NODE> (<ISE-ADMIN-NODE-IP>) port 9060
* successfully set certificate verify locations:
*   CAfile: /usr/share/ssl/certs/ca-bundle.crt
*   CAspace: /usr/share/ssl/certs/ca-bundle.crt
*   CApath: none
* SSL connection using DHE-RSA-AES256-SHA
* Server certificate:
*   subject: /CN=<ISE-ADMIN-NODE>
*   start date: 2013-11-26 00:56:55 GMT
*   expire date: 2014-11-26 00:56:55 GMT
*   common name: <ISE-ADMIN-NODE>
*   issuer: /CN=<ISE-ADMIN-NODE>
* Server auth using Basic with user 'username'
```

GET Guest User by ID Request

```
> GET /ers/config/guestuser/444
Authorization: Basic xxxxxxxxxxxxxxxxxxxx
User-Agent: curl/7.12.1 (i386-redhat-linux-gnu) libcurl/7.12.1 OpenSSL/0.9.7a zlib/1.2.1.2
libidn/0.5.6
Host: <ISE-ADMIN-NODE>:9060
Pragma: no-cache
ACCEPT:application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

Response

```
< HTTP/1.1 200 OK
< Pragma: No-cache
< Cache-Control: no-cache
< Expires: Wed, 31 Dec 1969 16:00:00 PST
< Set-Cookie: JSESSIONIDSSO=0FCBC2621A0897193FE3105B3FBA8F16; Path=/; Secure
< Set-Cookie: JSESSIONID=5B6092B3FCCE047F7282C52592FAFC7A; Path=/ers; Secure
< Date: Thu, 02 Jan 2014 23:01:59 GMT
< Content-Type: application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
< Content-Length: 1162
< Server:
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:guestuser xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com"
name="user1" id="b4bdf2b0-73e1-11e3-8cdf-000c29c56fc6">
<link type="application/xml" href="https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/444"
rel="self"/>
<guestAccessInfo>
<fromDate>08/06/2014 23:26</fromDate>
<toDate>08/07/2014 23:26</toDate>
<validDays>1</validDays>
</guestAccessInfo>
<guestInfo>
<company>New Company</company>
<emailAddress>john@example.com</emailAddress>
<firstName>John</firstName>
<lastName>Doe</lastName>
<notificationLanguage>English</notificationLanguage>
<phoneNumber>9999998877</phoneNumber>
<smsServiceProvider>Global Default</smsServiceProvider>
```

```

<userName>user1</userName>
</guestInfo>
<guestType>Daily (default)</guestType>
<personBeingVisited>sponsor@example.com</personBeingVisited>
<portalId>ff2d99e0-2101-11e4-b5cf-005056bf2f0a</portalId>
<reasonForVisit>Interview</reasonForVisit>
</ns3:guestuser>

```

Related Topics

See [Guest Passwords, page 6-6](#) for details on password visibility in the API.

Get All Guest Users

You can use the GET operation to retrieve all guest users in the ISE database and filter the results based on criteria such as name, username, or email address. The response includes the guest's username, ID, and a link to its full representation.

Table 7-31 Get All Guest Users Main Characteristics

Description	Retrieve a collection of Guest Users
Synopsis	GET /ers/config/guestuser/
Request Headers	Accept, Authorization, Host
QueryString	page, size, sortasc, sortdsc, filter
Request Message Body	n/a
Response Headers	Content-Length, Content-Type
Response Message Body	SearchResult
Response Status	200, 400, 401, 403, 404, 415, 500

Related Topics

[Filtering Parameters, page 6-9](#)

Get All Example

In the following example, the GET operation retrieves all guest users with a username that starts with `ilu` and a first name that starts with `b`.

Request

```

GET
https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/?page=0&size=10&sortasc=name&filter=name.STARTSW.ilu&filter=firstName.STARTSW.b

```

```

Content-Type - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Accept - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Authorization - Basic xxxxxxxxxxxxxxxxxxxxxxx

```

Response

```

HTTP/1.1 200 OK;
Date:Sat, 15 Dec 2012 21:55:05 GMT;
Content-Length:1439;

```

```

Content-Type:application/vnd.com.cisco.ise.ers.searchresult.1.0+xml;

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult xmlns:ns2="ers.ise.cisco.com" total="6">
  <resources>
    <ns2:resource name="ilucky01" id="61dc9060-46a1-11e2-b141-000c290fcf9a">
      <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/61dc9060-46a1-11e2-b141-000c290fcf9a" rel="self"/>
    </ns2:resource>
    <ns2:resource name="ilucky02" id="3f43bb40-468e-11e2-8f92-000c290fcf9a">
      <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/3f43bb40-468e-11e2-8f92-000c290fcf9a" rel="self"/>
    </ns2:resource>
    <ns2:resource name="ilucky03" id="6c65d6d0-468e-11e2-8f92-000c290fcf9a">
      <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/6c65d6d0-468e-11e2-8f92-000c290fcf9a" rel="self"/>
    </ns2:resource>
    <ns2:resource name="ilucky04" id="6948bdb0-46a1-11e2-b141-000c290fcf9a">
      <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/6948bdb0-46a1-11e2-b141-000c290fcf9a" rel="self"/>
    </ns2:resource>
    <ns2:resource name="ilucky05" id="abbb6440-46a1-11e2-b141-000c290fcf9a">
      <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/abbb6440-46a1-11e2-b141-000c290fcf9a" rel="self"/>
    </ns2:resource>
    <ns2:resource name="ilucky06" id="4d9a1530-46fd-11e2-b70b-000c290fcf9a">
      <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/4d9a1530-46fd-11e2-b70b-000c290fcf9a" rel="self"/>
    </ns2:resource>
  </resources>
</ns2:searchResult>

```

Create a Guest User

You can use the POST operation to create a new guest user account, which allows a guest to log in through the guest flow.

The guestType is required to create a guest user account.

Table 7-32 Create a Guest User Main Characteristics

Description	Create the specified Internal User
Synopsis	POST /ers/config/guestuser/
Request Headers	Accept, Authorization, Host
QueryString	n/a
Request Message Body	GuestUser
Response Headers	Content-Length, Content-Type, Location
Response Message Body	Resource of type GuestUser
Response Status	201, 400, 401, 403, 415, 500

Guest User XML Structure

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:guestuser name="guestUser" id="123456789" description="ERS Example user "
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
  <customFields>
    <entry>
      <key>some key</key>
      <value>some value</value>
    </entry>
    <entry>
      <key>another key</key>
      <value>and its value</value>
    </entry>
  </customFields>
  <guestInfo>
    <emailAddress>email@some.uri.com</emailAddress>
    <enabled>true</enabled>
    <password>asdlkj324ew</password>
    <phoneNumber>3211239034</phoneNumber>
    <smsServiceProvider>GLobal Default</smsServiceProvider>
    <userName>DS3ewdsa34wWE</userName>
  </guestInfo>
  <guestType>Contractor</guestType>
  <portalId>23423432523</portalId>
  <sponsorUserName>Mr Spons</sponsorUserName>
</ns3:guestuser>
```

Create a Guest User Example

Request

```
POST https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/
```

```
Content-Type - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

```
Accept - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

```
Authorization - Basic xxxxxxxxxxxxxxxxxxxxxxx
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:guestuser xmlns:ns2="identity.ers.ise.cisco.com">
  <guestAccessInfo>
    <fromDate>08/08/2014 08:15</fromDate>
    <toDate>08/09/2014 08:15</toDate>
    <validDays>1</validDays>
  </guestAccessInfo>
  <guestInfo>
    <company>New Company</company>
    <emailAddress>doe@example.com</emailAddress>
    <firstName>John</firstName>
    <lastName>Doe</lastName>
    <notificationLanguage>English</notificationLanguage>
    <phoneNumber>9999998877</phoneNumber>
    <smsServiceProvider>Global Default</smsServiceProvider>
    <userName>guestuser1</userName>
  </guestInfo>
  <guestType>Daily (default)</guestType>
  <personBeingVisited>sponsor@example.com</personBeingVisited>
  <portalId>ff2d99e0-2101-11e4-b5cf-005056bf2f0a</portalId>
  <reasonForVisit>Interview</reasonForVisit>
</ns2:guestuser>
```


Response

```
HTTP/1.1 201 Created;
Date:Sat, 15 Dec 2012 21:20:51 GMT;
Content-Length:0;
Location:https://<ISE-ADMIN-NODE>/ers/config/guestuser/e1bb8290-6ccb-11e3-8cdf-000c29c56fc6;
Set-Cookie:JSESSIONID=28CF43F1ACCC7448BED7255DC7B787EE; Path=/ers;
Secure;JSESSIONIDSSO=DB6D6900088D1863CA84863570392E4C; Path=/; Secure;
Content-Type:application/xml;
```

Related Topics

See [Guest Passwords, page 6-6](#) for details on password visibility in the API.

Update a Guest User

Updating a resource using the PUT operation gives you the ability to change the attributes of an existing guest user. A full or partial update can be done of the guest user's attributes.

Table 7-33 *Update a Guest User Main Characteristics*

Description	Update the specified Guest User
Synopsis	PUT /ers/config/guestuser/{id}
Request Headers	Accept, Authorization, Host
QueryString	n/a
Request Message Body	GuestUser
Response Headers	Content-Length, Content-Type
Response Message Body	List of updated fields
Response Status	200, 400, 401, 403, 404, 415, 500

Update User Example

Request

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/name/ilucky101
```

```
Content-Type - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

```
Accept - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

```
Authorization - Basic xxxxxxxxxxxxxxxxxxxxxxx
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ns2:guestuser xmlns:ns2="identity.ers.ise.cisco.com">
  <portalId>ff2d99e0-2101-11e4-b5cf-005056bf2f0a</portalId>
  <reasonForVisit>Interview</reasonForVisit>
</ns2:guestuser>
```

Response

```
Status:200 OK
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:updatedFields xmlns:ns2="ers.ise.cisco.com">
  <updatedField field="ReasonForVisit">
    <newValue>Interview</newValue>
    <oldValue>no reason</oldValue>
```

```

</updatedField>
<updatedField field="validDays">
  <newValue>0</newValue>
  <oldValue>1</oldValue>
</updatedField>
</ns2:updatedFields>

```

Delete a Guest User

You can delete a guest user's record from the ISE database using the database record ID. The user will not be able to log in during their next attempt.

Table 7-34 Delete a Guest User Main Characteristics

Description	Delete the specified Guest User
Synopsis	DELETE /ers/config/guestuser/{id}
Request Headers	Accept, Authorization, Host
QueryString	n/a
Request Message Body	n/a
Response Headers	Content-Length, Content-Type
Response Message Body	n/a
Response Status	204, 400, 401, 403, 404, 415, 500

Delete a Guest User Example

Request

```
DELETE https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/3333
```

```
Content-Type - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

```
Accept - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

```
Authorization - Basic xxxxxxxxxxxxxxxxxxxxxxxxx
```

Response

```
HTTP/1.1 200 OK
```

```
Date: Thu, 12 Jul 2012 23:59:59 GMT
```

Suspend a Guest User

Use the PUT operation to suspend a specific guest user. The user will not be able to log in during their next attempt. You must include a reason for the suspension. The reason can include spaces.

Table 7-35 Suspend a Guest User Main Characteristics

Description	Suspend the specified Guest User
Synopsis	PUT /ers/config/guestuser/suspend/{id}
Request Headers	Accept, Authorization, Host
QueryString	n/a

Table 7-35 Suspend a Guest User Main Characteristics

Request Message Body	reason
Response Headers	Content-Length, Content-Type
Response Message Body	Resource of type GuestUser
Response Status	204, 400, 401, 403, 404, 415, 500

Suspend a Guest User by ID Example

Request

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/suspend/3333
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Content-Type - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ns3:operationAdditionalData xmlns:ns2="identity.ers.ise.cisco.com"
xmlns:ns3="ers.ise.cisco.com">
  <requestAdditionalAttributes>
    <additionalAttribute name="reason" value="AUP not accepted"/>
  </requestAdditionalAttributes>
</ns3:operationAdditionalData>
```

Response

```
HTTP/1.1 204 No Content
Sat, 15 Dec 2012 10:14:38 GMT
```

Reinstate a Guest User

Use the PUT operation to reinstate a suspended guest's user account.

Table 7-36 Reinstate a Guest User Main Characteristics

Description	Reinstate the specified Guest User
Synopsis	PUT /ers/config/guestuser/reinstate/{id}
Request Headers	Accept, Authorization, Host
QueryString	n/a
Request Message Body	n/a
Response Headers	Content-Length, Content-Type
Response Message Body	Resource of type GuestUser
Response Status	204, 400, 401, 403, 404, 415, 500

Reinstate Guest User Example

Request

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/reinstate/33
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

Response

HTTP/1.1 204 OK
Date: Sat, 15 Dec 2012 10:20:48 GMT

Send an Email to a Guest User

Use the PUT operation to send an email to a guest user's email account. This requires an SMTP server to be configured in Cisco ISE.

The request requires a portal ID because the portal configuration contains information needed for the email body and subject.

Table 7-37 Send an Email to a Guest User Main Characteristics

Description	Send an email to the specified Guest User
Synopsis	PUT /ers/config/guestuser/email/{id}/portalId/{portalID}
Request Headers	Accept, Authorization, Host
QueryString	n/a
Request Message Body	senderEmail
Response Headers	Content-Length, Content-Type
Response Message Body	n/a
Response Status	204, 400, 401, 403, 404, 415, 500

Send an Email to a Guest User Example

Request

```
PUT
https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/email/4444/portalId/ff2d99e0-2101-11e4-
b5cf-005056bf2f0a
  Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
  Accept:
  application/vnd.com.cisco.ise.identity.guestuser.2.0+xml

<?xml version="1.0" encoding="UTF-8"?>
  <ns3:operationAdditionalData xmlns:ns2="identity.ers.ise.cisco.com"
xmlns:ns3="ers.ise.cisco.com">
    <requestAdditionalAttributes>
      <additionalAttribute name="senderEmail" value="sender Email"/>
    </requestAdditionalAttributes>
  </ns3:operationAdditionalData>
```

Response

HTTP/1.1 204 OK
Date: Sat, 15 Dec 2012 10:20:48 GMT

Send an SMS Text to a Guest User

Use the PUT operation to send a text message to a guest user's mobile phone. This requires an SMTP server to be configured in Cisco ISE.

The request requires a portal ID because the portal configuration contains information needed for the text body.

Table 7-38 *Send an Email to a Guest User Main Characteristics*

Description	Send an sms to the specified Guest User
Synopsis	PUT /ers/config/guestuser/sms/{id}/portalId/{portalID}
Request Headers	Accept, Authorization, Host
QueryString	n/a
Request Message Body	n/a
Response Headers	Content-Length, Content-Type
Response Message Body	n/a
Response Status	204, 400, 401, 403, 404, 415, 500

Send an SMS Example

Request

```
PUT
https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/sms/444/portalId/ff2d99e0-2101-11e4-b5cf-005056bf2f0a
  Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
  Accept:
  application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

Response

```
HTTP/1.1 204 OK
Date: Sat, 15 Dec 2012 10:20:48 GMT
```

Approve a Guest User

This operation allows you to approve a guest user account. This requires using the guest account ID.

Table 7-39 *Get API Version Main Characteristics*

Description	Approve the specified Guest User
Synopsis	PUT /ers/config/guestuser/approve/{id}
Request Headers	Accept, Authorization, Host
QueryString	n/a
Request Message Body	n/a
Response Headers	Content-Length, Content-Type
Response Message Body	API version
Response Status	200, 400, 401, 403, 404, 415, 500

Approve a Guest User Example

Request

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/approve/3333
  Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
  Accept: application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

Response

```
HTTP/1.1 204 OK
Date: Sat, 15 Dec 2012 10:20:48 GMT
```

Deny Approval for a Guest User Account

This operation allows you to deny approval for a guest user account. This requires using the guest account ID.

Table 7-40 Get API Version Main Characteristics

Description	Deny the specified Guest User
Synopsis	PUT /ers/config/guestuser/deny/{id}
Request Headers	Accept, Authorization, Host
QueryString	n/a
Request Message Body	n/a
Response Headers	Content-Length, Content-Type
Response Message Body	API version
Response Status	200, 400, 401, 403, 404, 415, 500

Deny a Guest User's Approval Example

Request

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/deny/7777
  Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
  Accept: application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

Response

```
HTTP/1.1 204 OK
Date: Sat, 15 Dec 2012 10:20:48 GMT
```

Reset Password for a Guest User Account

This operation allows you to reset the password for a guest user account. This requires using the guest account ID. This operation returns a new, generated password. You cannot specify your own password using the REST API.

Table 7-41 Get API Version Main Characteristics

Description	Reset password for the specified Guest User
Synopsis	PUT /ers/config/guestuser/resetpassword/{id}
Request Headers	Accept, Authorization, Host
QueryString	n/a
Request Message Body	n/a
Response Headers	Content-Length, Content-Type
Response Message Body	New password
Response Status	200, 400, 401, 403, 404, 415, 500

Reset a Guest User's Password Example

Request

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/resetpassword/7777
  Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
  Accept: application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

Response

```
HTTP/1.1 204 OK
Date: Sat, 15 Dec 2014 10:20:48 GMT
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:operationResult xmlns:ns2="ers.ise.cisco.com">
  <attributesList>
    <attribute value="DdsAASDs%$##@ssds12" name="password"/>
  </attributesList>
</ns2:operationResult>
```

Start Bulk Execution for Guest Users

A bulk request will allow you to send up to 500 operations in a single request, or up to 5000 operations based on ID.

If the request is valid, the server returns the status code 202 (ACCEPTED) and a unique bulk identifier in the LOCATION response header that you can use to track the bulk status using the Get Bulk Status operation.

Only one bulk is allowed to run at a time. If a bulk request was posted while another bulk is still running, the server will return with a response status 503 (Service Unavailable) with a corresponding descriptive message asking the client to try again later.

Table 7-42 Start Bulk Execution Main Characteristics

Description	Start Execute
Synopsis	PUT /ers/config/guestuser/bulk
Request Headers	Accept, Authorization, Host
QueryString	n/a
Request Message Body	BulkRequest

Table 7-42 Start Bulk Execution Main Characteristics

Response Headers	Content-Length, Content-Type
Response Message Body	n/a
Response Status	202, 400, 401, 403, 404, 415, 500

Create Guest Bulk Example

Request

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/bulk
Authorization: Basic
Content-Type: application/vnd.com.cisco.ise.identity.guestuserbulkrequest.1.0+xml
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:bulkRequest xsi:type="ns2:guestUserBulkRequest"
  resourceMediaType="vnd.com.cisco.ise.identity.guestuser.1.0+xml"
  operationType="create"
  xmlns:ns2="identity.ers.ise.cisco.com"
  xmlns:ns3="ers.ise.cisco.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <resourcesList>
    <resource xsi:type="ns2:GuestUser" description="created by bulk">
      <portalId>6ab68890-d0f1-11e3-a1d5-005056bf4687</portalId>
      <guestAccessInfo>
        <groupTag>group</groupTag>
        <validDays>2</validDays>
        <location>London</location>
        <ssid>guest_ssid</ssid>
      </guestAccessInfo>
      <guestInfo>
        <company>new company</company>
        <emailAddress>joe@example.com</emailAddress>
        <enabled>true</enabled>
        <firstName>John</firstName>
        <lastName>Doe</lastName>
        <phoneNumber>6033203311</phoneNumber>
        <userName>lucky7</userName>
        <password>1234</password>
        <notificationLanguage>English</notificationLanguage>
        <smsServiceProvider>ATT</smsServiceProvider>
      </guestInfo>
      <guestType>DAILY</guestType>
      <reasonForVisit>interview</reasonForVisit>
      <personBeingVisited>sponsor@cisco.com</personBeingVisited>
    </resource>
    ...
    <resource xsi:type="ns2:GuestUser" description="created by bulk">
      <portalId>6ab68890-d0f1-11e3-a1d5-005056bf4687</portalId>
      <guestAccessInfo>
        <groupTag>group</groupTag>
        <validDays>3</validDays>
        <location>London</location>
        <ssid>guest_ssid</ssid>
      </guestAccessInfo>
      <guestInfo>
        <company>new company</company>
        <emailAddress>mary@example.com</emailAddress>
        <enabled>true</enabled>
        <firstName>Mary</firstName>
```



```

    <lastName>Sue</lastName>
    <phoneNumber>6039990000</phoneNumber>
    <userName>lucky13</userName>
    <password>1234</password>
    <notificationLanguage>English</notificationLanguage>
    <smsServiceProvider>ATT</smsServiceProvider>
  </guestInfo>
  <guestType>DAILY</guestType>
  <reasonForVisit>interview</reasonForVisit>
  <personBeingVisited>sponsor@cisco.com</personBeingVisited>
</resource>
</resourcesList>
</ns3:bulkRequest>

```

Response

```

HTTP/1.1 202 ACCEPTED
Date: Thu, 12 Jul 2012 23:59:59 GMT
Location: https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/123443545334

```

Related Topics

[Get Bulk Status for Endpoints, page 7-15](#)

Get Bulk Status for Guest Users

If a bulk execution request is valid and no other bulk already in progress, the server returns a unique bulk identifier in the LOCATION response header. Use this ID to track the bulk status. The status report will be available for at least 2 hours after the operation's start time.

Table 7-43 Get Bulk Status Main Characteristics

Description	Monitor the specified bulk execution progress
Synopsis	GET /ers/config/guestuser/bulk/{bulkid}
Request Headers	Accept, Authorization, Host
QueryString	n/a
Request Message Body	n/a
Response Headers	Content-Length, Content-Type
Response Message Body	BulkStatus
Response Status	200, 400, 401, 403, 404, 415, 500

Get Bulk Status for Guest Users Example

Request

```

GET https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/bulk/53454354534 HTTP/1.1
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.guestuserbulkrequest.1.0+xml

```

Response

```

HTTP/1.1 200 OK
Date: Thu Mar 07 18:17:35 IST 2013 GMT
Content-Type: application/vnd.com.cisco.ise.ers.guestuserbulkrequest.1.0+xml
Content-Length: 16347

```

```

{
  <ns2:bulkStatus
    xmlns:ns2 = "ers.ise.cisco.com"
    successCount = "50"
    startTime = "Thu Mar 07 17:17:35 IST 2013"
    resourcesCount = "50"
    operationType = "create"
    resourceMediaType = "vnd.com.cisco.ise.ers.identity.guestuser.1.0+xml"
    failCount = "0"
    executionStatus = "COMPLETED"
    bulkId = "53454354534">

    <resourcesStatus>
      <resourceStatus
        status = "SUCCUESS"
        description = "created by bulk request"
        id = "23d068d0-873a-11e2-bad4-00215edbb2a8"/>

      ...

      <resourceStatus
        status = "SUCCUESS"
        description = "created by bulk request"
        id = "23cfa580-873a-11e2-bad4-00215edbb2a8"/>
    </resourcesStatus>
  </ns2:bulkStatus>
}

```

Change a Sponsor's Password

This operation allows you to change the password of the sponsor who is currently logged in. This requires using the portal ID.

Table 7-44 Get API Version Main Characteristics

Description	Update the logged-in sponsor's password
Synopsis	PUT /ers/config/guestuser/changeSponsorPassword/ {portalId}
Request Headers	Accept, Authorization, Host
QueryString	n/a
Request Message Body	n/a
Response Headers	Content-Length, Content-Type
Response Message Body	API version
Response Status	200, 400, 401, 403, 404, 415, 500

Change a Sponsor's Password Example

Request

```

PUT https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/changeSponsorPassword/88888
Host: cisco.com
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.identity.guestuser.2.0+xml

```

```
<?xml version="1.0" encoding="UTF-8"?>
<ns3:operationAdditionalData xmlns:ns2="identity.ers.ise.cisco.com"
xmlns:ns3="ers.ise.cisco.com">
  <requestAdditionalAttributes>
    <additionalAttribute name="newPassword" value="Cisco1234"/>
    <additionalAttribute name="currentPassword" value="Autom8me"/>
  </requestAdditionalAttributes>
</ns3:operationAdditionalData>
```

Response

```
HTTP/1.1 204 OK
Date: Sat, 15 Dec 2012 10:20:48 GMT
```

External RESTful Services APIs for Portals

The following table lists the External RESTful Services APIs for Portals:

Table 7-45 APIs Available for Portals

Operation	Method	URL	Content	QueryString
Get All Portals	GET	/ers/config/portal	n/a	Page, Size, sortacs or sortdsn, Filter
Get Portal by ID	GET	/ers/config/portal/{id}	n/a	

Get All Portals

The following table lists the main characteristics of the Get All Portals API call:

Table 7-46 Main Characteristics of Get All Portals API Call

Description	Retrieve a collection of portals
Synopsis	GET /ers/config/portal
Request Headers	Accept, Authorization, Host
QueryString	page, size, sortbyacn, sortbydcn, filter
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	SearchResult
Response Status	200, 400, 401, 403, 404, 415,500

Sample Request for Get All Portals Call

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/portal
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.portal.1.0+xml
```

Sample Response for Get All Portals Call

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.searchresult.1.0+xml
Content-Length: 16347
{
<?xml version="1.0" encoding="utf-8" standalone="yes"?> <ns2:searchResult total="2"
xmlns:ns2="ers.ise.cisco.com">
  <resources>
    <resource name="portal1" id="id1">
      <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/portal/id1" rel="self"/>
    </resource>
    <resource name="portal2" id="id2">
      <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/portal/id2" rel="self"/>
    </resource>
  </resources>
</ns2:searchResult>
```

Get Portal by ID

The following table lists the main characteristics of the Get Portal by ID API call:

Table 7-47 Main Characteristics of Get Portal by ID API Call

Description	Retrieve the specified portal
Synopsis	GET /ers/config/portal/{id}
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	Resource of type Portal
Response Status	200, 400, 401,403, 404, 415, 500

Sample Request for Get Portal by ID Call

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/portal/333
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.portal.1.0+xml
```

Sample Response for Get Portal by ID Call

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.identity.portal.1.0+xml Content-Length: 16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:portal name="sponsor" id="d7b703f0-b073-11e3-bd6c- 005056a15fa7"
xmlns:ns2="ers.ise.cisco.com"
xmlns:ns3="identity.ers.ise.cisco.com">
```

```

    <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/portal/333" rel="self" />
    <allowSponsorToChangeOwnPassword>false</allowSponsorToChangeOwnPassword>
    <GuestUserFieldList>
      <GuestUserField>
        <customType>false</customType>
        <dataType>DROPDOWN</dataType>
        <dictionaryLabelKey>ui_sms_provider_label</dictionaryLabelKey>
        <labelName>SMS Service Provider</labelName>
        <required>true</required>
      </GuestUserField>
      <GuestUserField>
        <customType>false</customType>
        <dataType>TEXT</dataType>
        <dictionaryLabelKey>ui_company_label</dictionaryLabelKey>
        <labelName>Company</labelName>
        <required>true</required>
      </GuestUserField>
      <GuestUserField>
        <customType>false</customType>
        <dataType>TEXT</dataType>
        <dictionaryLabelKey>ui_first_name_label</dictionaryLabelKey>
        <labelName>First name</labelName>
        <required>true</required>
      </GuestUserField>
      <GuestUserField>
        <customType>false</customType>
        <dataType>TEXT</dataType>
        <dictionaryLabelKey>ui_reason_visit_label</dictionaryLabelKey>
        <labelName>Reason for visit</labelName>
        <required>true</required>
      </GuestUserField>
      <GuestUserField>
        <customType>true</customType>
        <dataType>TEXT</dataType>
        <dictionaryLabelKey>ui_ssn-number_text_label</dictionaryLabelKey>
        <instructionText>social </instructionText>
        <labelName>ssn-number</labelName>
        <required>false</required>
      </GuestUserField>
      <GuestUserField>
        <customType>false</customType>
        <dataType>PHONE</dataType>
        <dictionaryLabelKey>ui_phone_number_label</dictionaryLabelKey>
        <labelName>Phone number</labelName>
        <required>true</required>
      </GuestUserField>
      <GuestUserField>
        <customType>false</customType>
        <dataType>EMAIL</dataType>
        <dictionaryLabelKey>ui_person_visited_label</dictionaryLabelKey>
        <labelName>Person being visited</labelName>
        <required>true</required>
      </GuestUserField>
      <GuestUserField>
        <customType>false</customType>
        <dataType>EMAIL</dataType>
        <dictionaryLabelKey>ui_email_address_label</dictionaryLabelKey>
        <labelName>Email address</labelName>
        <required>true</required>
      </GuestUserField>
      <GuestUserField>
        <customType>false</customType>
        <dataType>TEXT</dataType>

```

```

    <dictionaryLabelKey>ui_last_name_label</dictionaryLabelKey>
      <labelName>Last name</labelName>
      <required>true</required>
    </GuestUserField>
  </GuestUserFieldList>
</ns3:portal>
}

```

External RESTful Services APIs for Profiles

The following table lists the External RESTful Services APIs for Profiles:

Table 7-48 APIs Available for Portals

Operation	Method	URL	Content	QueryString
Get All Profiles	GET	/ers/config/profilerprofile	n/a	Page, Size, sortacs or sortdsn, Filter
Get Profile by ID	GET	/ers/config/profilerprofile/{id}	n/a	

Get All Profiles

The following table lists the main characteristics of the Get All Portals API call:

Table 7-49 Main Characteristics of Get All Portals API Call

Description	Retrieve a collection of profiles
Synopsis	GET /ers/config/profilerprofile
Request Headers	Accept, Authorization, Host
QueryString	page, size, sortbyacn, sortbydcn, filter
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	SearchResult
Response Status	200, 400, 401, 403, 404, 415,500

Sample Request for Get All Profiles Call

```

GET https://<ISE-ADMIN-NODE>:9060/ers/config/profilerprofile
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.profilerprofile.1.0+xml

```

Sample Response for Get All Profiles Call

```

HTTP/1.1 200 OK
Date: Thu, 12 Jul 2014 23:59:59 GMT

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

```

```
<ns2:searchResult total="2" xmlns:ns2="ers.ise.cisco.com">
  <nextPage type="application/xml" href="link-to-next-page" rel="next"/>
  <previousPage type="application/xml" href="link-to-previous-page" rel="previous"/>
  <resources>
    <resource name="name1" id="id1" description="description1"/>
    <resource name="name2" id="id2" description="description2"/>
  </resources>
</ns2:searchResult>
```

Get Portal by ID

The following table lists the main characteristics of the Get Profile by ID API call:

Table 7-50 Main Characteristics of Get Portal by ID API Call

Description	Retrieve the specified profile
Synopsis	GET /ers/config/profilerprofile/{id}
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	Resource of type Profile
Response Status	200, 400, 401,403, 404, 415, 500

Sample Request for Get Portal by ID Call

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/profilerprofile/333
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.profilerprofile.1.0+xml
```

Sample Response for Get Portal by ID Call

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2014 23:59:59 GMT

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:profilerprofile name="name" id="id" description="description"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com"/>
```

External RESTful Services APIs for Network Devices

The following table lists the External RESTful Services APIs for Network Devices:

Table 7-51 APIs Available for Portals

Operation	Method	URL	Content	QueryString
Get All Network Devices	GET	/ers/config/networkdevice	n/a	Page, Size, sortacs or sortdsn, Filter
Get Network Device	GET	/ers/config/networkdevice/{id}	n/a	
Create Network Device	POST	/ers/config/networkdevice	networkdevice	
Update Network Device	PUT	/ers/config/networkdevice/{id}	networkdevice	
Delete Network Device	DELETE	/ers/config/networkdevice/{id}	n/a	
Get Network Device Resource Version Info	GET	/ers/config/networkdevice/versioninfo	n/a	

Get All Network Devices

The following table lists the main characteristics of the Get All Network Devices API call:

Table 7-52 Main Characteristics of Get All Network Devices API Call

Description	Retrieve a collection of network device resources
Synopsis	GET /ers/config/networkdevice
Request Headers	Accept, Authorization, Host
QueryString	page, size, sortbyacn, sortbydcn, filter
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	SearchResult
Response Status	200, 400, 401, 403, 404, 415, 500

Sample Request for Get All Network Devices Call

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/networkdevice?page=1&size=20&sortacs=name
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.network.networkdevice.1.0+xml
```

Sample Response for Get All Network Devices Call

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.searchresult.1.0+xml Content-Length: 16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns2:searchResult
  xmlns:ns2="ers.ise.cisco.com" total="1">
  <resources>
```



```

        <resource name="nd1" id="0d008bb0-2539-11e3-84ad-
00215edbb2a8">
            <link type="application/xml"
href="https://10.56.13.196:9060/ers/config/networkdevice/0d0
08bb0-2539-11e3-84ad-00215edbb2a8" rel="self"/>
        </resource>
    </resources>
</ns2:searchResult>
}

```

Get Network Device by ID

The following table lists the main characteristics of the Get Network Device by ID API call:

Table 7-53 Main Characteristics of Get Network Device by ID API Call

Description	Retrieve the specified network device
Synopsis	GET /ers/config/networkdevice/{id}
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	Resource of type Network Device
Response Status	200, 400, 401, 403, 404, 415, 500

Sample Request for Get Network Device by ID Call

```

GET https://<ISE-ADMIN-NODE>:9060/ers/config/networkdevice/333
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.network.networkdevice.1.0+xml

```

Sample Response for Get Network Device by ID Call

```

HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.network.networkdevice.1.0+xml Content-Length:
16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns3:networkdevice
xmlns:ns2="ers.ise.cisco.com"
xmlns:ns3="network.ers.ise.cisco.com" name="nd1"
id="0d008bb0-2539-11e3-84ad-00215edbb2a8">
    <link type="application/xml"
href="https://10.56.13.196:9060/ers/config/networkdevice/0d0
08bb0-2539-11e3-84ad-00215edbb2a8" rel="self"/>
    <authenticationSettings>
        <enableKeyWrap>false</enableKeyWrap>
        <keyInputFormat>ASCII</keyInputFormat>
        <networkProtocol>RADIUS</networkProtocol>
        <radiusSharedSecret>*****</radiusSharedSecret>
    </authenticationSettings>
<NetworkDeviceIPList>
    <NetworkDeviceIP>
        <ipaddress>1.2.3.4</ipaddress>

```

```

        <mask>32</mask>
      </NetworkDeviceIP>
    </NetworkDeviceIPList>
    <modelName>Unknown</modelName>
    <NetworkDeviceGroupList>
      <NetworkDeviceGroup>1d8c62b0-2539-11e3-84ad-
00215edbb2a8</NetworkDeviceGroup>
      <NetworkDeviceGroup>37053aa0-2539-11e3-84ad-
00215edbb2a8</NetworkDeviceGroup>
    </NetworkDeviceGroupList>
    <softwareVersion>Unknown</softwareVersion>
  </ns3:networkdevice>
}

```

Create Network Device

The following table lists the main characteristics of the Create Network Device API call:

Table 7-54 Main Characteristics of Create Network Device API Call

Description	Create a specified network device
Synopsis	POST /ers/config/networkdevice/
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	NetworkDevice
Response Headers	Content-Length, Content-Type, Location
Response Message Body	N/A
Response Status	200, 400, 401, 403, 415, 500

Sample Request for Create Network Device Call

```

POST https://<ISE-ADMIN-NODE>:9060/ers/config/networkdevice/
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.cisco.ise.network.networkdevice.1.0+xml {
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <ns3:networkdevice
    xmlns:ns2="ers.ise.cisco.com"
    xmlns:ns3="network.ers.ise.cisco.com" name="nd2">
    <authenticationSettings>
      <enableKeyWrap>false</enableKeyWrap>
      <keyInputFormat>ASCII</keyInputFormat>
      <networkProtocol>RADIUS</networkProtocol>
      <radiusSharedSecret>acsi</radiusSharedSecret>
    </authenticationSettings>
    <NetworkDeviceIPList>
      <NetworkDeviceIP>
        <ipaddress>1.2.3.4</ipaddress>
        <mask>32</mask>
      </NetworkDeviceIP>
    </NetworkDeviceIPList>
    <modelName>Unknown</modelName>
    <NetworkDeviceGroupList>
      <NetworkDeviceGroup>1d8c62b0-2539-11e3-84ad-
00215edbb2a8</NetworkDeviceGroup>
      <NetworkDeviceGroup>37053aa0-2539-11e3-84ad-

```

```

    00215edbb2a8</NetworkDeviceGroup>
      </NetworkDeviceGroupList>
      <softwareVersion>Unknown</softwareVersion>
    </ns3:networkdevice>
  }

```

Sample Response for Create Network Device Call

```

HTTP/1.1 201 OK (see location header for the ID of the new device)
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.network.networkdevice.1.0+xml
Location: https://<ISE-ADMIN-NODE>:9060/ers/config/networkdevice/444

```

Update Network Device

The following table lists the main characteristics of the Update Network Device API call:

Table 7-55 Main Characteristics of Update Network Device API Call

Description	Update a specified network device
Synopsis	PUT /ers/config/networkdevice/{id}
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	NetworkDevice
Response Headers	Content-Length, Content-Type
Response Message Body	List of updated fields
Response Status	200, 400, 401, 403, 415, 500

Sample Request for Update Network Device Call

```

PUT https://<ISE-ADMIN-NODE>:9060/ers/config/networkdevice/333
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.network.networkdevice.1.0+xml {
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <ns3:networkdevice
    xmlns:ns2="ers.ise.cisco.com"
    xmlns:ns3="network.ers.ise.cisco.com"
    name="nd2_updated">
    <authenticationSettings>
      <enableKeyWrap>true</enableKeyWrap>
    </authenticationSettings>
  </ns3:networkdevice>
}

```

Sample Response for Update Network Device Call

```

HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.updatedfields.1.0+xml Content-Length: 529
{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns2:updatedFields
  xmlns:ns2="ers.ise.cisco.com">

```

```

<updatedField field="name">
  <newValue>nd2_updated</newValue>
  <oldValue>nd2</oldValue>
</updatedField>
<updatedField field="enableKeywrap">
  <newValue>>true</newValue>
  <oldValue>>false</oldValue>
</updatedField>
</ns2:updatedFields>
}

```

Delete Network Device

The following table lists the main characteristics of the Delete Network Device API call:

Table 7-56 Main Characteristics of Delete Network Device API Call

Description	Delete a specified network device
Synopsis	DELETE /ers/config/networkdevice/{id}
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	N/A
Response Status	200, 204, 400, 401, 403, 404, 415, 500

Sample Request for Update Network Device Call

```

DELETE https://<ISE-ADMIN-NODE>:9060/ers/config/networkdevice/333
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.network.networkdevice.1.0+xml

```

Sample Response for Update Network Device Call

```

HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT

```

External RESTful Services APIs for Network Device Groups

The following table lists the External RESTful Services APIs for Network Device Groups:

Table 7-57 APIs Available for SGTs

Operation	Method	URL	Content	QueryString
Get All Network Device Groups	GET	/ers/config/networkdevicegroup	n/a	page, size, sortacs or sortdsn, filter

Table 7-57 APIs Available for SGTs (continued)

Operation	Method	URL	Content	QueryString
Get Network Device Group	GET	/ers/config/networkdevicegroup/{id}	n/a	
Get Network Device Group Resource Version Info	GET	/ers/config/networkdevicegroup/versioninfo	n/a	

Get All Network Device Groups

The following table lists the main characteristics of the Get All Network Device Groups API call:

Table 7-58 Main Characteristics of Get All Network Device Groups API Call

Description	Retrieve a collection of Network Device Groups resources
Synopsis	GET /ers/config/networkdevicegroup
Request Headers	Accept, Authorization, Host
QueryString	page, size, sortbyacn, sortbydcn, filter
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	SearchResult
Response Status	200, 400, 401, 403, 404, 415, 500

Sample Request for Get All Network Device Groups API Call

```
GET
https://<ISE-ADMIN-NODE>:9060/ers/config/networkdevicegroup?page=1&size=20&sortacs=name
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.network.networkdevicegroup.1.0+xml
```

Sample Response for Get All Network Device Groups API Call

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.searchresult.1.0+xml Content-Length: 16347
{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
    <ns2:searchResult
      xmlns:ns2="ers.ise.cisco.com" total="0">
      <resources>
        <resource name="Location#All Locations#loc1" id="1d8c62b0-2539-11e3-84ad-00215edbb2a8"
          description="xxx">
          <link type="application/xml"
            href="https://10.56.13.196:9060/ers/config/networkdevicegroup/1d8c62b0-2539-11e3-84ad-00215edbb2a8" rel="self"/>
        </resource>
      </resources>
    </ns2:searchResult>
  </?xml>
}
```

```

    <resource name="Device Type#All Device Types#device type 555"
    id="37053aa0-2539-11e3-84ad-00215edbb2a8" description="vvv">
      <link type="application/xml"
    href="https://10.56.13.196:9060/ers/config/networkdevicegrou
    p/37053aa0-2539-11e3-84ad-00215edbb2a8" rel="self"/>
    </resource>
  </resources>
</ns2:searchResult>
}

```

Get Network Device Group

The following table lists the main characteristics of the Get Network Device Groups API call:

Table 7-59 Main Characteristics of Get Network Device Group API Call

Description	Retrieve the specified Network Device group
Synopsis	GET /ers/config/networkdevicegroup/{id}
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	Resource of type NetworkDeviceGroup
Response Status	200, 400, 401, 403, 404, 415, 429, 500

Sample Request for Get Network Device Group API Call

```

GET https://<ISE-ADMIN-NODE>:9060/ers/config/networkdevicegroup/333
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.network.networkdevicegroup.1.0+xml

```

Sample Response for Get Network Device Group API Call

```

HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.network.networkdevicegroup.1.0 +xml
Content-Length: 16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns3:networkdevicegroup
  xmlns:ns2="ers.ise.cisco.com"
  xmlns:ns3="network.ers.ise.cisco.com" name="Location#All
  Locations#loc1" id="1d8c62b0-2539-11e3-84ad-00215edbb2a8"
  description="xxx">
  <link type="application/xml"
  href="https://10.56.13.196:9060/ers/config/networkdevicegrou
  p/1d8c62b0-2539-11e3-84ad-00215edbb2a8" rel="self"/>
  <type>Location</type>
</ns3:networkdevicegroup>
}

```

External RESTful Services APIs for SGTs

The following table lists the External RESTful Services APIs for SGTs:

Table 7-60 APIs Available for SGTs

Operation	Method	URL	Content	QueryString
Get All SGTs	GET	/ers/config/sgt	n/a	page, size, sortacs or sortdsn, filter
Get SGT	GET	/ers/config/sgt/{id ¹ }	n/a	
Get GST Resource Version Info	GET	/ers/config/sgt/versioninfo	n/a	

1. The SGT ID is the UUID type as stored in the Cisco ISE database.

Get All SGTs

The following table lists the main characteristics of the Get All SGTs API call:

Table 7-61 Main Characteristics of Get All SGTs API Call

Description	Retrieve a collection of SGT resources
Synopsis	GET /ers/config/sgt
Request Headers	Accept, Authorization, Host
QueryString	page, size, sortbyacn, sortbydcn, filter
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	SearchResult
Response Status	200, 400, 401, 403, 404, 415, 429, 500

Sample Request for Get All SGTs API Call

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/sgt?page=0&size=20&sortacs=name
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.sga.sgt.1.0+xml
```

Sample Response for Get All SGTs API Call

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.searchresult.1.0+xml
Content-Length: 16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult total="2" xmlns:ns2="ers.ise.cisco.com">
<resources>
  <resource name="name1" id="id1" description="description1">
    <link type="application/xml" href="https://<ISE-ADMIN-NODE>:9060/ers/config/sgt/id1"
rel="self"/>
  </resource>
</resources>
</ns2:searchResult>
</?xml>
```

```

    </resource>
  </resources>
</ns2:searchResult>
}

```

Get SGT by ID

The following table lists the main characteristics of the Get SGT by ID API call:

Table 7-62 Main Characteristics of Get SGTs API Call

Description	Retrieve the specified SGT
Synopsis	GET /ers/config/sgt/{id}
Request Headers	Accept, Authorization, Host
QueryString	N/A
Request Message Body	N/A
Response Headers	Content-Length, Content-Type
Response Message Body	Resource of type InternalUser
Response Status	200, 400, 401, 403, 404, 415, 429, 500

Sample Request for Get SGT by ID API Call

```

GET https://<ISE-ADMIN-NODE>:9060/ers/config/sgt/333
Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.sga.sgt.1.0+xml

```

Sample Response for Get SGT by ID API Call

```

HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.sga.sgt.1.0+xml
Content-Length: 16347
{
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <ns3:sgt description="description" name="name" id="id" xmlns:ns2="ers.ise.cisco.com"
  xmlns:ns3="sga.ers.ise.cisco.com">
    <generationId>generationId</generationId>
    <isTagFromRange>isTagFromRange</isTagFromRange>
    <value>1</value>
  </ns3:sgt>
}

```


REST API Client

The External RESTful Services APIs enable you to perform CRUD (Create, Read, Update, Delete) operations on Cisco ISE resources. To build and test applications using the External RESTful Services APIs that communicate with and perform operations on Cisco ISE servers, you can use any industry standard REST API client, such as the POSTMAN plugin for Google Chrome.

Designed according to REST architecture and principles, POSTMAN enables you to send and retrieve standard HTTP and HTTPS requests and responses using the Google Chrome web browser. You can use the following standard HTTP methods to perform CRUD operations on Cisco ISE resources:

- GET
- POST
- PUT
- DELETE

The ERS API enables you to use these HTTP requests in various API calls, which in turn enable you to perform operations on the Cisco ISE servers. For a comprehensive list of operations in which these HTTP requests are used, see <ERS API Operations>.

**Note**

To download the POSTMAN plugin, go to <https://chrome.google.com/webstore/detail/postman-rest-client/fdmmgilgnpjigdojojjjoooidkmcomcm?hl=en>. For more information on using the POSTMAN plugin, go to <https://github.com/a85/POSTMan-Chrome-Extension/wiki>.

GET Method

Requests a representation of the specified resource. Requests using GET only retrieve data and do not have any other effect.

**Note**

This section shows how to use the POSTMAN plugin to invoke an ERS API call. This API call uses the GET HTTP method in addition to other components of the ERS API, which are not described in this section. For more details on various ERS API components such as the characteristics, requests, and responses, see [External RESTful Services API Operations](#).

The request body of the ERS API call that uses the GET HTTPS method contains the following three building blocks:

- [URI](#)
- [Accept Header](#)
- [Authorization Header](#)

URI

The GET method sends the URI to the Cisco ISE server and the HTTP reply is the raw result data. A typical URI must adhere to the following format:

- *[https://<Cisco ISE Server address:<port>/<namespace>/config/<Cisco ISE Resource Name>](#)*

Where *<Cisco ISE Server Address>* denotes the server address of the Cisco ISE server, *<port>* denotes the port 9060, *<namespace>* denotes the namespace to which the ISE Resource belongs to, and *<Cisco ISE Resource Name>* denotes the name of the Cisco ISE Resource.

The following example shows the URI that requests data for the *internaluser* ISE Resource:

- *https://10.56.13.196:9060/ers/config/internaluser.*

**Note**

The URI is not the request body; it is just a URL. This URL is sent to the server using the GET method.

Accept Header

The Accept Header must adhere to the following format:

- *application/vnd.com.cisco.ise.<resource-namespace>.<resource-type>.<major version>.<minor version>+xml*

Where *<resource-namespace>* denotes the namespace to which the ISE Resource belongs to, *<resource-type>* denotes the type of the ISE Resource, *<major-version>* denotes the major version number of the ISE deployment, and *<minor-version>* denotes the minor version number of the ISE deployment.

The following example shows a typical accept header:

- *application/vnd.com.cisco.ise.identity.internaluser.1.0+xml*

Authorization Header

The Authorization Header contains the encryption authorization key that is embedded into the GET request. After specifying the authorization credentials, you must generate the encryption key, which is then embedded into the request body.

**Note**

For more information on generating the encryption key, see [Making the GET Request Using POSTMAN, page 7-58](#).

Making the GET Request Using POSTMAN

Procedure

Step 1 Open the POSTMAN plugin in the Google Chrome browser.

Step 2 Create a new collection using the options in the left pane.



Note For more information on using the POSTMAN plugin, go to <https://github.com/a85/POSTMan-Chrome-Extension/wiki>.

Step 3 From the drop-down menu, choose **GET**.

Step 4 In the URL bar, enter the URI.

The URI specifies the Cisco ISE server with which you are trying to communicate and the ISE resource that you are trying to access. For more information on the format of the URI, see [URI, page 7-57](#).

Step 5 Click the **Basic Auth** tab.

The options that enable you to specify the user access credentials appear.

Step 6 Specify your access credentials in the Username and Password fields and click **Refresh Headers**.

POSTMAN displays an Authorization header with an encryption key.

Step 7 Add an accept header by specifying the following value:

application/vnd.com.cisco.ise.ers.<namespace>.<ise resource>.1.0+xml



Note For more information on the Accept Header, see [Accept Header, page 7-58](#).

Step 8 Click **Send**.

The POSTMAN plugin displays a 200 OK status response indicating that the request is successful. The request also returns the details of the resources that you have specified in the URL.

POST Method

Requests that the server accept the entity enclosed in the request as a new subordinate of the web resource identified by the URI.



Note

This section shows how to use the POSTMAN plugin to invoke an ERS API call. This API call uses the POST HTTP method in addition to other components of the ERS API, which are not described in this section. For more details on various ERS API components such as the characteristics, requests, and responses, see [External RESTful Services API Operations](#).

The request body of the ERS API call that uses the POST HTTP method contains the following three building blocks:

- [URI](#)
- [Content-Type Header](#)
- [Authorization Header](#)

URI

The POST method sends the URI to the Cisco ISE server. A typical URI must adhere to the following format:

- *https://<Cisco ISE Server address:<port>/<namespace>/config/<Cisco ISE Resource Name>*

Where *<Cisco ISE Server Address>* denotes the server address of the Cisco ISE server, *<port>* denotes the port 9060, *<namespace>* denotes the namespace to which the ISE Resource belongs to, and *<Cisco ISE Resource Name>* denotes the name of the Cisco ISE Resource.

The following example shows the URI that requests data for the *internaluser* ISE Resource:

- *https://10.56.13.196:9060/ers/config/internaluser.*



Note

The URI is not the request body; it is just a URL. This URL is sent to the server using the POST method.

Content-Type Header

The Content-Type Header must adhere to the following format:

- `application/vnd.com.cisco.ise.<resource-namespace>.<resource-type>.<major version>.<minor version>+xml`

Where `<resource-namespace>` denotes the namespace to which the ISE Resource belongs to, `<resource-type>` denotes the type of the ISE Resource, `<major-version>` denotes the major version number of the ISE deployment, and `<minor-version>` denotes the minor version number of the ISE deployment.

The following example shows a typical accept header:

- `application/vnd.com.cisco.ise.identity.internaluser.1.0+xml`

Authorization Header

The Authorization Header contains the encryption authorization key that is embedded into the POST request. After specifying the authorization credentials, you must generate the encryption key, which is then embedded into the request body.

**Note**

For more information on generating the encryption key, see [Making the POST Request Using POSTMAN, page 7-60](#).

Making the POST Request Using POSTMAN

Procedure

Step 1 Open the POSTMAN plugin in the Google Chrome browser.

Step 2 Create a new collection using the options in the left pane.

**Note**

For more information on using the POSTMAN plugin, go to <https://github.com/a85/POSTMan-Chrome-Extension/wiki>.

Step 3 From the drop-down menu, choose **POST**.

Step 4 In the URI bar, enter the URI.

The URI specifies the Cisco ISE server with which you are trying to communicate and the ISE resource that you are trying to access. For more information on the format of the URI, see [URI, page 7-59](#).

Step 5 Click the **Basic Auth** tab.

The options that enable you to specify the user access credentials appear.

Step 6 Specify your access credentials in the Username and Password fields and click **Refresh Headers**.

POSTMAN displays an Authorization header with an encryption key.

Step 7 Add a Content-Type header by specifying the following value:

`application/vnd.com.cisco.ise.ers.<namespace>.<ise resource>.1.0+xml`

**Note**

For more information on the Accept Header, see [Content-Type Header, page 7-60](#).

- Step 8** From the drop-down menu that appears next to the raw button, choose **XML**.
- Step 9** Click **raw**.
- Step 10** The POSTMAN plugin opens an editing pane that enables you to specify the body of the POST request.
- Step 11** Enter the message body of your POST request in the editing pane.

**Note**

This message body must contain the details corresponding to the ISE resource that you trying to create on the ISE server. For example, while creating an interaluser, you must specify details such as the name of internaluser, description of the interaluser, password, and so on. For more details on the message body of the ERS APIs that use the POST request and the details of the ISE resources that you need to specify, see [External RESTful Services API Operations](#).

- Step 12** Click **Send**.
- The POSTMAN plugin displays a 201 CREATED status response indicating that the request is successful. You can go to the ISE GUI to verify whether the ISE resource you have added appears in the ISE GUI.

PUT Method

Requests that the enclosed entity be stored under the supplied URI. If the URI refers to an already existing resource, it is modified; if the URI does not point to an existing resource, then the server can create the resource with that URI.

**Note**

This section shows how to use the POSTMAN plugin to invoke an ERS API call. This API call uses the PUT HTTP method in addition to other components of the ERS API, which are not described in this section. For more details on various ERS API components such as the characteristics, requests, and responses, see [External RESTful Services API Operations](#).

The request body of the ERS API call that uses the POST HTTP method contains the following three building blocks:

- [URI](#)
- [Content-Type Header](#)
- [Authorization Header](#)

URI

The PUT method sends the URI to the Cisco ISE server. A typical URI must adhere to the following format:

- *https://<Cisco ISE Server address:<port>/<namespace>/config/<Cisco ISE Resouce Name>*
Where *<Cisco ISE Server Address>* denotes the server address of the Cisco ISE server, *<port>* denotes the port 9060, *<namespace>* denotes the namespace to which the ISE Resource belongs to, and *<Cisco ISE Resource Name>* denotes the name of the Cisco ISE Resource.

The following example shows the URI that requests data for the *interaluser* ISE Resource:

- *https://10.56.13.196:9060/ers/config/internaluser.*

**Note**

The URI is not the request body; it is just a URL. This URL is sent to the server using the PUT method.

Content-Type Header

The Content-Type Header must adhere to the following format:

- *application/vnd.com.cisco.ise.<resource-namespace>.<resource-type>.<major version>.<minor version>+xml*

Where *<resource-namespace>* denotes the namespace to which the ISE Resource belongs to, *<resource-type>* denotes the type of the ISE Resource, *<major-version>* denotes the major version number of the ISE deployment, and *<minor-version>* denotes the minor version number of the ISE deployment.

The following example shows a typical accept header:

- *application/vnd.com.cisco.ise.identity.internaluser.1.0+xml*

Authorization Header

The Authorization Header contains the encryption authorization key that is embedded into the PUT request. After specifying the authorization credentials, you must generate the encryption key, which is then embedded into the request body.

**Note**

For more information on generating the encryption key, see [Making the PUT Request Using POSTMAN, page 7-62](#).

Making the PUT Request Using POSTMAN

Procedure

Step 1 Open the POSTMAN plugin in the Google Chrome browser.

Step 2 Create a new collection using the options in the left pane.



Note For more information on using the POSTMAN plugin, go to <https://github.com/a85/POSTMan-Chrome-Extension/wiki>.

Step 3 From the drop-down menu, choose **PUT**.

Step 4 In the URI bar, enter the URI.

The URI specifies the Cisco ISE server with which you are trying to communicate and the ISE resource that you are trying to access. For more information on the format of the URI, see [URI, page 7-61](#).

Step 5 Click the **Basic Auth** tab.

The options that enable you to specify the user access credentials appear.

Step 6 Specify your access credentials in the Username and Password fields and click **Refresh Headers**.

POSTMAN displays an Authorization header with an encryption key.

Step 7 Add a Content-Type header by specifying the following value:
application/vnd.com.cisco.ise.ers.<namespace>.<ise resource>.1.0+xml



Note For more information on the Accept Header, see [Content-Type Header, page 7-62](#).

Step 8 From the drop-down menu that appears next to the raw button, choose **XML**.

Step 9 Click **raw**.

Step 10 The POSTMAN plugin opens an editing pane that enables you to specify the body of the POST request.

Step 11 Enter the message body of your POST request in the editing pane.



Note This message body must contain the details corresponding to the ISE resource that you trying to update on the ISE server. For example, while updating an internaluser, you must specify details such as the name of internaluser, description of the internaluser, password, and so on. For more details on the message body of the ERS APIs that use the POST request and the details of the ISE resources that you need to specify, see [External RESTful Services API Operations](#).

Step 12 Click **Send**.

The POSTMAN plugin displays a 201 CREATED status response indicating that the request is successful. You can go to the ISE GUI to verify whether the ISE resource you have added appears in the ISE GUI.

Delete Method

Deletes the specified resource.



Note This section shows how to use the POSTMAN plugin to invoke an ERS API call. This API call uses the DELETE HTTP method in addition to other components of the ERS API, which are not described in this section. For more details on various ERS API components such as the characteristics, requests, and responses, see [External RESTful Services API Operations](#).

The request body of the ERS API call that uses the DELETE HTTP method contains the following three building blocks:

- [URI](#)
- [Accept Header](#)
- [Authorization Header](#)

URI

The DELETE method sends the URI to the Cisco ISE server. A typical URI must adhere to the following format:

- *https://<Cisco ISE Server address:<port>/<namespace>/config/<Cisco ISE Resouce Name>*

Where *<Cisco ISE Server Address>* denotes the server address of the Cisco ISE server, *<port>* denotes the port 9060, *<namespace>* denotes the namespace to which the ISE Resource belongs to, and *<Cisco ISE Resource Name>* denotes the name of the Cisco ISE Resource.

The following example shows the URI that requests data for the *internaluser* ISE Resource:

- *https://10.56.13.196:9060/ers/config/internaluser.*

**Note**

The URI is not the request body; it is just a URL. This URL is sent to the server using the GET method.

Accept Header

The Accept Header must adhere to the following format:

- *application/vnd.com.cisco.ise.<resource-namespace>.<resource-type>.<major version>.<minor version>+xml*

Where *<resource-namespace>* denotes the namespace to which the ISE Resource belongs to, *<resource-type>* denotes the type of the ISE Resource, *<major-version>* denotes the major version number of the ISE deployment, and *<minor-version>* denotes the minor version number of the ISE deployment.

The following example shows a typical accept header:

- *application/vnd.com.cisco.ise.identity.internaluser.1.0+xml*

Authorization Header

The Authorization Header contains the encryption authorization key that is embedded into the DELETE request. After specifying the authorization credentials, you must generate the encryption key, which is then embedded into the request body.

**Note**

For more information on generating the encryption key, see [Making the DELETE Request Using POSTMAN, page 7-64](#).

Making the DELETE Request Using POSTMAN

Procedure

Step 1 Open the POSTMAN plugin in the Google Chrome browser.

Step 2 Create a new collection using the options in the left pane.



Note For more information on using the POSTMAN plugin, go to <https://github.com/a85/POSTMan-Chrome-Extension/wiki>.

Step 3 From the drop-down menu, choose **DELETE**.

Step 4 In the URL bar, enter the URI.

The URI specifies the Cisco ISE server with which you are trying to communicate and the ISE resource that you are trying to access. For more information on the format of the URI, see [URI, page 7-63](#).

Step 5 Click the **Basic Auth** tab.

The options that enable you to specify the user access credentials appear.

Step 6 Specify your access credentials in the Username and Password fields and click **Refresh Headers**.

POSTMAN displays an Authorization header with an encryption key.

Step 7 Add an accept header by specifying the following value:

application/vnd.com.cisco.ise.ers.<namespace>.<ise resource>.1.0+xml



Note For more information on the Accept Header, see [Accept Header, page 7-64](#).

Step 8 Click **Send**.

The POSTMAN plugin displays a 200 OK status response indicating that the request is successful. The ISE resource that you have specified is deleted from the ISE server.
