

Demonstrate Navigation through Secure Firewall's API-Explorer

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Review Navigation through FMC API Explorer](#)

[Review Navigation through FDM API Explorer](#)

[Troubleshoot](#)

Introduction

This document describes the navigation through Application Programming Interface (API) explorer of Cisco FMC and Cisco FDM.

Prerequisites

Basic understanding of REST API.

Requirements

It is required for this demonstration to have access to the Firepower Management Center (FMC) GUI with at least one device managed by this Firepower Management Center (FMC). For the FDM part of this demonstration, it is needed to have a Firepower Threat Defense (FTD) managed locally to have access to the FDM GUI.

Components Used

- FMCv
- FTDv
- FTDv Locally managed

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Review Navigation through FMC API Explorer

To access the FMC API explorer, navigate to the next URL:

https://<FMC_mgmt_IP>/api/api-explorer

You must log in with the same credentials used for the FMC GUI. These credentials are entered in a window similar to the next one when you enter the API explorer URLs.

Sign in to access this site

Authorization required by <https://10.88.243.36:43162>
Your connection to this site is not secure

Username

Password

Sign in

Cancel

Once logged in, it is seen that the API queries are divided by categories corresponding to the possible calls you can make using APIs.



Note: Not all configuration functions available from the GUI or CLI are available through the APIs.

[Download OAS 2.0 Spec](#)
[Download OAS 3.0 Spec](#)
[Logout](#)

Cisco Firewall Management Center Open API Specification 1.0.0 OAS3

/fmc_oas3.json

Specifies the REST URLs and methods supported in the Cisco Firewall Management Center API. Refer to the version specific [REST API Quick Start Guide](#) for additional information.

[Cisco Technical Assistance Center \(TAC\) - Website](#)
[Send email to Cisco Technical Assistance Center \(TAC\)](#)
[Cisco Firewall Management Center Licensing](#)

Domains:

- Troubleshoot >
- Backup >
- Network Map >
- Devices >
- Policy Assignments >
- Device HA Pairs >
- Health >

When clicking a category, it expands showing you the different calls available for this category. These calls are shown along with their respective REST methods and the Universal Resource Identifier (URI) of that call.

- Integration >
- Device Groups >
- Status >
- Device Clusters >
- System Information >
- Object >
- Policy** ▾

GET	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies/{objectId}
PUT	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies/{objectId}
DELETE	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies/{objectId}
GET	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies
POST	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies
GET	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies/{containerUUID}/accessrules/{objectId}

In the next example, you make a request to see the access policies configured in the FMC. You click the corresponding method to expand it, then click the **Try it out button**.

Something to emphasize is that you can parametrize your queries with the available parameters in each API call. Only those with red asterisks are mandatory, the others can be left empty.

GET /api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies

Retrieves, deletes, creates, or modifies the access control policy associated with the specified ID. Also, retrieves list of all access control policies.

Parameters Try it out

Name	Description
name string <small>(query)</small>	If parameter is specified, only the policy matching with the specified name will be displayed. Cannot be used if object ID is specified in path. <input type="text" value="name - If parameter is specified, only the poli"/>
filter string <small>(query)</small>	Value is of format (including quotes): "locked:{true false}" locked query parameter when set to 'true' returns list of Access Policies which are locked and when set to 'false' returns policies which are unlocked. <input type="text" value="filter - Value is of format (including quotes): <"/>
offset integer(\$int32) <small>(query)</small>	Index of first item to return. <input type="text" value="offset - Index of first item to return."/>
limit integer(\$int32) <small>(query)</small>	Number of items to return. <input type="text" value="limit - Number of items to return."/>
expanded boolean <small>(query)</small>	If set to true, the GET response displays a list of objects with additional attributes. <input type="text" value="--"/>

For example, the domainUUID is mandatory for all API calls, but on the API Explorer this fills automatically.

Next step is to click **Execute** to make this call.

name string <small>(query)</small>	If parameter is specified, only the policy matching with the specified name will be displayed. Cannot be used if object ID is specified in path. <input type="text" value="name - If parameter is specified, only the poli"/>
filter string <small>(query)</small>	Value is of format (including quotes): "locked:{true false}" locked query parameter when set to 'true' returns list of Access Policies which are locked and when set to 'false' returns policies which are unlocked. <input type="text" value="filter - Value is of format (including quotes): <"/>
offset integer(\$int32) <small>(query)</small>	Index of first item to return. <input type="text" value="offset - Index of first item to return."/>
limit integer(\$int32) <small>(query)</small>	Number of items to return. <input type="text" value="limit - Number of items to return."/>
expanded boolean <small>(query)</small>	If set to true, the GET response displays a list of objects with additional attributes. <input type="text" value="--"/>
domainUUID * required string <small>(path)</small>	Domain UUID <input type="text" value="e276abec-e0f2-11e3-8169-6d9ed49b625f"/>

Execute

Before clicking Execute, you can see examples of responses to the calls to get an idea of the possible responses you can get depending on whether the request is correct or not.

Execute

Responses

Code	Description	Links
200	OK	No links

Media type: Examples: Example 1: GET /fmc_config/v1/domain/DomainUUID/policy/accesspolicies (Test GET ALL Success of Acc

Controls Accept header.

Example Value | Schema

```

{
  "links": "/fmc_config/v1/domain/DomainUUID/policy/accesspolicies?offset=0&limit=2",
  "items": [
    {
      "type": "AccessPolicy",
      "name": "AccessPolicy1_updated",
      "description": "policy to test FMC implementation",
      "defaultAction": {
        "id": "id_of_default_action",
        "type": "AccessPolicyDefaultAction"
      }
    },
    {
      "type": "AccessPolicy",
      "name": "AccessPolicy2_updated",
      "description": "policy to test FMC implementation",
      "defaultAction": {
        "id": "id_of_default_action",
        "type": "AccessPolicyDefaultAction"
      }
    }
  ]
}

```

Once the API call is executed, you obtain, together with the response payload, the response code. In this case 200, which corresponds to an OK request. You also get the cURL and the URL of the call you just made. This information is useful if you want to make this call with an external client/software.

The answer obtained returns the ACPs configured in the FMC along with their objectID. In this case, you can see this information in the red box in the next image:

Execute Clear

Responses

Curl

```

curl -X 'GET' \
  'https://10.88.243.36:43162/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/policy/accesspolicies' \
  -H 'accept: application/json' \
  -H 'X-auth-access-token: d1594a50-3f98-4519-875b-50c70b454552'

```

Request URL

```

https://10.88.243.36:43162/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/policy/accesspolicies

```

Server response

Code	Details
200	<p>Response body</p> <pre> { "links": { "self": "https://10.88.243.36:43162/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/policy/accesspolicies?offset=0&limit=25" }, "items": [{ "type": "AccessPolicy", "links": { "self": "https://10.88.243.36:43162/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/policy/accesspolicies/00505683-186A-0ed3-0000-004294967299" }, "name": "ACP_cchanes", "id": "00505683-186A-0ed3-0000-004294967299" }], "paging": { "offset": 0, "limit": 25, "count": 1, "pages": 1 } } </pre>

Download

This objectID is the value you enter in calls that require reference to this ACP. For example, to create a rule within this ACP.

The URIs that contain values between curly brackets {} are values required to make this call. Remember that domainUUID is the only value that is automatically filled.

GET	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies/{containerUUID}/accessrules/{objectId}
PUT	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies/{containerUUID}/accessrules/{objectId}
DELETE	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies/{containerUUID}/accessrules/{objectId}
GET	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies/{containerUUID}/accessrules
PUT	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies/{containerUUID}/accessrules
POST	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies/{containerUUID}/accessrules
DELETE	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies/{containerUUID}/accessrules
GET	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies/{containerUUID}/defaultactions/{objectId}
PUT	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies/{containerUUID}/defaultactions/{objectId}
GET	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies/{containerUUID}/loggingsettings/{objectId}
PUT	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies/{containerUUID}/loggingsettings/{objectId}
GET	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies/{containerUUID}/operational/hitcounts
PUT	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies/{containerUUID}/operational/hitcounts
DELETE	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies/{containerUUID}/operational/hitcounts

The values required for these calls are specified in the call description. To create rules for an ACP, you require the policyID, as you can see in the next image:

POST	/api/fmc_config/v1/domain/{domainUUID}/policy/accesspolicies/{containerUUID}/accessrules
Retrieves, deletes, creates, or modifies the access control rule associated with the specified policy ID and rule ID. If no ID is specified, retrieves list of all access rules associated with the specified policy ID. Check the response section for applicable examples (if any).	

This policyID is entered in the field specified as containerUUID, another required field for POST methods is the payload or request body. You can use the examples given to modify to your needs.

containerUUID required
 string
 (path)
 The container id under which this specific resource is contained.

domainUUID required
 string
 (path)
 Domain UUID

Request body required application/json

The input access control rule model.

Examples:
Example 1 : POST /fmc_config/v1/domain/DomainUUID/policy/accesspolicies/containerUUID/accessrules (Test POST of Access rule)

```
{
  "action": "ALLOW",
  "enabled": true,
  "type": "AccessRule",
  "name": "Rule1",
  "sendEventsToFMC": false,
  "logFiles": false,
  "logBegin": false,
  "logEnd": false,
  "variableSet": {
    "name": "Default Set",
    "id": "VariableSetUUID",
    "type": "VariableSet"
  },
  "vlanTags": {
    "objects": [
      {
        "type": "VlanTag",

```

Example of modified payload:

```
{ "action": "ALLOW", "enabled": true, "type": "AccessRule", "name": "Testing API rule", "sendEventsToFMC": false, "logFiles": false,
"logBegin": false, "logEnd": false, "sourceZones": { "objects": [ { "name": "Inside_Zone", "id": "8c1c58ec-8d40-11ed-b39b-f2bc2b448f0d",
"type": "SecurityZone" } ] }, "destinationZones": { "objects": [ { "name": "Outside_Zone", "id": "c5e0a920-8d40-11ed-994a-900c72fc7112",
"type": "SecurityZone" } ] }, "newComments": [ "comment1", "comment2" ] }
```




Note: The available zones, together with their IDs, can be obtained using the next query.

GET

`/api/fmc_config/v1/domain/{domainUUID}/object/securityzones`

Once you execute the previous call, you get a 201 response code, indicating that the request has succeeded and has led to the creation of the resource.

```
Server response
Code    Details
201    Response body
{
  "metadata": {
    "ruleIndex": 6,
    "section": "Default",
    "category": "--Undefined--",
    "accessPolicy": {
      "name": "ACP_cchanes",
      "id": "005056B3-1B6A-0ed3-0000-004294967299",
      "type": "AccessPolicy"
    }
  },
  "links": {
    "self": "https://10.88.243.36:43162/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/policy/accesspolicies/005056B3-1B6A-0ed3-0000-000268435456"
  },
  "enabled": true,
  "action": "ALLOW",
  "name": "Testing API rule",
  "type": "AccessRule",
  "id": "005056B3-1B6A-0ed3-0000-000268435456",
  "variableSet": {
    "name": "Default Set",
    "id": "76fa83ea-c972-11e2-8be8-8e45bb1343c0",
    "type": "VariableSet"
  },
  "sourceZones": {
    "objects": [

```

Finally, you must make a deployment for these changes to take effect in the FTD whose ACP was modified. For this, you have to obtain the list of devices that have changes ready to be deployed.

GET /api/fmc_config/v1/domain/{domainUUID}/deployment/deployabledevices

Retrieves list of all devices with configuration changes, ready to be deployed.

The example contains a pair of devices configured in High Availability. You must obtain the ID of this HA, in case of being a standalone device, you must obtain the ID of that device.

```
Responses
Curl
curl -X 'GET' \
  'https://10.88.243.36:43162/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/deployment/deployabledevices' \
  -H 'accept: application/json' \
  -H 'X-auth-access-token: 41f2e4aa-c681-4064-8cdc-6f734785dba9'
Request URL
https://10.88.243.36:43162/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/deployment/deployabledevices
Server response
Code    Details
200    Response body
{
  "links": {
    "self": "https://10.88.243.36:43162/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/deployment/deployabledevices?offset=0&limit=25"
  },
  "items": [
    {
      "version": "1689794173607",
      "name": "HA_FTD72",
      "type": "DeployableDevice"
    }
  ],
  "paging": {
    "offset": 0,
    "limit": 25,
    "count": 1,
    "pages": 1
  }
}
```

The query needed to obtain the device ID of the HA is as follows:

GET /api/fmc_config/v1/domain/{domainUUID}/devicepairs/ftddevicepairs

Retrieves or modifies the Firewall Threat Defense HA record associated with the specified ID. Creates or breaks or deletes a Firewall Threat Defense HA pair. If no ID is specified for a GET, retrieves list of all Firewall Threat Defense HA pairs.

With the device ID and the deployment version number, you can modify the payload of the next call example to make the call to perform this deployment.

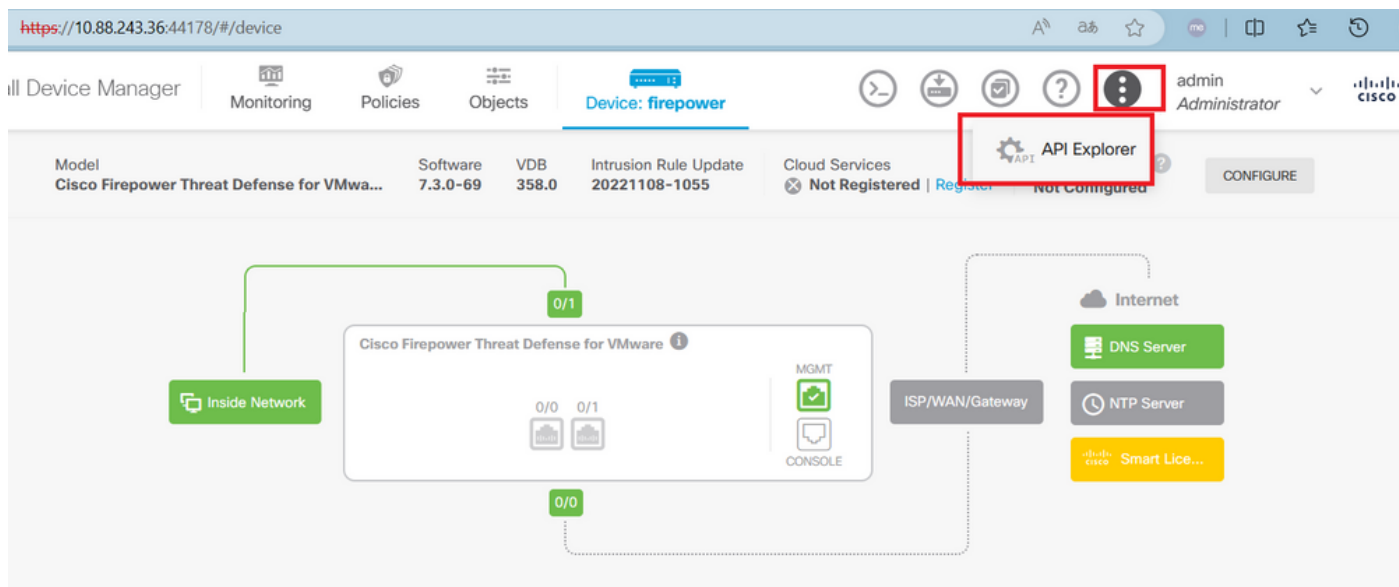
POST /api/fmc_config/v1/domain/{domainUUID}/deployment/deploymentrequests

Creates a request for deploying configuration changes to devices. *Check the response section for applicable examples (if any).*

Once this call is executed, if everything is correct, you get a response with code 202.

Review Navigation through FDM API Explorer

To access the FDM API Explorer, it is possible to use a button on the FDM GUI to go directly to it, as shown in the next image:



Once in the API Explorer, you notice that the queries are also divided into categories.

The following is a list of resources you can use for programmatic access to the device using the Secure Firewall Threat Defense REST API. The resources are organized into groups of related resources. Click a group name to see the available methods and resources. Click a method/resource within a group to see detailed information. Within a method/resource, click the **Model** link under **Response Class** to see documentation for the resource.

You can test the various methods and resources through this page. When you fill in parameters and click the **Try it Out!** button, you interact directly with the system. GET calls retrieve real information. POST calls create real objects. PUT calls modify existing objects. DELETE calls remove real objects. However, most changes do not become active until you deploy them using the POST /operational/deploy resource in the Deployment group. Although some changes, such as to the management IP address and other system-level changes, do not require deployment, it is safer to do a deployment after you make any configuration changes.

The REST API uses OAuth 2.0 to validate access. Use the resources under the Token group to get a password-granted or custom access token, to refresh a token, or to revoke a token. You must include a valid access token in the Authorization: Bearer header on any HTTPS request from your API client.

Before using the REST API, you need to finish the device initial setup. You can complete the device initial setup either through UI or through InitialProvision API.

You can also refer to [this](#) page for a list of API custom error codes. (Additional errors might exist.)

NOTE: The purpose of the API Explorer is to help you learn the API. Testing calls through the API Explorer requires the creation of access locks that might interfere with regular operation. We recommend that you use the API Explorer on a non-production device.

Cisco makes no guarantee that the API version included on this Firepower Threat Device (the "API") will be compatible with future releases. Cisco, at any time in its sole discretion, may modify, enhance or otherwise improve the API based on user feedback.

AAASetting	Show/Hide	List Operations	Expand Operations
ASPathList	Show/Hide	List Operations	Expand Operations
AccessPolicy	Show/Hide	List Operations	Expand Operations

To expand a category, you must click it, and then you can expand each of the operations by clicking on any of them. The first thing found inside each operation is an example of an OK response for this call.

AccessPolicy Show/Hide List Operations Expand Operations

- GET /policy/accesspolicies/{parentId}/accessrules
- POST /policy/accesspolicies/{parentId}/accessrules
- DELETE /policy/accesspolicies/{parentId}/accessrules/{objId}
- GET /policy/accesspolicies/{parentId}/accessrules/{objId}
- PUT /policy/accesspolicies/{parentId}/accessrules/{objId}
- GET /policy/accesspolicies

Response Class (Status 200)

Model	Example Value
	<pre>{ "items": [{ "version": "string", "name": "string", "defaultAction": { "action": "PERMIT", "eventLogAction": "LOG_FLOW_START", "intrusionPolicy": { "id": "string", "name": "string" } } }] }</pre>

The next thing you see are the parameters available to constrain the responses of the call you make. Remember that only the fields marked as required are mandatory to make such a call.

Response Content Type

Parameters

Parameter	Value	Description	Parameter Type	Data Type
offset	<input type="text"/>	An integer representing the index of the first requested object. Index starts from 0. If not specified, the returned objects will start from index 0	query	integer
limit	<input type="text"/>	An integer representing the maximum amount of objects to return. If not specified, the maximum amount is 10	query	integer
sort	<input type="text"/>	The field used to sort the requested object list	query	string
filter	<input type="text"/>	The criteria used to filter the models you are requesting. It should have the following format: {key}{operator}{value}; {key}{operator}{value}. Supported operators are: "!=" (not equals), "=" (equals), "~" (similar). Supported keys are: "name", "fts". The "fts" filter cannot be used with other filters.	query	string

Finally, you find the possible response codes that this call can return.

Response Messages

HTTP Status Code	Reason	Response Model	Headers				
401		<table border="1"><thead><tr><th>Model</th><th>Example Value</th></tr></thead><tbody><tr><td></td><td><pre>{ "status_code": 0, "message": "string", "internal_error_code": 0 }</pre></td></tr></tbody></table>	Model	Example Value		<pre>{ "status_code": 0, "message": "string", "internal_error_code": 0 }</pre>	
Model	Example Value						
	<pre>{ "status_code": 0, "message": "string", "internal_error_code": 0 }</pre>						
403		<table border="1"><thead><tr><th>Model</th><th>Example Value</th></tr></thead><tbody><tr><td></td><td><pre>{ "status_code": 0, "message": "string", "internal_error_code": 0 }</pre></td></tr></tbody></table>	Model	Example Value		<pre>{ "status_code": 0, "message": "string", "internal_error_code": 0 }</pre>	
Model	Example Value						
	<pre>{ "status_code": 0, "message": "string", "internal_error_code": 0 }</pre>						

If you want to make this call, you must click **Try It Out**. To find this button, you have to scroll down until you find this button since it is located at the bottom of each call.

520

Model	Example Value
	<pre>{ "status_code": 0, "message": "string", "internal_error_code": 0 }</pre>

TRY IT OUT

When you click the Try It Out button, if it is a call that does not require more fields, it executes immediately and gives you the response.

TRY IT OUT Hide Response

Curl

```
curl -X GET --header 'Accept: application/json' 'https://10.88.243.36:44178/api/fdm/v6/policy/accesspolicies'
```

Request URL

```
https://10.88.243.36:44178/api/fdm/v6/policy/accesspolicies
```

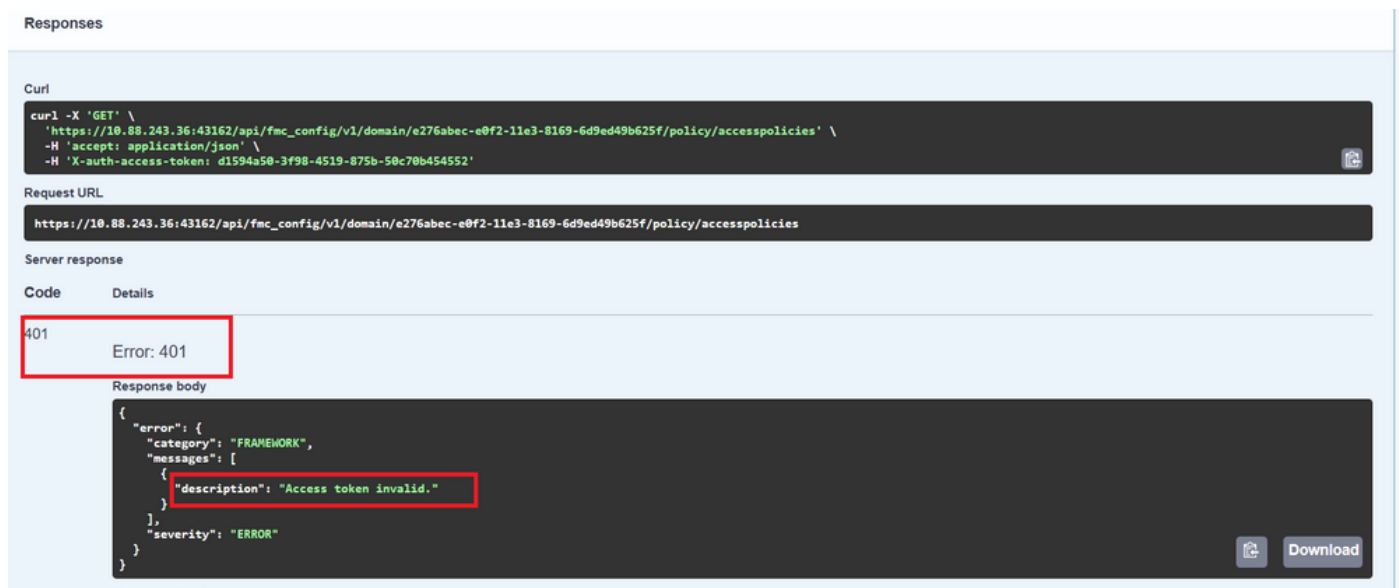
Response Body

```
{
  "items": [
    {
      "version": "ka4esjod4iebr",
      "name": "NGFW-Access-Policy",
      "defaultAction": {
        "action": "DENY",
        "eventLogAction": "LOG_NONE",
        "intrusionPolicy": null,
        "syslogServer": null,
        "hitCount": {
          "hitCount": 0,
          "firstHitTimeStamp": "",
          "lastHitTimeStamp": "",
          "lastFetchTimeStamp": ""
        }
      }
    }
  ]
}
```

Troubleshoot

Each call generates an HTTP response code and response body. This helps you to identify where the error is.

The next is a common error that occurs when the session has expired, indicating that the token is invalid because it has expired.



The screenshot displays a REST client interface with the following sections:

- Responses**: The main header of the interface.
- Curl**: A terminal window showing the command: `curl -X 'GET' \ 'https://10.88.243.36:43162/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/policy/accesspolicies' \ -H 'accept: application/json' \ -H 'X-auth-access-token: d1594a50-3f98-4519-875b-50c70b454552'`
- Request URL**: `https://10.88.243.36:43162/api/fmc_config/v1/domain/e276abec-e0f2-11e3-8169-6d9ed49b625f/policy/accesspolicies`
- Server response**: A table with two columns: **Code** and **Details**. The first row shows a **401** status code and the text **Error: 401**.
- Response body**: A JSON object: `{ "error": { "category": "FRAMEWORK", "messages": [{ "description": "Access token invalid." }] }, "severity": "ERROR" }`

The next are examples of HTTP response codes that calls can return:

- **2xx series**: Success. There are several status codes: 200 (GET and PUT), 201 (POST), 202, 204 (DELETE). They indicate a successful API call.
- **30x series**: Redirection. Can be used when a client originally used HTTP and was redirected to HTTPS.
- **4xx series**: Client-side failure in the API call that was sent from the client to the server. Two examples include a 401 status code, indicating the session is not authenticated, and a 403 code, indicating a forbidden access attempt.
- **5xx series**: Server, device, or service-side failure. This could be the result of the device API service being disabled, or inaccessible over the IP network