NetIQ Secure API Manager 2.1

API Help

June 2022

Legal Notice

The only warranties for products and services of Micro Focus and its affiliates and licensors ("Micro Focus") are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

For information about legal notices, trademarks, disclaimers, warranties, export and other use restrictions, U.S. Government rights, patent policy, and FIPS compliance, see Micro Focus Legal Information (https://www.microfocus.com/legal).

Contents

About This Book

The *NetIQ Secure API Manager Help* provides conceptual information and step-by-step guidance for building your API library and managing it.

Intended Audience

This guide provides information for individuals responsible for creating, maintaining, and using APIs. You must be familiar with REST, APIs, Swagger, coding, SOAP, and WebSockets and we assume that you know and understand these concepts. This guide does not contain detailed information about these technologies.

Additional Documentation

For the most recent version of this guide and other Secure API Manager documentation resources, visit the Secure API Manager Documentation website (https://www.microfocus.com/documentation/secure-api-manager/).

Contact Information

We want to hear your comments and suggestions about this book and the other documentation included with this product. You can use the **comment on this topic** link at the bottom of each page of the online documentation, or send an email to Documentation-Feedback@netiq.com.

For specific product issues, contact Micro Focus Customer Care at https://www.microfocus.com/support-and-services/.

1 Welcome to Secure API Manager

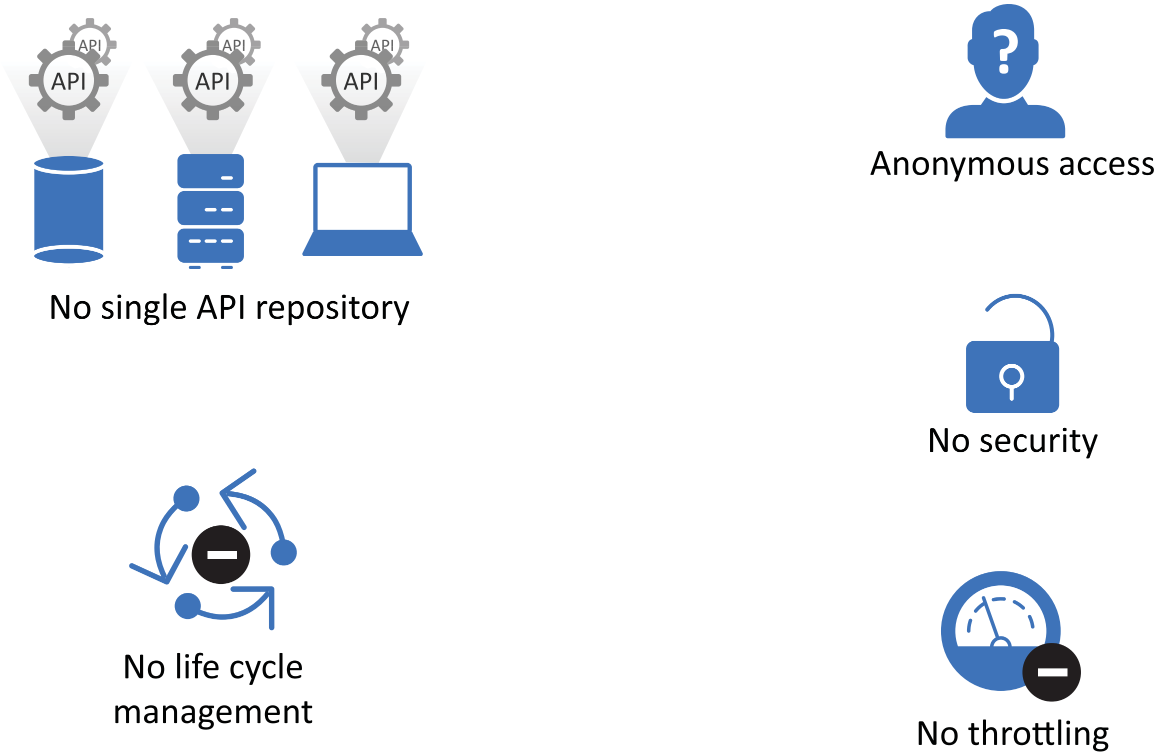
Application programming interfaces (APIs) are sets of definitions, protocols, and tools for building software. There is a lot of software and many items that make up the **Internet of Things (IoT)** use APIs to provide functionality that your business requires. The APIs also provide the ability to customize the software to solve your business problems.

Secure API Manager provides a central location to add, manage, and secure the APIs that your company uses. You add the APIs once to Secure API Manager and they are available for reuse. You can see all of the available APIs in a single location, making it easy for you to combine multiple APIs to create new functionality.

How Secure API Manager Solves API Management Issues

As you add APIs to your IT infrastructure, you might run into several management issues as depicted in the following graphic.

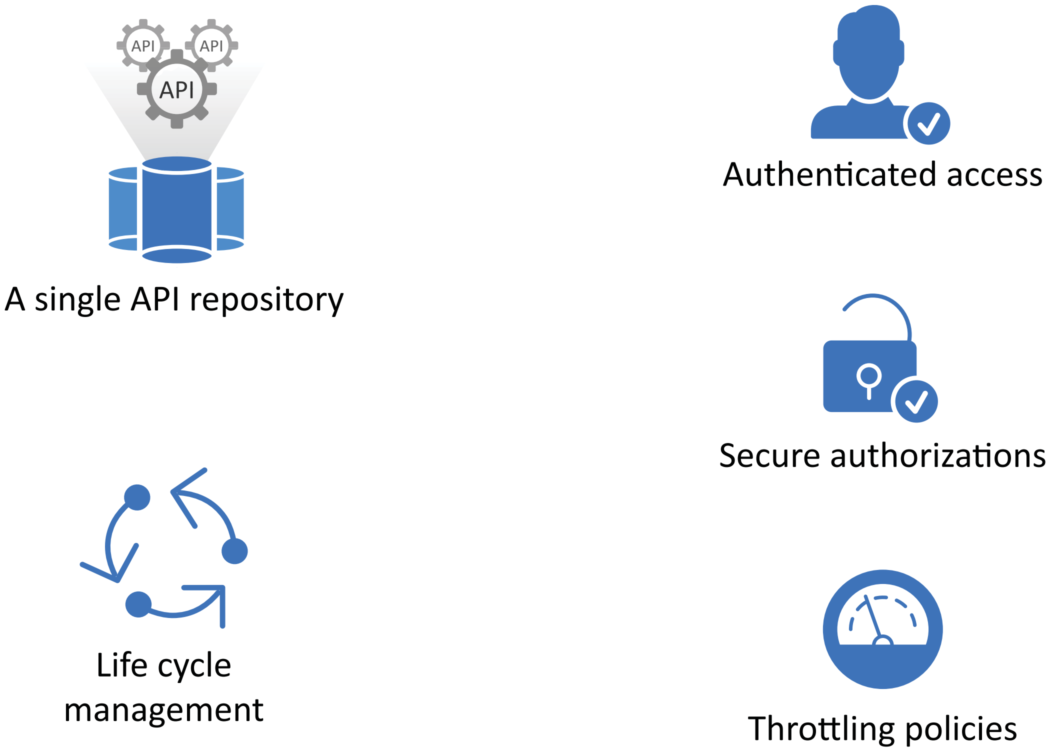
Figure 1-1 *Issues Managing APIs*



- **No single API repository:** Not having a single API repository can cause duplicate work, can cause APIs to be lost if a hard drive fails, or can take a lot of time for each person who wants to use an API having to find the API.
- **No life cycle management:** Not having a life cycle management system for the APIs can result in APIs with security issues being used. It can cause a lot of confusion if there are multiple administrators and they do not know which APIs to use.
- **Anonymous access:** If you do not have a system to manage your APIs, you do not have a way to show who had access to which API for auditing purposes. As security breaches continue, it is important to ensure that access to create and modify the APIs is controlled and that there can be an audit trail of who created or modified the APIs.
- **No security:** If you do not have an API management system, you do not have a way to provide an audit trail for who accessed which API. It also means that you must manually add any authorization information in the Header of each API. This takes a lot of time and the authorization information might not meet your organization's security policies.
- **No throttling:** Throttling provides the ability to control the throughput to the API. If you cannot limit the throughput to the API, it can cause performance issues with the system that is hosting the API.

NetIQ solves these issues by providing a system that allows you to manage, create, and control the APIs used in your environment through Secure API Manager. The following graphic depicts the management solutions that Secure API Manager provides.

Figure 1-2 *Secure API Manager API Management Solutions*



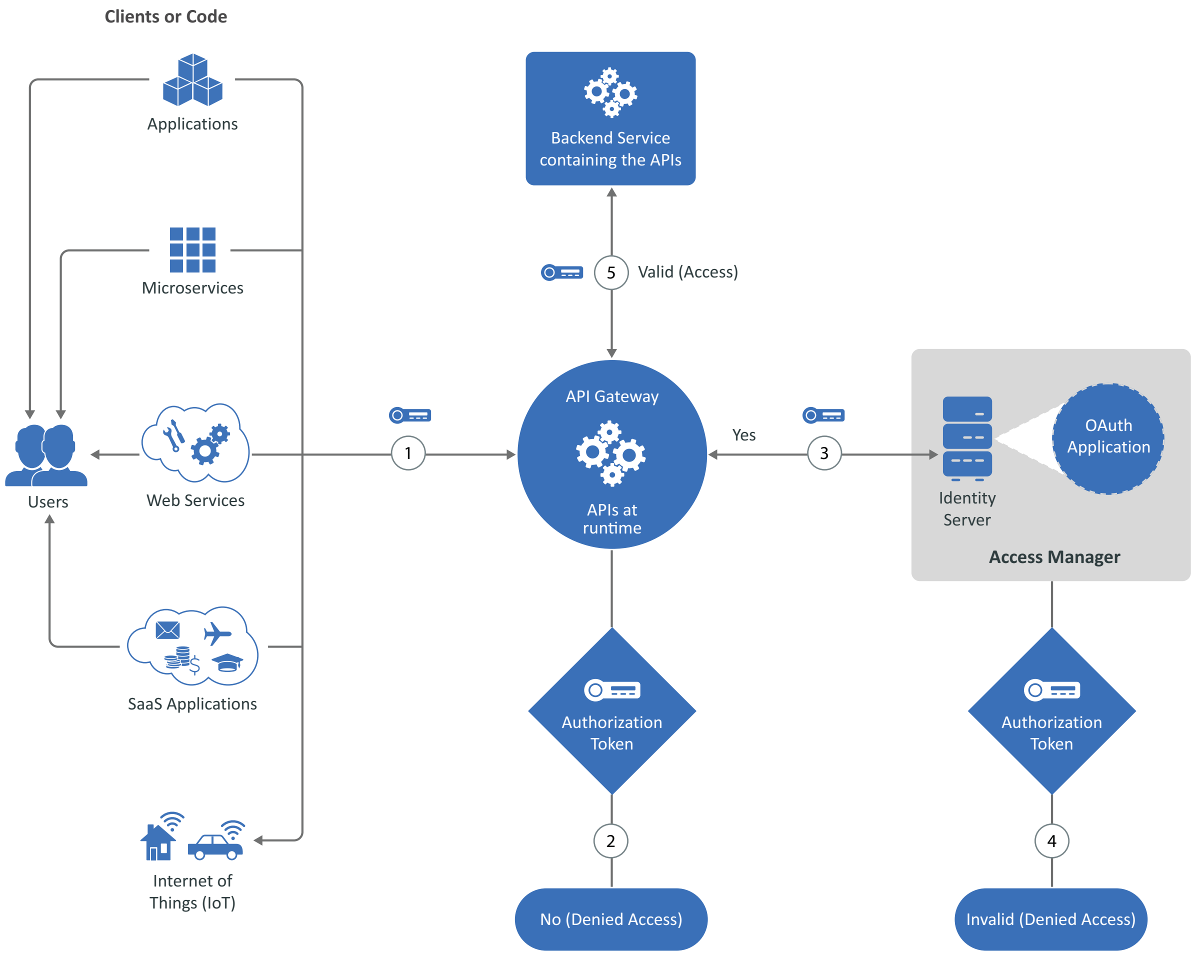
- **Single API repository:** Secure API Manager provides a single repository, with fault tolerance, where you can store your APIs securely. You do not have to worry about hard drive failures on your laptop. It also allows you to have internal and external people access and use the APIs without emailing a copy of the API to them.
- **Life cycle management:** Secure API Manager automatically versions the APIs and allows you to deprecate the APIs that are no longer in use. By deprecating the APIs, you can keep a historical record of how the APIs have evolved and changed.
- **Authenticated access:** Secure API Manager uses the roles and scopes to control who has access to the APIs. When you create an API group, you assign the Access Manager roles and scopes to any APIs that you subscribe to in this group. This provides authenticated access to the APIs and controls who can consume the APIs.
- **Secure authorizations:** Secure API Manager secures authorizations to the APIs through Access Manager OAuth clients.
- **Throttling policies:** Secure API Manager provides subscription tiers that you select when you create an API. The subscription tiers limit the number of authorizations to the API.

Understanding API Authorizations

Understanding how Secure API Manager authorizes access to APIs helps you understand why you are required to add specific information when you create APIs. It also helps you understand the calls you must add to the applications and services that use the APIs stored in Secure API Manager.

Secure API Manager controls access to APIs through OAuth authorizations. When you configure Secure API Manager, it automatically creates an OAuth 2 application for you in Access Manager. Secure API Manager uses the authorizations tokens from this OAuth 2 application to secure access to the APIs. When an API developer creates an API in the Publisher, the developer adds the authorization token to the API from this OAuth 2 application. The following graphic shows the flow of the API authorization from the application, service, or item through the API Gateway to the Access Manager Identity Server.

Figure 1-3 *How Secure API Manager Authorizes APIs*



1. When an application, service, or item calls an API, the call goes to the API Gateway. The API Gateway contains the APIs in a run-time environment.
2. The API Gateway checks to see if the call for the API contains an OAuth token. If it does not, the API Gateway rejects the call and the application, service, or item receives a message stating the API is not available.

3. If the call for the API does contain an OAuth token, the API Gateway sends the call to the Identity Server.
4. The Identity Server checks the OAuth application to see if the token is valid. If the token is not valid, the Identity Server sends that information to the API Gateway and the API Gateway rejects the call. The application, service, or item receives a message stating that the API is not available.
5. If the token is valid, the Identity Server sends that information to the API Gateway. The API Gateway then allows the call for the API to go to the backend service. The backend service knows that the call is valid and provides the additional functionality of the application, service, or item in the API call.

Required Knowledge

To work with APIs you must have a basic understanding of APIs, REST, SOAP, OAuth, Swagger, and WebSocket. You must also be able to read computer code. This guide is not a primer for these topics. The procedures and processes assume that you understand these concepts. There are many different sources of information about these topics. Here are a few of the topics you must know:

- APIs
- OAuth 2.0
- REST
- SOAP
- Swagger
- WebSocket

Accessing the Publisher and the Store

The Publisher is where you create, manage, and publish the APIs. The Store is where you subscribe to the published APIs so that you can use the APIs in your applications, services, or items.

The Publisher and the Store are available through the Access Manager user portal. There are two new appmarks on the user portal that allow you to access the Publisher and the Store. The name of the appmarks are:

- **Publisher:** APIs: Create/Publisher
- **Store:** APIs: Subscribe

Your administrator or IT department should provide the URL for you to access the user portal. By default, the URL for the user portal is:

```
https://NAM-Base-URL:8443/nidp/portal
```

Your administrator or IT department should also ensure that you have the proper permissions for you to access these appmarks. By default, no one has the permissions assigned to access the appmarks. The administrator must assign the correct permissions for you to access the Publisher and the Store.

2 Managing the APIs through the Publisher

The **Publisher** is where you create, manage, and publish the APIs for use internally or externally by your partners. The Publisher is a web application that you access through an appmark in the Access Manager user portal. The Publisher appmark is named `APIs: Create/Publish`. Your administrator or IT department must assign you the proper permissions to be able to access this appmark.

- "Create an API" on page 13
- "Manage APIs" on page 20

Create an API

Publisher > New

The Publisher allows you to create REST APIs and manage those APIs. When you create an API, you define the general settings, the backend service, one or more endpoints, and rate-limiting policies for the API.

You must understand REST and know how to create APIs. For more information, see the RFC 7231 Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content (https://tools.ietf.org/html/rfc7231) page.

Secure API Manager stores the APIs on a specific Identity Server cluster. You must select which Identity Server cluster you want to use when you create the APIs. If you have internal and external Identity Server clusters, you can use them to further separate the APIs. Use the following information to help you create an API.

- "Define General Settings" on page 13
- "Define the Backend Service" on page 14
- "Define Endpoints" on page 17
- "Define Rate Limiting Policies" on page 18

Define General Settings

Publisher > New > General

The general settings allow you to give the API a name, and add a description, a version, and tags to help you easily search for the API. You also must define the backend service, define endpoints, and define rate-limiting policies for the API.

To define the general settings for the API:

1. Ensure that you selected the appropriate Identity Server cluster on the main page of the Publisher before proceeding.
2. Click **General**, then define the general settings for the API:

Name

Specify a unique name for the API.

Type

Select the type of API you want to create. With this release, Secure API Manager supports only REST APIs. You cannot change the API type.

Version

Specify the version of the API.

Description

Specify a detailed description of the API so that anyone who uses the API understands what it does. If you clone the API, update the description of the API to describe the differences between the APIs.

API Base URL Path

Specify the base URL path to the endpoint for the API. You can define this path to be whatever you want.

Subscription Tiers

Select one of the available subscription tiers to control the number of authorizations to the API. Secure API Manager administrators create these Limiting Policies during the configuration of the API Gateway. The Limiting Policies protect the API Gateway from receiving too many requests.

Tags

Specify a tag for the API to simplify searching for the API. Secure API Manager includes the specified tags as search parameters for the APIs.

Transports

Select whether the API communicates over **HTTP** or **HTTPS**.

3 Click **OK** to save the API.

or

Click **Backend Service** to define the backend service that the API uses.

If you click **Show JSON**, you can view the API in JSON.

Define the Backend Service

Publisher > New > Backend Service

Secure API Manager provides an automatic validation of SSL for the backend service when you include SSL in the API. It validates the SSL connection in two ways:

- Validating the backend service server's certificate chain to either a well-known trusted root or a configured trusted root.
- Validating that the backend service domain name matches the name in the backend service server's certificate.

By default, Secure API Manager has the backend service SSL certificate validation enabled. You can disable it for testing purposes or in situations where network professionals determine that it is not needed. You can disable the backend service SSL certificate validation process by deselecting **Validate SSL Certificate** when you edit an API. However, we recommend that you always leave it enabled.

Disabling the backend service SSL certificate validation means that you do not have to have a Trusted Root certificate in the backend service configuration.

Enabling the backend service SSL certificate validation means:

- If the backend service server's certificate uses a well-known trusted root certificate, then you do not have to configure a Trusted Root for the Backend Service.
- If the backend service server's certificate does not use a well-known Trusted Root certificate, then you must configure a Trusted Root for the backend service. The Trusted Root must follow these guidelines:
    - Each certificate in the chain:
        - Must be in PEM format
        - Must have no duplicate certificates
        - Must be a well-formed certificate chain
    - Secure API Manager uses the domain names or IP addresses you added when creating or editing an API under **Backend Service > Services** as the value to match against the server certificate's subject name. Secure API Manager uses domain names if one exists; otherwise, it uses an IP address. This means that if you use an IP address in the **Backend Service > Service** definition, the backend service server certificate must include a subject alternate name detailing the IP addresses you added.

To define the backend service:

1. Ensure that you have defined the general settings before defining the backend service.
2. Click **Backend Service**, then define the settings for the backend service of the API:

   Service Name

   Specify the name of the backend service.

   Protocol

   Select whether the service on the backend service communicates over **HTTP** or **HTTPS**.

   IMPORTANT: The protocol that you select is the only protocol allowable for any backend service that you add to this API. You cannot specify an HTTP backend service as well as an HTTPS backend service under the same API. Every backend service that you add to this API must be reachable by the protocol you select in this field.

   URL Context

   Specify the URL context of the service on the backend service.

   Trusted Root

   Paste the trusted root certificate or the chain of certificates in the `.pem` format from the backend service. The process of validating the backend service certificates contains multiple ways of providing this information. Ensure that you understand the process of validating the backend service certificates to have this option work.

3 Click **Add Service** to define the settings for one or more of the backend services that the API accesses:

Host

Specify the fully qualified DNS name of the backend service.

Port

Specify the port that the backend service uses. By default, HTTP uses 80 and HTTPS uses 443.

4 Click **Advanced** to define advanced options for the backend service.

4a Define and validate the SSL handshake configuration using the following information:

Validate SSL Certificate

Select whether you want Secure API Manager to validate the SSL chain to the backend service. We recommend that you leave it enabled for security reasons.

Enable SNI

Enable Server Name Indication (SNI) and specify a host name if the backend service is a virtual host using TLS and the backend service selects the destination based on SNI.

SNI is an extension to the Transport Layer Security (TLS) computer networking protocol by which a client indicates which host name it is attempting to connect to at the start of the handshaking process. SNI allows services to present multiple certificates on the same IP address and TCP port number, thus allowing multiple secure (HTTPS) websites (or any other service over TLS) to be served by the same IP address without requiring all those sites to use the same certificate.

If you did not specify a domain name, Secure API Manager uses the host name from the backend service definition. Secure API Manager always sends the host header with the upstream domain name in the proxied requests.

Host Name to validate SSL/SNI

Specify the host name to use to validate the SSL certificate and the SNI configuration. If you do not specify a name, Secure API Manager uses the host name of the backend service you specified in Step 3.

4b Define the API request to the backend service configuration using the following information:

Pass Authorization Header to Backend Service

Select this option to pass authorization headers to the backend service through the API calls.

HEADERS

Secure API Manager added the name of the backend service you defined in Step 3. You can add additional headers, edit the headers, and delete the headers. If you edit and change the value of the host header, Secure API Manager updates the value you defined in Step 3 with this new value.

4c Click **OK** to save these changes.

5 Click **OK** to save the API.

or

Click **Endpoints** to define the endpoints for the API.

6 (Optional) click **Show JSON** to view the API in JSON.

Define Endpoints

Publisher > New > Endpoints

Secure API Manager allows you to define as many endpoints for an API as you need. It also provides the ability to add parameters to each endpoint. You use the endpoints to help with the communication between client APIs and server APIs. You must know the endpoints for your API to be able to define them. You can add endpoints to an API after you have created the API.

To define endpoints:

1. Ensure that you have defined the general settings and the backend service for the API before proceeding.
2. Click **Endpoints > Add Endpoint** to define the endpoints for the API.
3. Click **General**, then define the general settings for the endpoint of the API:

   Methods

   Select the appropriate REST methods for the API. The methods define the action that the API performs. For a definition of each method, see:

   - HTTP Request Methods (https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol#Request_methods)
   - RFC 7231 Request Methods (https://tools.ietf.org/html/rfc7231#section-4)
   - RFC 5789 Patch Method for HTTP (https://tools.ietf.org/html/rfc5789)

   Endpoint URL Path

   Specify the URL path for the API endpoint. The path can be anything that you want. The Publisher displays the path you specify under this field as **Access at http(s):** so that you can verify that the path is correct.

   Backend Service Path URL

   Specify the base of the URL for the backend service. The Publisher displays the path under this field as **Backend Service Endpoint Base URL:** so that you can verify that it is correct.

   Action

   Select how you want to include the backend service URL to the API. You can do nothing, prepend, append, or replace the backend service URL in the API.

   Mime Type

   Specify the mime type of the API endpoint. For example, `application/json` or `text/html`.
4. Click **Parameters**, then click **Add Parameter** to define the parameters for the API:

   Name

   Specify a name for the parameter.

   Type

   Select whether the parameter performs queries, contains data, or is a header.

   Description

   Specify a detailed description of the API parameter so that other people can understand the endpoint parameter.

Data Type

Select the appropriate data type of the parameter.

Value

Specify the value for the parameter.

Action

Select the type of action the parameter performs.

Required

Select **Required** if the parameter is required for the API endpoint.

5 Click **OK**.

6 (Optional) To add additional parameters, click **Add Parameter**, then repeat Step 4 and Step 5 for each additional parameter.

7 (Optional) To add additional endpoints, click **Add Endpoint**, then repeat Step 3 through Step 5 for each additional endpoint.

8 Click **OK** to save the API.

or

Click **Rate Limiting Policies** to define rate-limiting policies per subscription and per user.

9 (Optional) Click **Show JSON** to view the API in JSON.

Define Rate Limiting Policies

Publisher > New > Rate Limiting Policies

Secure API Manager allows you to create rate-limiting policies to protect the throughput to the backend service so that is not overwhelmed. There are three types of rate-limiting policies:

- **Limiting Policies:** These policies are per API and the Secure API Manager administrator is responsible for creating and maintaining the Limiting Policies. These Limiting Policies protect the API Gateway from being overwhelmed with requests.
- **Subscription Rate Policies:** These policies are per subscription, and per API. The API developers are responsible for creating and maintaining the Subscription Rate Policies.
- **Subscription User Rate Policies:** These policies are per user, per subscription, and per API. The API developers are responsible for creating and maintaining the Subscription User Rate Policies.

You select the Limiting Policies in the **Subscription Tier** when you define the general settings for the API. If you have questions about what the Limiting Policies allow, you can also ask the Secure API Manager administrator for more details.

As the API developer, you can create two additional rate-limiting policy types. Limiting Policies take precedence over Subscription Rate Policies and Subscription User Rate Policies. For example, you create a Subscription Rate Policy that allows 5,000 requests per second. The Limiting Policy allows only 1,000 requests per second. You will see a maximum of 1,000 requests per second. The Limiting Policies take precedence over these other rate-limiting policies to protect the functionality of the API Gateway.

To define the rate-limiting policies:

1. Ensure that you have defined the general settings, the backend service, and the endpoints for the API before you proceed.
2. Click **Rate Limiting Policies**, then define the appropriate rate-limiting policy for your environment.

   Subscription Rate Policies

   Secure API Manager applies these policies per subscription to the API.

   Subscription User Rate Policies

   Secure API Manager applies these policies per user and per subscription to the API.
3. Click **New Policy** for the appropriate rate-limiting policies for your environment, then use the following information to create the rate-limiting policy:

   Quota

   Specify the number of requests per the time period, then select the time period you want to use. Read the information about the request count policy to understand how Secure API Manager processes the requests to the APIs.

   Count

   Specify the maximum number of requests that Secure API Manager allows during a certain period of time.

   Time Period

   Specify the amount of time when Secure API Manager limits the requests to the APIs in seconds, minutes, or hours.

   Burst Rate Limit

   If required, define a Burst Rate Limit for the users and subscriptions of this API. to help control the number of requests that happen during the specified time period. The Burst Rate Limit works only if you set the values to be more restrictive than the Quota.

   Define optimal burst rate limit

   Select this option to be able to define the optimal burst rate limits for the quota. If you do not select this option, you cannot define anything else for this rate-limiting policy.

   Count

   Specify the maximum number of requests that Secure API Manager allows during a certain period of time. For the Burst Rate Limit to work, this value must be smaller than the value you defined in the **Quota**.

   Time Period

   Specify the amount of time when Secure API Manager limits the requests in seconds, minutes, or hours. This value must be smaller than the value you defined in the Quota.
4. Click **OK** to save the API.
5. (Optional) Click **Show JSON** to view the API in JSON.

Manage APIs

Select **Publisher**

The Publisher provides a single location for you to manage the APIs hosted in Secure API Manager. There are several different management tasks you can perform in the Publisher. You can search for APIs and include the tags you specified when you created the API in the search parameters.

- "View APIs in the Publisher" on page 20
- "Edit an API or View a Deprecated or Subscribed API" on page 20
- "Clone an API" on page 21
- "Deprecate an API" on page 21
- "Manage Multiple APIs" on page 22

View APIs in the Publisher

The Publisher displays all of the APIs stored in Secure API Manager. It also provides different ways to view and search for the APIs.

To view the APIs:

1. Access the Publisher.
2. Select the appropriate API Gateway cluster to view the APIs associated with it.
3. To search for a specific API, in the **Search field**, specify the name of the API to search for the API.
4. To view the APIs, use one or more of the following options:
   - **4a** In the upper right corner, click **View deprecated APIs** to view only the deprecated APIs.
   - **4b** In the upper right corner, click **View unsubscribed APIs** to view only the unsubscribed APIs.
   - **4c** In the upper right corner, click **View table layout** to view all of the APIs in a table layout.
   - **4d** In the upper right corner, click **View tile layout** to view all of the APIs in a tile layout.
   - **4e** In the upper right corner, in the **Sort by** field, click **Name** to sort the APIs by name in ascending or descending order.
   - **4f** In the upper right corner, in the **Sort by** field, click **Description** to sort by the description in ascending or descending order.

Edit an API or View a Deprecated or Subscribed API

You can change any of the parameters, endpoints, or backend service information stored in an API. It allows you to update an API if the certificate changes or if the backend service changes.

You cannot edit an API if it has subscriptions or if it is deprecated. If an API has subscriptions, that means that the API is in use and you cannot make changes to it, otherwise the service the API provides would be disrupted. If an API is deprecated, it is no longer in use. You can view the details of these APIs but you cannot edit them.

The best practice for editing an API is to make a clone of the API and then make the appropriate changes to the clone. This results in a new API with a new version as well as no interruptions to currently in-use APIs.

To edit an API or view a deprecated or subscribed API:

1. Select the API you want to edit.
2. In the upper right corner of the API, click the **Actions**.
3. Click **Edit** or click **View**.

   The **Edit** option appears if the API is not deprecated or does not have subscriptions. The **View** option appears if the API is deprecated or has subscriptions.
4. View the details of the API or change the appropriate items in the API.

Clone an API

Secure API Manager provides this cloning feature to simplify creating and deprecating APIs. The best practice for using cloning is to create a clone of an API you want to deprecate and then deprecate the first the API. Secure API Manager automatically increases the version number of the API for you when you clone an API. Cloning saves you time so that you do not have to enter the same parameters again for a new version. After you clone the API, you modify the API.

IMPORTANT: Secure API Manager does not allow you to clone an API more than one time. If you need a new clone, clone the cloned API and deprecate the first clone.

To clone an API:

1. Select the API you want to clone.
2. In the upper right corner of the API, click the **Actions**.
3. Click **Clone**.
4. Make the appropriate changes to make a new API. A best practice is to update the description of the API to record what is different in the new API from the original API.

   NOTE: Secure API Manager does not allow you to change the version number of the cloned API. It automatically increments the version of the cloned API for you.

Deprecate an API

If you have created a new version of an API but you need to keep the older APIs for compliance or other reasons, you can deprecate these APIs. The deprecated APIs still appear in the Publisher but the Store does not display them.

To deprecate an API:

1. In the Publisher, select the appropriate API to deprecate.
2. In the upper right corner of the API, click the **Actions**.
3. Click **Deprecate**. The API now contains an exclamation mark on it.
4. (Conditional) If you need to make a deprecated API available again, click the **Actions**, then click **Deprecate** again. The exclamation mark is removed and anyone can access the API in the Store again.

Manage Multiple APIs

The Publisher allows you to delete or deprecate multiple APIs at the same time. This allows you to work more efficiently.

To manage multiple APIs:

1. In the Publisher, select **Edit** in the top left corner.
2. Click the APIs that you want to delete or deprecate.
3. In the top left corner, click **Delete** or **Deprecate**, then click the confirmation message.
4. In the top left corner, click **Close** to exit the editor.

3 Using the Store

The **Store** is a single location in Secure API Manager where you can access, view, and subscribe to the available APIs. The Store is a web application that you access through an appmark in the Access Manager user portal. The Store appmark is named `APIs: Subscribe`. Your administrator or IT department must assign you the proper permissions to be able to access this appmark.

Create an API Group

Select **Store**.

To be able to subscribe to an API you must create an API group in the Store for the APIs that you want to use. To secure access to the APIs, Secure API Manager is tightly integrated with Access Manager to provide the OAuth authorizations for the APIs. The API group allows you to select the available OAuth clients, roles, and scopes from Access Manager to limit access to the APIs through the OAuth authorizations.

The Access Manager roles and scopes that you assign when you create the API group are global roles and scopes. The roles and scopes apply to all APIs that you subscribe to in the API group. You can add more granular access by adding additional roles and scopes to an API endpoint.

Secure API Manager stores the APIs on a specific Identity Server cluster. When you create an API group, you must select the Identity Server cluster that contains the APIs you want to use.

To create a group:

1. In the **Cluster** field, select the appropriate Identity Server cluster.
2. Click **New**.
3. Use the following information to define the group:

   API Group Name

   Specify a unique name for the API group.

   OAuth 2 Client

   Select an OAuth client from Access Manager to provide the OAuth tokens to secure access to the API.

   Manage OAuth Clients

   Register and manage the OAuth clients that you select for use with this group of APIs. You must register the OAuth client to allow the API authorization to work.

Scopes

Select the default scope APIManagerScope or any scopes that you have created to restrict access to the APIs. The scopes that the Store displays are from Access Manager. These are global scopes for all APIs assigned to this API group.

Roles

Select appropriate roles to restrict access to the API. These are global roles for all APIs assigned to this API group.

4 Click **OK** to save the group.

5 (Optional) Click **Show JSON** to view the API group in JSON.

Manage the OAuth Clients

Select **Store > New > Manage OAuth Clients**

You select which OAuth client that you want to use when you create the API group to allow the API authorizations to work. After you select the proper OAuth client you must register the OAuth client with the Access Manager Identity Server. When you create the API group, you can register, edit, view, or delete any of the selected OAuth clients.

You manage the OAuth clients for the API group in the Store.

- "Register an OAuth Client" on page 24
- "Edit a Registered OAuth Client" on page 27
- "View an OAuth Client that Contains the Client ID and Client Secret" on page 27
- "Delete a Registered OAuth Client" on page 27

Register an OAuth Client

Select **Store > New > Manage OAuth Clients > Register New Client**

You must register the OAuth client that you selected when creating the API group with the Identity Server in Access Manager. Registering the OAuth client allows the Identity Server to authorize access to the APIs if the calls to the APIs have the proper information about the OAuth client in them.

To register an OAuth client:

1 (Conditional) If you are creating a new API group, click **New**.

2 (Conditional) If you want to register a new client to an existing API group, in the upper right corner of the API group, click **Actions**, then click **Edit**.

3 Click **Manage OAuth Clients**.

4 Click **Register New Client**.

5 Under **Client Configuration**, use the following information to configure the OAuth client:

Enable Client

Select **Enable Client** to allow this OAuth client to authorize requests to the APIs assigned to the group.

Client Name

Specify the name of the OAuth client that appears in the list of available OAuth clients when you create the API group.

Client Type

Select **Web Based** for the client type. Secure API Manager supports only web-based OAuth client applications.

Login Redirect URIs

Specify the URI for the client type that the Identity Server uses to send the authorization code and implicit requests. The format for the web-based OAuth client application is:

`https://client.example.org/callback`

Grants Required

You must select certain options for Secure API Manager to work. You can select more of the available grant types if you need them for your environment. Available grant types are:

- Authorization Code - mandatory
- Implicit
- Resource Owner Credentials - mandatory
- Client Credentials - mandatory
- SAML 2.0 Assertion

Token Types

You must select certain tokens that the authorization server uses to send to this client application. The token types are:

- Code - mandatory
- ID Token
- Refresh Token
- Access Token - mandatory

Refresh Token

Select **Always Issue New Token** if you want to issue a new refresh token for each refresh token request.

6 (Conditional) If you selected **ID Token** in **Token Types** under **Client Configuration**, click **OpenID Connect Configuration**, then configure the following settings:

JSON Web Key Set URI

To encrypt the ID token using the public key of the client application, you must specify the JSON public key URI for the client. The Identity Server requires the public key to retrieve the encryption key for the JSON public key URI. For example:

`https://client.example.org/my_public_keys.jwks`

ID Token Signed Response Algorithm

Select **RS256**. This is the algorithm that the Identity Server uses.

WARNING: If you select **None**, the Identity Server sends the ID token as an unsigned token. Ensure that you select **None** only if you can trust the integrity of an unsigned ID token.

ID Token Encrypted Response Algorithm

Select **RSA1_5.** Ensure that you select the same algorithm that you defined in the specified JSON Web Key Set URI so that the client application can use the private key to decrypt the token.

ID Token Encrypted Response Enc

This field gets automatically populated based on the algorithm selected in **ID Token Encrypted Response Algorithm**. It should be **A128CBC-HS256** for the **RSA1_5** algorithm.

7 (Optional) Click **Token Configuration**, then configure the settings for the token using the following information:

NOTE: These settings override the global settings for the Identity Server that the Access Manager administrator has defined.

Authorization Code Timeout

Specify the duration after which the authorization code expires.

Access Token and ID Token Timeouts

Use the default values for the Secure API Manager configuration.

Refresh Token Timeout

Use the default values for the Secure API Manager configuration.

Access Token and Refresh Token Format

Select the **JWT** token format. This is required for Secure API Manager to work.

8 (Optional) Click **Logout Configuration** to configure logout options and behaviors for the OAuth client using the following information:

Logout URI

Specify the URL that Identity Server uses to log out a user.

Enable Session Token

Select this option to send session ID and issue query parameters to the iframe HTML element. OpenID provider monitors the login status of a client application through the iframe HTML element.

Logout Redirect URIs

Specify the URL where the Identity Server redirects the user after logout. For example, `https://`*`client.example.org`*`/logout`.

9 (Optional) Click **Consent Screen Configuration** to configure any consent information that you want to present to that user.

Client Logo URL

Specify the URL of the logo that you want to include on the consent page.

Privacy Policy

Specify the URL of the privacy policy you want to include on the consent page. You can define your privacy policy.

Terms of Service URL

Specify the URL of the terms of service.

Contacts

Specify the email addresses of the people related to this client application.

10 (Optional) Click **Authorized JavaScript origins (CORS)** and add **Domains**. Domains configured here can access restricted resources available on the client application. Do not specify the port if you are using port 80 or 443. For example:

```
beem://www.test.com:port, fb://app.local.url:port, https://
namapp.com:port
```

11 Click **OK** to register the client with the Identity Server.

Edit a Registered OAuth Client

You can change the information in the OAuth client at any time for any reason. You access the registered OAuth clients in the API group.

To edit a registered OAuth client:

1 In the API group that contains the OAuth client, click the menu in the upper right corner, then click **Edit**.

2 Click **Manage OAuth Client**.

3 On the right side of the registered OAuth client, click **Edit**.

4 Make any of the appropriate changes for the OAuth client. The fields are the same ones that you see when you register an OAuth client.

5 Click **OK** to save your changes.

View an OAuth Client that Contains the Client ID and Client Secret

You need to access the client ID and client secret of a registered OAuth client to add the calls for the APIs to ensure that the calls can be authorized by the OAuth client through the Identity Server.

To view the details of a registered OAuth client:

1 In the API group that contains the OAuth client, click the menu in the upper right corner, then click **Edit**.

2 Click **Manage OAuth Client**.

3 On the right side of the registered OAuth client, click **View**.

4 In the top section, you see the **Client ID** and an option to click to view the **Client Secret**.

5 Click **OK** to close the window.

Delete a Registered OAuth Client

You can delete any registered OAuth clients from the configuration of any API group.

To delete a registered OAuth client:

1 In the API group that contains the OAuth client that you want to view, click the menu in the upper right corner, then click **Edit**.

2 Click **Manage OAuth Client**.

3 On the right side of the registered OAuth client, click **Delete**.

4 Confirm the deletion.

5 Click **OK** to close the window.

Manage the API Groups

Select **Store**

The Store allows you to manage the API groups that you create. You can view, sort, clone, and delete the API groups that you create.

- "View the API Groups in the Store" on page 28
- "Edit an API Group" on page 28

View the API Groups in the Store

The Store displays all of the API groups stored in Secure API Manager. It also provides different ways to view and search for the API groups.

To view the API Groups:

1 In the Store, select the appropriate Identity Server cluster to view the API groups associated with it.

2 In the **Search** field, specify the name of the API group to search for the API group by name.

3 Use one or more of the following options to view the API groups:

3a In the upper right corner, click **View table layout** to view all of the API groups in a table layout.

3b In the upper right corner, click **View tile layout** to view all of the API groups in a tile layout.

3c In the **Sort by** field, click **Name** to sort the API groups in name by either ascending or descending order.

Edit an API Group

You can change any of the settings of the API group or update information about the OAuth clients when you edit an API group. If you need to manage the OAuth clients that you have registered or you need to register a new OAuth client, you perform these procedures when you edit the API group.

To edit an API group:

1 Select the API group that you want to edit.

2 In the upper right corner of the API group, click the **Actions**.

3 Click **Edit**.

4 (Optional) In **Scopes** and **Roles**, click the plus sign to add additional scopes and roles specific to this API.

5 Change the appropriate settings in the API group or manage the OAuth clients.

Manage Subscribed APIs

Select **Store** > ***API Group*** > **Manage Subscribed APIs**

The Store allows you to subscribe or unsubscribe from the APIs in a specific API group. When you subscribe to the API, you select the subscription tier and you can see the available endpoints in the API.

- "Subscribe to an API" on page 29
- "Unsubscribe from an API" on page 29
- "Add Specific Roles and Scopes to an API Endpoint" on page 30

Subscribe to an API

After you create an API group, you can view and subscribe to the available APIs in the API group. You must subscribe to the APIs to be able to use them. After you subscribe to an API you should select which subscription tier you want to use and you can see the available endpoints for the API.

To subscribe to an API:

1 On the API group, click **Manage Subscribed APIs**.

2 Search for the available APIs on the right side of the page.

3 Select the API you want to subscribe to, then drag and drop it on the left side of the screen.

4 To add the subscription tier or view the available endpoints:

4a In the upper right corner of the API, click **Edit**.

4b In **Subscription Tier**, view the subscription tier assigned to this API.

4c In **Endpoints**, you can view the endpoints for the API or add roles and scopes to the endpoint.

4d Click **OK** to save the changes.

5 Click **OK** to save the subscriptions.

Unsubscribe from an API

You can subscribe from any API at any time. You might unsubscribe from an API if you no longer use the API.

To unsubscribe from an API:

1 On the API group, click **Manage Subscribed APIs**.

2 Select the subscribed API on the left side of the screen and drag it to the right side of the screen.

3 Click **OK** to save the changes.

Add Specific Roles and Scopes to an API Endpoint

Select ***API*** **> Edit >** ***API Endpoint*** **> Scope** and **Role**

You can add roles and scopes to a specific API endpoint of a subscribed API. You define global roles and scopes when you create an API group. If you need more granular control to a specific API endpoint, you can add additional roles and scopes to an API endpoint if you are subscribed to the API.

To add specific roles and scopes to a subscribed API endpoint:

1 On the API group, click **Manage Subscribed APIs**.

2 On the subscribed API, click **Edit**.

3 Click the appropriate endpoint.

4 To add scopes to the endpoint:

- **4a** View the global scopes in the **Scopes** field, then click **Global Scopes** to view the currently assigned global scopes.
- **4b** In **Scopes**, click the plus sign.
- **4c** Select one or more of the additional scopes you want to add that are not in the global scopes.
- **4d** Click outside the list of scopes to close the list of scopes.

5 To add roles to the endpoint:

- **5a** View the global roles in the **Roles** field, then click **Global Roles** to view the currently assigned global roles.
- **5b** In **Roles**, click the plus sign.
- **5c** Select one or more of the additional roles you want to add that are not in the global roles.
- **5d** Click outside the list of roles to close the list of roles.

6 Click **OK** twice to save the changes.

Obtain an OAuth Token to Make API Calls

After you have subscribed to an API, you can use two different OAuth methods to make API calls through Secure API Manager. The endpoints are defined in the OAuth 2.0 RFC under Protocol Endpoints.

- "Authorization Endpoint" on page 31
- "Token Endpoint" on page 31

Authorization Endpoint

The authorization endpoint employs Access Manager as the owner of the user's credentials and the handler of the authentication process. The advantage of using this method is that the client implementation never obtains the user's credentials. It also allows Access Manager to execute multi-factor authentications. This is the preferred method.

To invoke the authorization profile, create an authorized endpoint request as described in the *Access Manager 5.0 OAuth Application Developer Guide*. You must perform an additional step that is specific to Secure API Manager. You must include the scope `APIManagerScope` in the list of scopes in the request.

Token Endpoint

The token endpoint profile allows a client that has the user's credentials to obtain an OAuth token in a single request. To invoke this endpoint, the client must own the client's username and password. Many client implementations do not want to protect the user's credentials (or ever have them go to the client), so this method is generally not preferred.

To invoke the token profile, create a token endpoint request as defined in the *Access Manager 5.0 OAuth Application Developer Guide*. You must perform an additional step that is specific to Secure API Manager. You must include the scope `APIManagerScope` in the list of scopes in the request. For example:

```
grant_type=password&client_id=557c9074-6a09-4f1a-83a6-
0d3b442e1cc0&client_secret=he3Mmy3IGgL9dxhufNHo312DQqTJYI8mB6GHA&username=
fred&password=test0123&scope=APIManagerScope
```

4 Controlling Access to the APIs through the Roles and Scopes

The Secure API Manager roles (https://www.microfocus.com/documentation/secure-api-manager/2-1/secure-api-manager-admin/configure-policies.html) control who has access to which Secure API Manager consoles. They do not control access to the APIs. It is the Access Manager roles and scopes that limit access to the APIs. You must configure Secure API Manager to use the Access Manager roles and scopes. Secure API Manager creates a scope for you to use to limit access to the APIs. Use the following information to understand this process and how to configure Secure API Manager.

- “Understanding How Secure API Manager Uses the Access Manager Scopes and Roles to Determine API Access” on page 33
- “Configuring the Access Manager Scopes and Roles Usage in Secure API Manager” on page 35
- “Restricting Access to APIs with Access Manager Scopes and Roles in the Publisher” on page 37
- “Modifying the Applications or Services that Access the APIs” on page 37

Understanding How Secure API Manager Uses the Access Manager Scopes and Roles to Determine API Access

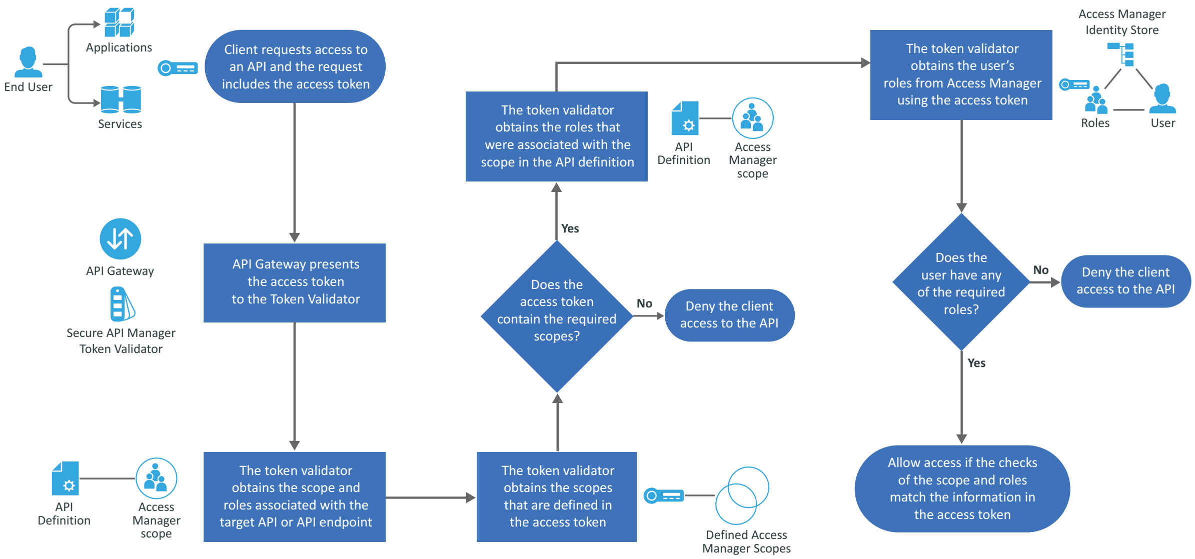
Secure API Manager allows you to make APIs accessible to the public or limits access to the APIs. To make APIs available to the public means that anyone who knows the full path to the API can call the API. Secure API Manager integrates with Access Manager to provide OAuth2 tokens for any requests to the APIs. This is a separate point of integration with Access Manager than controlling specific users to be able to access the APIs.

Secure API Manager uses the Access Manager roles and scopes to control access to the APIs and specific API endpoints. This way you do not have to create and manage additional accounts or roles for users to access and use the APIs.

To control access to the APIs you create a scope for one or more APIs. If you have multiple API endpoints, you can also use one scope to control access to all of the API endpoints. You would create a different scope for an API endpoint if you want a different set of users to be able to access a specific API endpoint. You can use the same scope for multiple APIs or for multiple API endpoints associated with different APIs. You associate the Access Manager roles with each scope to control the access to the APIs and the API endpoints.

The following graphic depicts how Secure API Manager integrates with Access Manager to control access to the APIs.

Figure 4-1 *How Secure API Manager integrates with Access Manager to control access to the APIs*


End User
Applications
Services
Client requests access to an API and the request includes the access token
API Gateway
Secure API Manager Token Validator
API Gateway presents the access token to the Token Validator
API Definition
Access Manager scope
The token validator obtains the scope and roles associated with the target API or API endpoint
The token validator obtains the scopes that are defined in the access token
Defined Access Manager Scopes
Does the access token contain the required scopes?
No
Deny the client access to the API
Yes
The token validator obtains the roles that were associated with the scope in the API definition
API Definition
Access Manager scope
The token validator obtains the user's roles from Access Manager using the access token
Access Manager Identity Store
Roles
User
Does the user have any of the required roles?
No
Deny the client access to the API
Yes
Allow access if the checks of the scope and roles match the information in the access token

The workflow shows how Secure API Manager controls access to the APIs through the Access Manager scopes:

1. An application, service, or client makes a call to the API and that request includes an access token.
2. The API Gateway presents the access token to the token validator for Secure API Manager.
3. The token validator obtains the scope and roles associated with the target API or target API endpoint.
4. The token validator obtains the scopes that are defined in the access token.
   a. The token validator checks the access token to see if it contains the required scopes for access.
   b. If the required scopes are not present, Secure API Manager denies the application, service, or client access to the API.
   c. If the required scopes are present, the token validator proceeds with additional evaluations.
5. The token validator obtains the roles that were associated with the scope in the API definition.
6. The token validator obtains the user's roles from Access Manager using the access token.
7. The token validator checks to see if the user has any of the required roles.
   a. If the required roles are not present, Secure API Manager denies the client access to the API.
   b. If the required roles are present, Secure API Manager allows access if the checks of the scope and the roles match the information in the access token.

Configuring the Access Manager Scopes and Roles Usage in Secure API Manager

By default, Secure API Manager creates a scope named APIManagerScope. You include the scope name in the Authorization Code grant type that the OAuth clients use to call the API. The OAuth2 protocol refers to the application, service, or client as a **client**.

Process with the following section, if you need to create a different scope. Otherwise, proceed to Restricting Access to APIs with Access Manager Scopes and Roles in the Publisher.

- "Creating an Attribute Map in Access Manager for Secure API Manager" on page 35
- "Creating a Scope for One or More APIs in Access Manager" on page 35

Creating an Attribute Map in Access Manager for Secure API Manager

You create an attribute map to control access to the APIs that are available in the Store. The attribute maps allows the Secure API Manager token validator to access the Access Manager user's roles to ensure that the user has the correct roles to access an API or specific API endpoints.

To create an attribute map in Access Manager:

1. From the Access Manager Dashboard, click **Devices**, then select **Identity Servers**.
2. Click the **Shared Settings** tab.
3. Click **New** to create a new attribute map.
4. Specify a unique name that you can remember and that you associate with Secure API Manager, such as *ForSAPIMAllUserRoles*.
5. Click **Finish**.
6. Select **Support WSTrust and Oauth**, then click **Next** at the end of the page.
7. Click **New** to add an attribute definition to the map.
8. Select **Local attribute**, then select **All Roles**.
9. Click **OK** to save the attribute map entry, then click **Finish** to complete the creation of the attribute map.

Creating a Scope for One or More APIs in Access Manager

You create a scope for one or more APIs or API endpoints that you want to protect with Access Manager. You can use the same scope for multiple APIs or for specific API endpoints that cross multiple APIs in Secure API Manager.

1. From the Dashboard in Access Manager, click the Identity Server cluster you associated with the API Gateway cluster.
2. Click the **OAuth and OpenID Connect** tab.
3. On the menu, click **Resource Servers**.

4 Create a new resource server for the APIs or API endpoints as follows:

- **4a** Click **New.**
- **4b** Specify a name for the resource server that represents the APIs or API endpoints so that it is easy to remember.
- **4c** Click **Finish** to create the new resource server.
- **4d** Repeat Step 4a through Step 4c to create a new resource server for the APIs or API endpoints you want to protect.

5 Add a scope to each resource server for the APIs or API endpoints as follows:

- **5a** On the Resource Server page, click the name of the appropriate resource server.
- **5b** Click **Scopes**.
- **5c** Click **New**.
- **5d** Specify a scope name and description. Ensure that you use something that represents the APIs or API endpoints.
- **5e** Click **User Attributes**, then select the **Require user permission** option. Ensure that you select this option, otherwise the integration fails.
- **5f** Click **Next** at the end of the page.
- **5g** On the Step 2 page, select the attribute map you created in Step 4 on page 35.
- **5h** Click **Finish** at the end of the page.
- **5i** Repeat Step 5a through Step 5h for each scope that you need to create for each resource server.

6 Update the Identity Server cluster with the new resource servers and scopes as follows:

- **6a** From the Dashboard, click **Devices > Identity Servers**.
- **6b** In the **Status** column, click **Update All**.
- **6c** Click **OK**.

7 Update Secure API Manager to have the roles and scopes appear as follows:

- **7a** From the Access Manager Dashboard, click the appropriate API Gateway cluster.
- **7b** Click **Actions > Update All**.

8 (Optional) Create any role policies that might be required for API access. The set of roles that the API developers see in the Publisher comes from the list of all configured role policies (https://www.microfocus.com/documentation/access-manager/5.0/admin/b5pm00a.html) in Access Manager.

Restricting Access to APIs with Access Manager Scopes and Roles in the Publisher

You can create the resource servers and scopes before or after you create the APIs in the Publisher. At some point in the process, you must associate the scopes in Access Manager with the APIs or specific API endpoints defined in the Publisher to control access to the APIs or the specific API endpoints.

You can associate a single scope with one or more APIs, all of the endpoints in an API, or with a specific API endpoint. To limit access to an API, you must assign the scope to the API and assign the scope to the single API endpoint in the API. You can assign the same scope to multiple endpoints in an API or you can assign different scopes to the different endpoints in the API if you want different users accessing the different endpoints. You select the scope when you create an API group.

If you want to limit API access by roles, you must include the scope name in authorization code grant from the client that calls the APIs. For example, if you are using the default scope, the OAuth client must use APIManageScope as the authorization code grant for any OAuth client that consumes the API.

Modifying the Applications or Services that Access the APIs

After you have linked the API or the specific API endpoints to the Access Manager scopes, you must modify the application, service, or client that makes the call to the APIs stored in the API Gateway. The OAuth2 protocol refers to the application, service, or client as a **client**.

If you do not modify the client, when a user tries to use the client, they receive a 401 or 403 error. The 401 error means that the token is invalid or missing. The 403 error means that the token is valid but it is missing the required scope or role for the endpoint. Both errors mean that the client is missing the access token from the OAuth2 request or that the current access token is invalid. To ensure that the users do not receive the 401 or 403 errors you must add an Authorization Code to the client.

When you add the Authorization Code to the client, it builds an Authorization Code OAuth profile request to initiate user login to obtain an access token. If the API endpoint requires a scope, you must add the name of the Access Manager scope in the body of the request. For more information, see the OAuth 2.0 Authorization Code Grant (https://oauth.net/2/grant-types/authorization-code/) website.