



Reviewing the API Documentation

- [About the Cisco APIC-EM API Documentation, page 1](#)
- [Testing the Cisco APIC-EM APIs, page 6](#)

About the Cisco APIC-EM API Documentation

Cisco APIC-EM controller provides interactive, northbound Representational State Transfer (REST) API documentation. You can use the REST API documentation to help you integrate the controller with your larger network management system and administer your network.



Note

The controller displays northbound REST API documentation based upon services that have been installed in the service catalog on the root.

To access the northbound REST API documentation, from the **Global** toolbar, click **API**.

**Note**

The REST API documentation is based on Swagger 1.2 specifications.

Figure 1: API Window

Available APIs

- [File](#)
- [Flow Analysis](#)
- [IP Geolocation](#)
- [IP Pool Manager](#)
- [Inventory](#)
- [Network Discovery](#)
- [Network Plug and Play](#)
- [PKI Broker Service](#)
- [Policy Administration](#)
- [Role Based Access Control](#)
- [Scheduler](#)
- [Task](#)
- [Topology](#)

File

APIC-EM Service API based on the Swagger™ 1.2 specification

- [Terms of service](#)
- [Cisco DevNet](#)

file-service : File Service API Show/Hide | List Operations | Expand Operations | Raw

[BASE URL: <https://172.19.172.83/api/v1/api-docs/file-service> , API VERSION: 1.0]

The interactive northbound REST API documentation provides:

- Links to information about the northbound REST APIs terms of services and the Cisco developer community website:
 - **Terms of Service**—Review the terms and services for accessing the server where the APIs are located.
 - **Cisco DevNet**—Access the Cisco developer community website. This website offers developer information, community forums, a developer sandbox, and other developer aids.
- A list of supported northbound REST APIs used by the controller and organized by application:
 - **File**
 - **IP Pool Manager**
 - **Network Plug and Play Service**
 - **Policy Administration**
 - **Role Based Access Control**
 - **Scheduler**
 - **Task**
 - **Topology**



Note Only applications with an active service running display in the menu list.

- A list of supported methods for each northbound REST API including:
 - **GET**—To retrieve a resource.
 - **POST**—To create a resource.
 - **PUT**—To change the state of a resource or to update it.
 - **DELETE**— To remove or delete a resource.
- Methods of the API:
 - **Show/Hide**—Displays or hides supported methods of the API (GET, POST, PUT, and DELETE).
 - **List Operations**—Displays the supported methods of the API (GET, POST, PUT, and DELETE).
 - **Expand Operations**—Displays an expanded view of the methods of the API including:
 - **Implementation Notes**—Brief descriptions of what the northbound REST API does, including some specific details of the implementation.
 - **Response Class**—Model and Model Schema views, as well as a Response Content Type:
 - **Parameters**—Parameter, Description, Parameter Type, Data Type definitions (string, integer, or model), as well as input fields if required for testing.
 - **Error Status Codes**—HTTP status code and reason definitions.

- **Raw content**—Provides **Raw** content for the external Swagger UI (user provided) to access the northbound REST API. Content is provided in text file format.

To get a better understanding of the northbound REST APIs, you can run sample methods and get resultant outputs. For more information, see [Testing the Cisco APIC-EM APIs](#), on page 6.

Related Topics

[Reviewing and Testing the Cisco APIC-EM APIs](#)

[Common External RESTful Services HTTP Response Codes](#), on page 5

Supported HTTPS Methods and General Structure

The following table describes the supported HTTPS methods and structure for the Cisco APIC-EM.

HTTPS Method Type	Structure
GET	Use the following values with the GET method type: <ul style="list-style-type: none"> • /noun • /noun/count • /noun/{start}/{end} • /noun/{noun-id}
POST	The POST method type returns a 409 response code if posting a duplicated resource, or the following response: <pre>{"response":"id-of-created-resource"}</pre>
PUT	The PUT method type returns the following response: <pre>{"response":"message-about-attributes-that-changed"}</pre>
DELETE	The DELETE method type returns a 404 response code if it fails, or the following response: <pre>{"response":"message-about-deletion"}</pre>

The following are general guidelines for the supported HTTPS methods for the Cisco APIC-EM:

- Each method is used as an operation on a single resource
- Each resource is expressed as a singular noun (for example: network-device, link, interface, host, etc.)
- The resource is referenced in the HTTPS method by an ID number (not by name)
- The maximum number of entries returned by a GET API is 500 (default). No API should provide more than 500 (default) entries in a single call.
- A CRUD (Create, READ, Update, Delete) operation on all resources is not allowed in a single API call.

Common External RESTful Services HTTP Response Codes

External RESTful services return common HTTP response codes as described in the tables below. In addition to the status codes returned in the response header, each response may have additional content (in JSON file format) according to the nature of the request.

Table 1: Success (2xx) Codes

Status Code	Description
200 OK	The request was successful. The result is contained in the response body.
201 Created	The POST/PUT request was fulfilled and a new resource has been created. Information about the resource is in the response body.
202 Accepted	The request was accepted for processing, but the processing has not been completed.
204 No Content	The request was successful, however no content was returned.
206 Partial Content	The GET request included a Range Header, and the server responded with the partial content matching the range.

Table 2: Client Error (4xx) Codes

Status Code	Description
400 Bad Request	The client made a request that the server could not understand (for example, the request syntax is incorrect).
401 Unauthorized	The client's authentication credentials included with the request are missing or invalid.
403 Forbidden	The server recognizes the authentication credentials, but the client is not authorized to perform this request.
404 Not Found	The client made a request for a resource that does not exist.
409 Conflict	The target resource is in a conflicted state (for example, an edit conflict where a resource is being edited by multiple users). Retrying the request later might succeed.
415 Unsupported Media Type	The client sent a request body in a format that the server does not support (for example, XML to a server that only accepts JSON).

Table 3: Server Error (5xx) Codes

Status Code	Description
500 Internal Server Error	The server could not fulfill the request.
501 Not Implemented	The server has not implemented the functionality required to fulfill the request.
503 Service Unavailable	The server is (temporarily) unavailable.

Related Topics

[Reviewing and Testing the Cisco APIC-EM APIs](#)

[About the Cisco APIC-EM API Documentation, on page 1](#)

Testing the Cisco APIC-EM APIs

Before You Begin

You can test the Cisco APIC-EM northbound REST APIs in the **API** window.

-
- Step 1** From the **Global** toolbar, click **API**.
- Step 2** From the list of available APIs, choose an API.
For example, choose the **Role Based Access Control** API.
- Step 3** From the list of Role Based Access Control APIs, choose an API to view its supported methods.

For example, choose the **user** API.

Figure 2: Supported Methods for User Management API

APIC - Enterprise Module
API 1 🔔
admin ⚙️

Role Based Access Control

APIC-EM Service API based on the Swagger™ 1.2 specification

[Terms of service](#)
[Cisco DevNet](#)

role : Role Description API Show/Hide | List Operations | Expand Operations | Raw

ticket : Ticket Management API Show/Hide | List Operations | Expand Operations | Raw

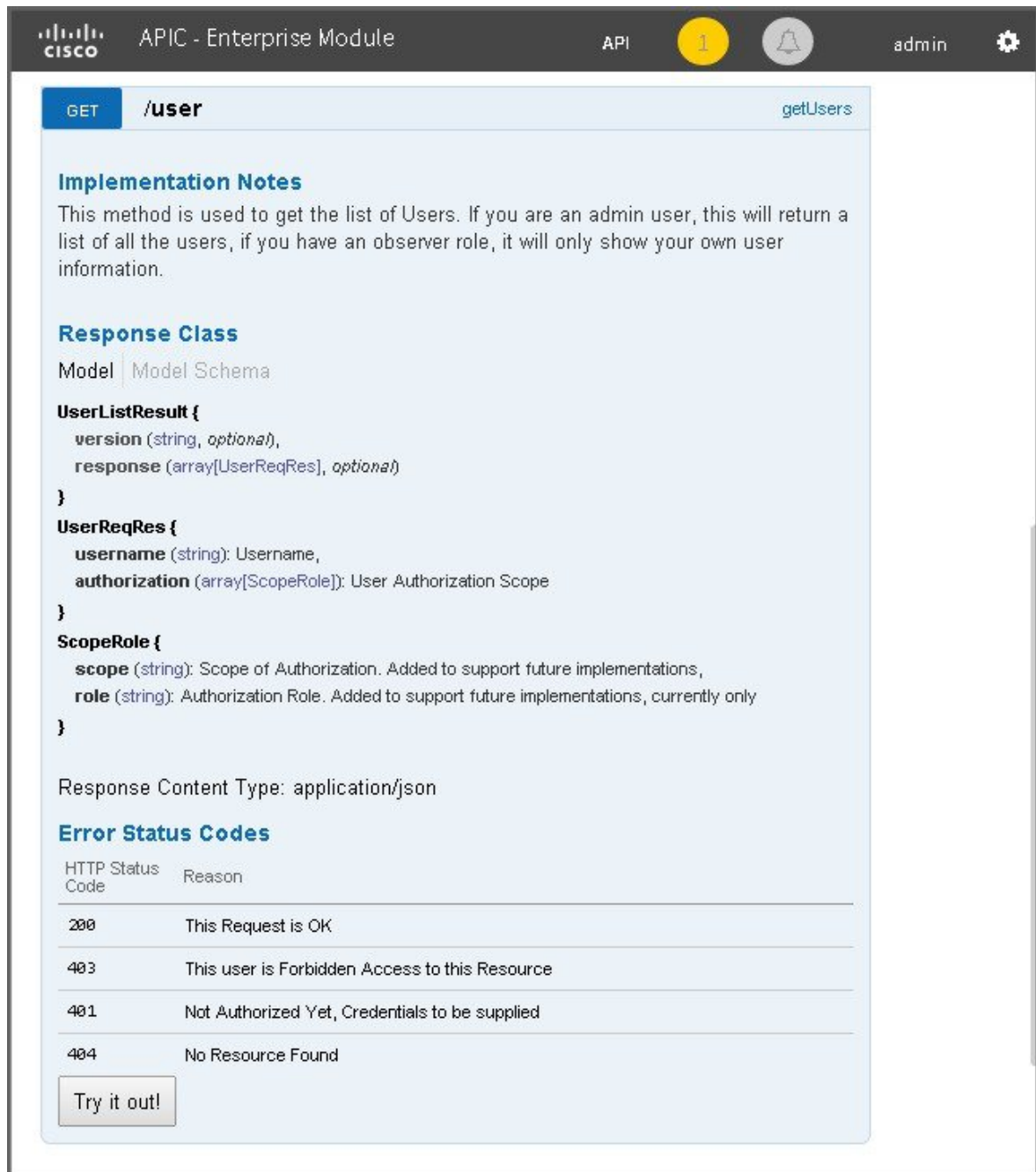
user : User Management API Show/Hide | List Operations | Expand Operations | Raw

	Method	Operation
POST	/user	addUser
GET	/user	getUsers
PUT	/user	updateUser
GET	/user/passphrase/auto	getAutoPassphrase
GET	/user/passphrase/auto/{seedPhrase}	getAutoPassphrase
GET	/user/password-policy/invalid-attempt-count	getAttemptCount
PUT	/user/password-policy/invalid-attempt-count	updateAttemptCount
PUT	/user/password-policy/lock-expiry-time	updateLockExpiryTimeout
GET	/user/password-policy/lock-expiry-time	getLockExpiry
PUT	/user/status	updateUserStatus
GET	/user/status/{username}	getUserStatus
GET	/user/{username}	getUser
DELETE	/user/{username}	deleteUser

407041

Step 4 Click **Expand Operations**.

Figure 3: Expanded View of getUsers API



Step 5 Test the API by clicking the **Try it out!** button located at the bottom of each expanded API method window.

Note Enter content into any of the required content fields prior to testing the API.

For example, click the **Try it out!** button for **GET /user** and review the following output:

- **Request URL**—Displays the request URL created and sent to the controller for the appropriate method (GET, POST, PUT, DELETE)
- **Response Body**—Displays an example of a response to the request URL.
- **Response Code**—Displays the error status code for example response.

- **Response Headers**— Displays the responses returned by the RESTful Services; the specific HTTP headers used are displayed.

Figure 4: Output the getUsers API



