Introduction to ERS APIs

Overview

This chapter provides guidelines and examples for using the supported External RESTful Services APIs and related API calls. These calls enable you to perform CRUD (Create, Read, Update, Delete) operations on Cisco ISE resources.

Supported Cisco ISE Resources

The Cisco ISE External RESTful Services allow you to perform operations on the following types of ISE resources:

- End points
- End point identity groups
- Guest users

- Identity groups
- Internal users
- Portals
- Profiler policies
- Network devices
- Network device groups
- Security groups

Full request and response examples are provided in Chapter 7, "External RESTful Services API Operations.".

External RESTful Services API Authentication and Authorization

The External RESTful Services APIs are based on HTTPS protocol and REST methodology and uses port 9060.

The External RESTful Services APIs support basic authentication. The authentication credentials are encrypted and are part of the request header.

The ISE administrator must assign special privileges to a user to perform operations using the External RESTful Services APIs. The ISE administrator can assign the following two roles to perform operations using the External RESTful Services APIs:

- External RESTful Services Admin—For full access to all ERS APIs (GET, POST, DELETE, PUT).
- External RESTful Services Operator—For Read Only access (GET request only).

If you do not have the required permissions and still try to perform operations using the External RESTful Services APIs, you will receive an error response.

Related Topics

- Creating a New Cisco ISE Administrator
- Sponsor Authentication and Authorization, page 6-1

Enabling External RESTful Services APIs from the GUI

You must enable the Cisco ISE REST API in order for applications developed for a Cisco ISE REST API to be able to access Cisco ISE. The Cisco REST APIs uses HTTPS port 9060, which is closed by default. If the Cisco ISE REST APIs are not enabled on the Cisco ISE admin server, the client application will receive a time-out error from the server for any Guest REST API request.

External RESTful Service requests of all types are valid only for the primary ISE node. Secondary nodes have read-access (GET requests).

Procedure

Step 1 Enter the Cisco ISE URL in the address bar of your browser (for example, `https://<ise hostname or ip address>/admin/`).

Step 2 Enter your username and case-sensitive password.

Step 3 Click **Login** or press **Enter**.

Step 4 Choose **Administration > Settings > ERS Settings**.

Step 5 Choose **Enable ERS for Read/Write** for the Primary Administration Node.

Step 6 Choose **Enable ERS for Read** for All Other Nodes if there are any secondary nodes.

Step 7 Click **Submit**.

Note All REST operations are audited and the logs are logged in the system logs.

Related Topics

- Prerequisites for Using the External RESTful Services API Calls, page 7-1

External RESTful Services API Status

You can verify whether the External RESTful Services APIs are enabled for the primary and secondary nodes on the **Administration > Settings > ERS Settings** page in the GUI.

The External RESTful Services APIs are not enabled by default. If you try to evoke the External RESTful Services API calls before enabling them, you will receive an error response.

External RESTful Services APIs have a debug logging category, which you can enable from the debug logging page of the Cisco ISE GUI.

Note For more information, see the Debug Log Configuration Options section of the *Cisco Identity Services Engine Admin Guide, Release 1.3*.

Data Validation

CRUD data sent to the server is validated with the same validation rules that Cisco ISE uses for the GUI. All validations are centralized in a validation layer. All XML data being posted is validated against the schema.

The following two types of validations occur: data validation and structural validation. Data validation validates the data to be ISE compliant, for example, mandatory fields, field length, types, and so on. Where as, structural validation happens against the schema, for example, fields order, names, and so on.

Namespaces

You must maintain strict namespaces within resources names and URIs as follows:

- Identity for internal user, endpoint, endpoint groups and identity groups
- SGA for SGT
- External RESTful Services for all other object resources such as search results that appears in the response message (the client do need to send ERS namespace in requests)

The Accept/Content-Type headers must contain the following namespaces:

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

For example: `application/vnd.com.cisco.ise.identity.internaluser.1.0+xml`

The request XML should contain the namespace definition as follows:

```
identity.ers.ise.cisco.com

sga.ers.ise.cisco.com
```

For example:

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

<ns:endpoint xmlns:ns="identity.ers.ise.cisco.com" id="id">

<group>Profiled</group>

…

</ns:endpoint>
```

External RESTful Services SDK

You can use the External RESTful Services SDK to start building your own tools. You can access the External RESTful Services SDK from the following URL: `https://<ISE-ADMIN-NODE>:9060/ers/sdk`.

External RESTful Services SDK can be accessed by the External RESTful Services Admin users only. The SDK consists the following components:

- Quick reference API documentation
- List of all available API operations
- Schema files available for download
- Sample application in Java available for download
- Use cases in curl script format
- Instructions on using Chrome Postman

External RESTful Services Schema File

The External RESTful Services SDK is shipped with three XSD schema files that describe the structure of the objects that are supported on ISE ERS interfaces.

The three XSD files are:

- ers.xsd
- identity.xsd
- sga.xsd
- network.xsd

You can use the schema with available tools such as JAXB to generate schema classes.

You can develop HTTP client or use any third-party HTTP client code and integrate it with the schema classes that are generated from the XSD files.

Note The XML sent in the content is validated against the schema, therefore, field order and syntax in the XML should be the same as it appears in the schema. Otherwise, you will receive a bad request status code.

Downloading the Schema File

Procedure

Step 1 Enter the following URL in the address bar of your browser to log into the External RESTful Services SDK page: `https://<ISE-ADMIN-NODE>:9060/ers/sdk`

Step 2 Enter the Username and the case-sensitive password corresponding to an External RESTful Services Admin.

Step 3 Click **Login** or press **Enter**.

Step 4 In the **Downloads** category, click **Schema Jar (ers-schema-1.3-iteration-01-SNAPSHOT.jar)**.

Step 5 Save the file to your local machine.

Related Topics

- External RESTful Services API Authentication and Authorization, page 5-2

External RESTful Service Requests and Responses

This section provides information on the request and response headers as well as the status codes returned by ERS.

External RESTful Service Requests Headers

Table 5-1 ERS Request Headers

Header	Supported Values	Description of Use	Required
Accept	Guest REST API Resource Media-Type	Indicates to the server what media type(s) this client is willing to accept including the resource version.	Yes, on GET/GET ALL/DELETE/GET VERSION operations (these contain no message body)
Authorization	"Basic " plus username and password (per RFC 2617	Identifies the authorized user making this request.	Yes, on all requests.
Content-Length	Guest REST API Resource Media-Type	Indicates to the server what media type(s) this client is willing to accept including the resource version.	Yes, on requests that contain a message body.
Content-Type	Media type describing the request message body	Describes the representation and syntax of the request message body.	Yes, on requests that contain a message body.

External RESTful Service Response Headers

Table 5-2 ERS Mandatory Response Headers

Header	Supported Values	Description of Use	Required
Content-Length	Length (in bytes) of the response message body.	Describes the size of the message body.	Yes, on responses that contain a message body.
Content-Type	Media type describing the response message body.	Describes the representation and syntax of the response message body.	Yes, on responses that contain a message body.
Location	Canonical URI of a newly created resource.	Returns a new URI that can be used to request a representation of the newly created resource.	Yes, on responses to requests that create new server side resources accessible via a URI.

Common External RESTful Service HTTP Status Codes

Table 5-3 Description of the HTTP Response Codes Returned By ERS

HTTP Status	Description
200 OK	The request was successfully completed. If this request created a new resource that is addressable with a URI, and a response body is returned containing a representation of the new resource, a 200 status will be returned with a Location header containing the canonical URI for the newly created resource.
201 Created	A request that created a new resource was completed, and no response body containing a representation of the new resource is being returned. A Location header containing the canonical URI for the newly created resource should also be returned.
202 Accepted	The request has been accepted for processing, but the processing has not been completed. Per the HTTP/1.1 specification, the returned entity (if any) SHOULD include an indication of the request's current status, and either a pointer to a status monitor or some estimate of when the user can expect the request to be fulfilled.
204 No Content	The server fulfilled the request, but does not need to return a response message body.
400 Bad Request	The request could not be processed because it contains missing or invalid information (such as validation error on an input field, a missing required value, and so on).
401 Unauthorized	The authentication credentials included with this request are missing or invalid.
403 Forbidden	The server recognized your credentials, but you do not possess authorization to perform this request.
404 Not Found	The request specified a URI of a resource that does not exist.
405 Method Not Allowed	The HTTP verb specified in the request (DELETE, GET, HEAD, POST, PUT) is not supported for this request URI.

Table 5-3 Description of the HTTP Response Codes Returned By ERS (continued)

HTTP Status	Description
406 Not Acceptable	The resource identified by this request is not capable of generating a representation corresponding to one of the media types in the Accept header of the request.
409 Conflict	A creation or update request could not be completed, because it would cause a conflict in the current state of the resources supported by the server (for example, an attempt to create a new resource with a unique identifier already assigned to some existing resource).
415 Unsupported Media Type	The media type specified in the Accept header is not supported by the server. This will be the common response when the client resources version is no longer supported by the server.
429 Too many requests	There are too many simultaneous ERS requests.
500 Internal Server Error	The server encountered an unexpected condition which prevented it from fulfilling the request.
501 Not Implemented	The server does not (currently) support the functionality required to fulfill the request.
503 Service Unavailable	The server is currently unable to handle the request due to temporary overloading or maintenance of the server.

Note In addition to the status codes return in the response header, each request might have additional xml content according to the nature of the request.

Version Control with External RESTful Services APIs

The External RESTful Services APIs provide backward compatibility with previous Cisco ISE versions. The External RESTful Services APIs has a versioning mechanism for API version management. All non-guest resources are of version 1.0 and no backward compatibility is required.

Each RESTful resource has a model version (major.minor). The version must be part of the request header with the syntax as follows:

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

For example, to get internal user resource version 1.0, the following request is passed:

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

After authenticating and authorizing the request, a version match check is performed with the matching results mentioned in the following table:

Versions match	Outcome
No version sent	The server returns status 415 “Unsupported Media Type”
Client version equal to server version	The server proceeds with processing the request.
Client minor version not equal to server minor version	The server adds a response warning message describing the versions gap and proceeds processing the request.
Client and server major version does not match	Server returns status 415 with a corresponding error message.

Note Each resource has an API to retrieve a list of server supported versions.

Searching and Filtering

All filtering and searching operations are through the use of filtering.

You search for a resource by sending a GET request to a resource URI. By default, the result is the first page (page index = 0) with default size of 20. By adding filter, sort and paging parameters, described in the following sections, to the URI, the client can control this search.

The resources resulting from a paging, filter, or sort request are bundled in a `<resources>` collection, which contains for each resource its name, ID, description, and link to its full representation. This gives the client the ability to easily drill down into the resource.

Filtering Parameters for External RESTful Services APIs

You can perform simple filtering operation through the filter query string parameter. You can send more than one filter. The logical operator common to all filter criteria is by default AND. You can change this by using the “filtertype=or” query string parameter.

Each resource data model description should specify if an attribute is a filtered field.

For example, to get internal users with first name starting with ‘a’ and belonging to the identity group ‘Finance’, the following request is passed:

```
GET
/ers/config/internaluser/?page=0&size=20&sortacs=name&filter=name.STARTSW.a&filter=identit
yGroup.EQ.Finance
```

The following table shows the available filter operators:

Parameter	Description
EQ	Equals
GT	Greater Than
LT	Less Then
STARTSW	Start With

Parameter	Description
ENDSW	End With
CONTAINS	Contains

The following table shows the list of filterable attributes for each resource:

Resource	Filterable Attributes
Endpoints	mac, portalUser, profile, profileId, staticGroupAssignment, staticProfileAssignment
Internal User	name
Endpoint Identity Group	name
Identity Groups	name
SGTs	name

Paging Parameters for External RESTful Services APIs

All External RESTful Servive search results are paged. Paging parameter are passed in the URI using query parameters.

For example, to get internal users first 20 records sorted in accending order by “name” field, the following request is passed:

```
GET /ers/config/internaluser?page=0&size=20&sortasc=name
```

The following table shows the available paging parameters:

Parameter	Description	Default Value
page	Page starting index	0
size	Page size	20 (max. = 100)
sortasc	Sorting field with Ascending direction, out of a set of available fields for sorting. (For alphabetic characters, it is sorted A to Z.)	`name`
sortdsc	Sorting field with Descending direction, out of a set of available fields for sorting. (For alphabetic characters, it is sorted Z to A.)	`name`

External RESTful Services System Flow

Common External RESTful Services flow consists of HTTPS request sent from client and an HTTPS response from server. The flow differs by requests types, URIs, request headers, response headers, and response contents.

Figure 5-1 ERS Success Flow Sequence

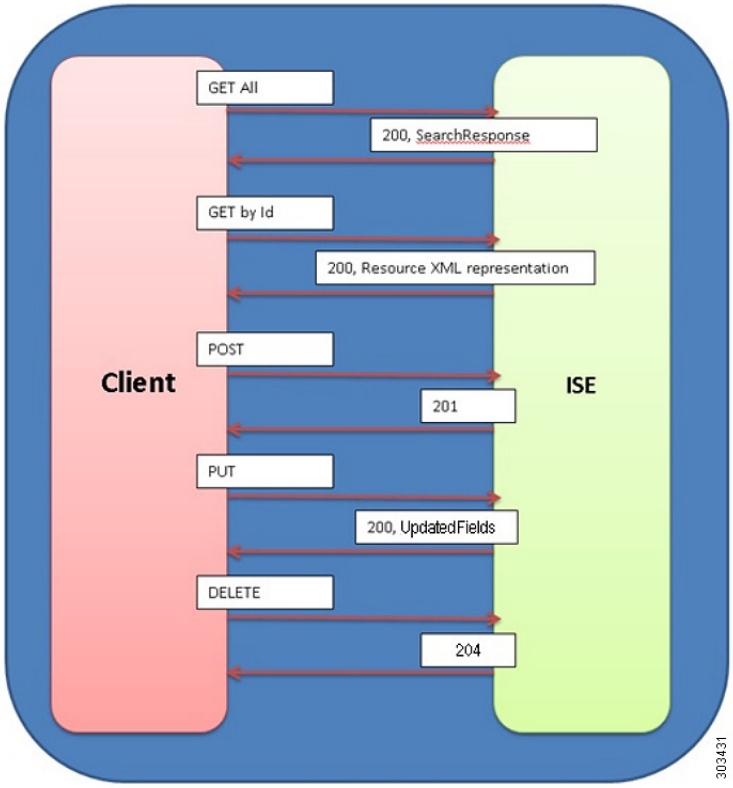
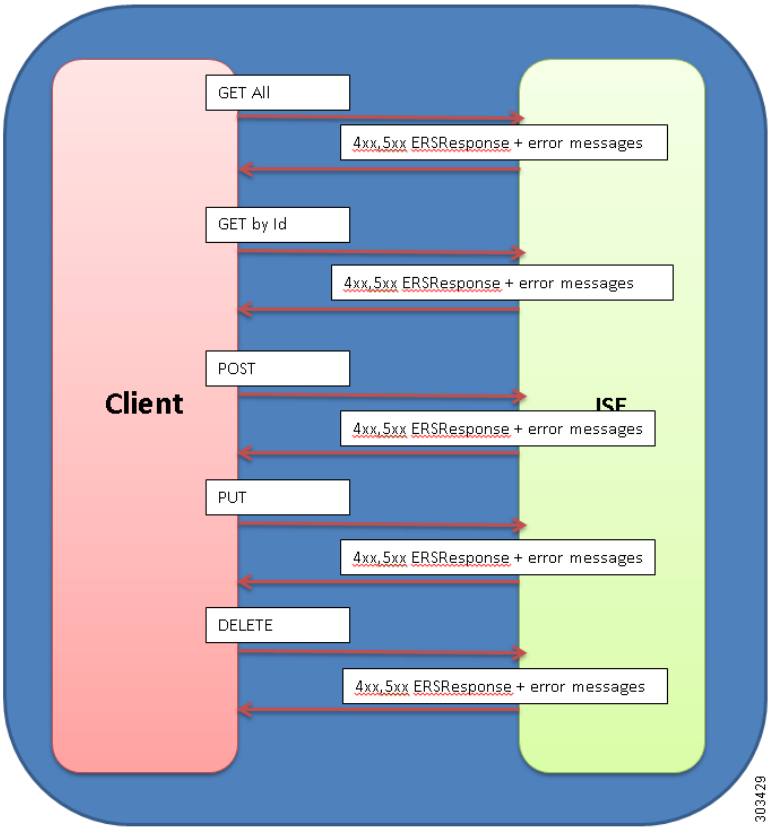


Figure 5-2 ERS Failure Flow Sequence



Hyperlinks

Usage of hyperlinks within XML representation is one of the strongest characteristics of Hypermedia as the Engine of Application State (HATEOS). Hyperlinks are mostly used to convey to the client that the resource has a representation at the given URI.

External RESTful Services links are aligned with the Atom definition of links declared in the following namespace:

http://www.w3.org/2005/Atom namespace.

The following table shows the mandatory link attributes for the External RESTful Services:

Attribute	Description
Href	This contains the link's URI.
Rel	This attribute, which originally meant “relation,” indicates the following type of links: • “self”—link to refresh current representation • “next”—in collection for getting next page • “info”—get more info about of the resource
Type	This is a hint to the media type of the representation that the server may return for the link's URI - usually is “application/xml”

Example of links within a search result

The following example shows links within a search result:

```
<ns2:searchResult xmlns:ns2="ers.ise.cisco.com"   total="1163">
<link rel="self"
href="http://cisco.com/ers/config/internaluser?page=0&size=20"
type="application/xml"/>
<link rel="next" href="http://cisco.com/ers/config/internaluser?page=20&size=20"
type="application/xml"/>
<resources>
<link rel="john doe" href="http://cisco.com/ers/config/internaluser/333"
type="application/xml"/>
<link rel="jeff smit" href="http://cisco.com/ers/config/internaluser/444"
type="application/xml"/>
.
.
.
</resources>
</ns2:searchResult>
```

Bulk Operations

A bulk request will allow you to send up to 500 operations in a single request (or 5000 operations by ID). You can run create, update, and delete operations all resources as well as some resource-specific operations like registering end points.

All operations in a request must be of same type, which means that you cannot send a mixed resources request.

Each resource has its own transaction and the order of the transactions is not guarantee as it is a multi-threaded execution.

The Cisco ISE server parses the request and validates its structure. If the request is valid and no other bulk already in progress, the execution starts and the server returns the status code `202 (ACCEPTED)` and a unique bulk identifier in the LOCATION response header. This ID allows you to track the bulk status later on using the Get Bulk Status operation. The status report will be available for at least 2 hours after the operation’s start time.

If a failure occurs while executing on a resource, the failure is logged in the status report and the execution proceeds to next resource.

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