NetIQ Secure API Manager 1.0

API Management Guide

September 2019

Legal Notice

Contents

About this Book

The *NetIQ Secure API Manager API Management Guide* 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. This guide assumes that you know and understand these concepts. For more information, see “Required Knowledge” on page 8.

Access Manager Administrators

Administrators who add scopes and roles to control access to the APIs that are available in the Secure API Manager Store.

Developers

Developers who access and use the Secure API Manager Publisher and Store. They create new APIs, use existing APIs, and combine multiple APIs to create new services that end users can use.

Other Information in the Library

The library provides the following information resources in addition to this guide:

Release Notes

Provide information specific to this release of Secure API Manager, such as known issues.

Installation Guide

Provides installation steps specific to this release of Secure API Manager.

Administration Guide

Provides configuration and management information for this release of Secure API Manager.

1 API Management Overview

Application programming interfaces (APIs) are sets of definitions, protocols, and tools for building software. Much 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 software to solve your business problems.

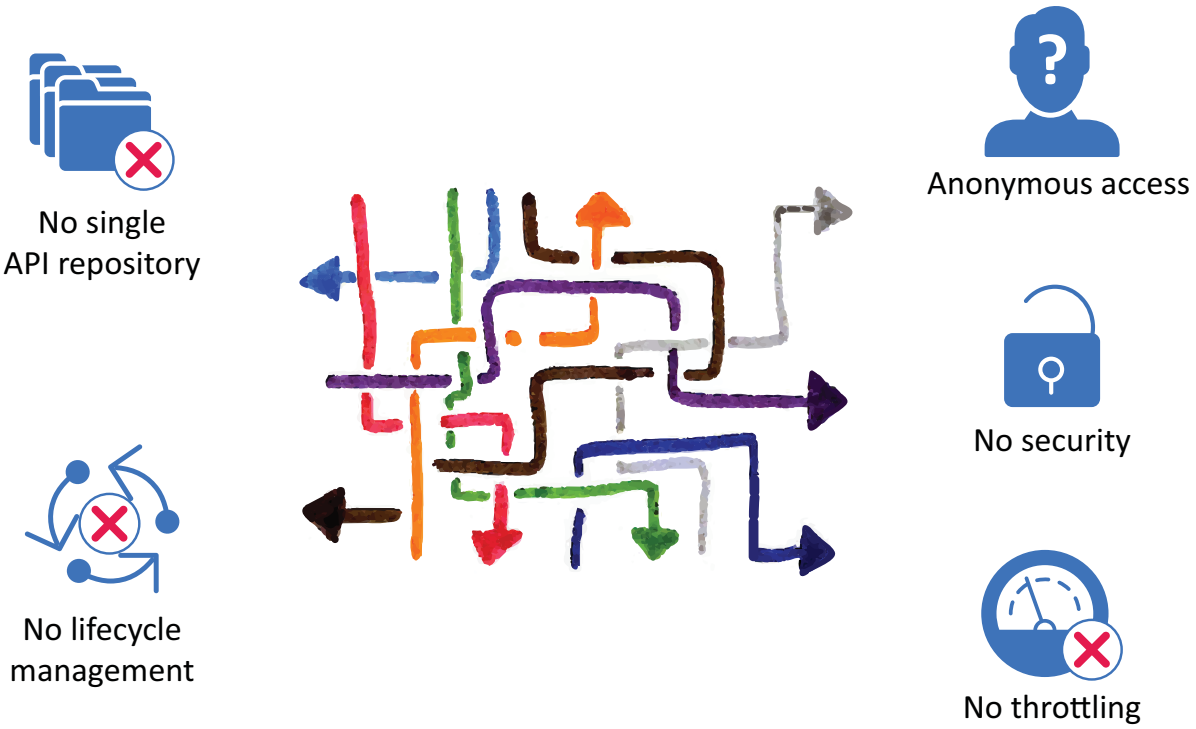
Secure API Manager gives you a single place to add, manage, audit, and secure the APIs that your company uses. You add the APIs once to Secure API Manager and they are available for reuse. You can see all of the available APIs in a single location, making it easy for you to combine multiple APIs to create new functionality while seamlessly requiring access to the APIs through NetIQ Access Manager.

- "How Secure API Manager Solves API Management Issues" on page 7
- "Required Knowledge" on page 8

How Secure API Manager Solves API Management Issues

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

Figure 1-1 *Business Problems Managing APIs*



- No single repository

- No lifecycle management
- Anonymous access
- No security
- No throttling

NetIQ solves these issues by providing a system that allows you to manage, create, control, and audit the APIs used in your environment through Secure API Manager. It gives you:

- A single repository for all of your APIs
- A lifecycle system to track the state of the APIs
- Throttling capabilities to limit throughput to certain APIs
- A detailed analytics system to show you which APIs are being used the most
- Secure access to the APIs due to integration with NetIQ Access Manager

NetIQ intends for you to use Secure API Manager in conjunction with NetIQ Access Manager to provide secure access to your APIs. The purpose of this guide is to help you understand how to add, manage, and secure the APIs for your company.

For detailed information about the purpose of Secure API Manager and the Secure API Manager architecture, see "Secure API Manager Overview" in the *NetIQ Secure API Manager 1.0 Installation Guide*

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

2 Getting Started

Secure API Manager contains different components that allow you to manage all of your APIs in a single repository. You can create APIs, create applications by combining multiple APIs, view subscriptions, and view API analytics. You access different components to perform different tasks. The components are:

- **Publisher:** Allows users with the correct roles to create new APIs or add existing APIs to the Store. Any Publisher role automatically grants rights to the Store as well.
- **Store:** Allows users with the correct roles to use the available APIs and combine APIs to make new applications. Users with the Store role do not have access to the Publisher.

Use the following information to help you access the different UI components.

- "Accessing the Publisher" on page 9
- "Accessing the Store" on page 9

Accessing the Publisher

You should receive an email from the administrator of Secure API Manager that contains a URL, a user name, and a password for you to use to log in to the system. The URL is the DNS name of the server that is running the Lifecycle Manager. For more information, see "Lifecycle Manager" in the *NetIQ Secure API Manager 1.0 Installation Guide*.

1. Access the URL provided in the email sent to you by the administrator.

   `https://lifecycle-manager-dns-name:9444/publisher`

2. Specify your user name and password, then click **Sign In**.

We recommend that you change your password for security reasons. You must access the Store to change your password. For more information, see "Accessing the Store" on page 9.

Accessing the Store

You should receive an email from the administrator of Secure API Manager that contains a URL, a user name, and a password for you to use to log in to the system. For more information, see "Lifecycle Manager" in the *NetIQ Secure API Manager 1.0 Installation Guide*.

1. Access the URL provided in the email from the administrator.

   `https://lifecycle-manager-dns-name:9444/store`

2. Specify your user name and password, then click **Sign In**.

After you log in, we recommend that you change your password for security reasons. To change your password, click your name in the toolbar, then click **My Account**.

3 Creating and Publishing APIs

Secure API Manager allows you to add all of the APIs in your IT environment to a single repository. The Publisher is that single repository where you store the APIs. You add existing APIs or create new APIs in the Publisher. You then publish the APIs to the Store where developers access all of the available APIs.

The people adding the APIs are developers or IT administrators who have been provided sufficient information to add the APIs to the Publisher. Use the following information to help you add APIs in the Publisher.

Ensure that you have the required knowledge before creating APIs. For more information, see “Required Knowledge” on page 8.

Designing a Prototype REST API

Secure API Manager allows you to create new REST APIs in the Publisher. This allows you to create and test a prototype REST API without worrying about users accessing and using the REST API before it is ready for production. The Publisher provides components that allow you to test the REST API and rewrite the REST API easily. You must know and understand REST to create a prototype REST API. For more information, see “Required Knowledge” on page 8.

When you create a REST API, you must use an inline script or a valid endpoint to complete the creation of the REST API. Typically, you create REST APIs with inline scripts for testing purposes. You use a REST API prototype for the purpose of early access to the REST API and testing the REST API before deploying it.

You can deploy a new REST API or a new version of an existing REST API as a prototype. This gives subscribers early access to the REST API that they can try out without a subscription or throttling, and provide feedback to improve the REST API. Over a period of time, you can make changes according to the feedback that you receive from the users and then publish the REST API in your production environment.

To design and publish a prototype REST API:

1. Log in to the Publisher using the account your Secure API Manager administrator gave you.

   `https://`*`liefcycle-manager-dns-name`*`:9444/publisher`

2. To create a prototype REST API:
   - **2a** Click **Design a New REST API**, then click **Start Creating**.
   - **2b** Use the following information to define the general information about the REST API:

Name

Specify a name for the REST API that appears in the Store. No spaces are allowed.

Context

Specify the URI context path of the REST API. It is case-sensitive.

Version

Specify the version of the REST API. This helps you manage the lifecycle of the REST API.

Visibility

Select whether the REST API is **Public** or **Restricted by Roles**. For more information, see Chapter 4, “Controlling Access to the APIs through the Access Manager Roles,” on page 23.

Description

Specify a description of the REST API that appears in the Store. The description helps people understand the purpose of the REST API.

Tags

Specify tags in a comma-separated list to describe the REST API. Use common words or key phrases as tags.

Select Image

Upload an image to represent the REST API in the Store. The maximum dimensions are 100 x 100 pixels.

2c Define the REST API. You must specify the path to the REST API resource you are creating. You can then either import a file or use the Swagger UI to define the REST API. You must also add the REST calls the API will use. For more information, see the “REST API Tutorial (https://restfulapi.net/)”.

3 To implement the prototyped REST API:

3a Click **Next: Implement**.

3b Select **Prototyped API**.

3c Select the implementation method of **Inline** or **Endpoint**. If you select **Endpoint**, you must provide a valid endpoint for the new REST API. If you select **Inline**, this REST API is for testing purposes only.

3d (Conditional) If you select **Inline**, expand the GET call and add your script here.

3e (Conditional) If your environment requires CORS, select **Enable API based CORS Configuration**.

CORS allows you to define additional domains that are in your environment. By default, to stop cross-site scripts, Secure API Manager does not allow multiple domain names. For more information, see “Cross-origins resource sharing Wiki.”

3f Use the following information to enable CORS:

Access Control Allow Origins

Select this option to allow all domain names that contain the origin domain name.

Access Control Allow Headers

Add any additional headers to this section to allow Secure API Manager to use additional domains.

Access Control Allow Methods

Ensure that the correct REST methods are listed.

Access Control Allow Credentials

Select this option to allow credentials from other domains.

4 Review the REST API resource information that you have created.

5 Click **Deploy as a Prototype**.

6 Select **Go to API Store** to open the new REST API in the Store.

7 Click the **API Console** tab.

8 Click **GET** to expand it, then click **Try it out** to test the REST API.

You can test the prototype REST API by accessing it and using it in the Store. For more information, see "Invoking and Testing the REST APIs" on page 40.

After you have thoroughly tested the prototype REST API, you can then publish the REST API in your production environment. For more information, see Chapter 8, "Managing the Lifecycle of an API," on page 43. If you have documentation to add to the API, proceed to Chapter 6, "Managing Documentation for the APIs," on page 33.

Adding and Publishing an Existing REST API

Secure API Manager allows you to add existing REST APIs to the Publisher to create a single repository for all of your APIs. The Publisher consumes the Swagger file or the Swagger URL for the existing APIs. You must understand REST and Swagger to create a REST API. For more information, see "Required Knowledge" on page 8.

If you do not have a Swagger file or Swagger URL for the API, you can manually add the API to the Publisher as if you were creating a new API. For more information, see "Designing a Prototype REST API" on page 11.

Adding the REST API to the Store occurs during the add process. This is called publishing the REST API.

To add and publish an existing REST API:

1 Log in to the Publisher using the account your Secure API Manager administrator gave you.

`https://lifecycle-manager-dns-name:9444/publisher`

2 To add a REST API:

2a Click **Add New API**.

2b Select **I Have an Existing API**.

2c Select **Swagger File**, browse to and select the file, then click **Start Creating**.

or

Select **Swagger URL** and specify the URL, then click **Start Creating**.

2d Use the following information to define the REST API:

Name

Specify a name for the REST API that appears in the Store. No spaces are allowed.

Context

Specify the URI context path of the REST API. It is case sensitive.

Version

Specify the version of the REST API. This helps you manage the lifecycle of the REST API.

Visibility

Select whether the REST API is **Public** or **Restricted by Roles**. For more information, see Chapter 4, “Controlling Access to the APIs through the Access Manager Roles,” on page 23.

Description

Specify a description of the REST API that appears in the Store. The description helps people understand the purpose of the REST API.

Tags

Specify tags in a comma-separated list to describe the REST API. Use common words or key phrases as tags.

Select Image

Upload an image to represent the REST API in the Store. The maximum dimensions are 100 x 100 pixels.

API Definition

This section contains all of the REST calls defined in the Swagger file. You can edit the Swagger file to make changes.

3 To implement the REST API:

3a Click **Next: Implement**.

3b Select **Managed API**.

3c Add the production endpoints of the REST API as follows:

Endpoint Type

Select **HTTP/REST Endpoint** for your endpoint type. You only use the SOAP endpoint only if you used SOAP to create the API.

Production Endpoint

Specify the back-end URL for the REST API.

Endpoint Security Scheme

(Conditional) If the REST API requires it, add the credentials of the back-end service.

3d Select whether to enable a message mediation policy.

The message mediation policy allows you to convert the input or output of the REST API from XML to JSON or from JSON to XML. For more information, see “Converting JSON to XML” on page 20.

3e (Conditional) If your environment requires CORS, select **Enable API based CORS Configuration**.

CORS allows you to define additional domains that are in your environment. By default, to stop cross-site scripts, Secure API Manager does not allow multiple domain names. For more information, see “Cross-origins resource sharing Wiki”.

3f Use the following information to enable CORS:

Access Control Allow Origins

Select this option to allow all domain names that contain the origin domain name.

Access Control Allow Headers

Add any additional headers to this section to allow Secure API Manager to use additional domains.

Access Control Allow Methods

Ensure that the correct REST methods are listed.

Access Control Allow Credentials

Select this option to allow credentials from other domains.

4 Click **Next: Manage API**.

5 Configure the REST API as follows:

Make this the Default Version

Select this option to make the version of the published REST API the default version so that when you access the REST API through a URL you do not have to enter a specific version. For example, if the REST API version is 2.5 and the URL is `https://my.company.com/timesheet/2.5`, users just have to enter `https://my.company.com/timesheet/`.

Transports

Select whether you want to use HTTPS or HTTP. HTTPS is the secure transport type.

Response Caching

Select whether you want to cache the response from the REST API. Caching is disabled by default. Enabling this option speeds up the response of the REST API because Secure API Manager caches the response. If you enable this option, ensure that you define a cache timeout period.

6 Configure the following throttling options for the REST API:

Maximum Backed Throughput

This option limits the number of calls Secure API Manager allows to the back-end. If you select **Specify**, you must specify the number of transactions per second (TPS) for the production environment and the sandbox environment.

Subscription Tiers

Select the appropriate tier that allows the correct number of requests per second. When users subscribe to the REST APIs, the subscription tiers controls the request to the API.

Advanced Throttling Options

Select whether you want the throttling policy applied at the REST API level. If you select the API level, Secure API Manager ignores the other policies and does not apply them.

7 Select whether the REST API is used in a production environment, sandbox environment, or both types of environments.

8 Define the business information about the REST API. For example, specify the business owner of the REST API and the technical owner of the REST API.

9 Add scopes to limit who has access to the REST API. For more information, see Chapter 4, “Controlling Access to the APIs through the Access Manager Roles,” on page 23.

10 Click **Save & Publish**, then decide whether to continue editing the REST API, access the Store, or view an overview of the REST API.

You can access and test the REST API in the Store to ensure that it works. For more information, see “Invoking and Testing the REST APIs” on page 40. If you have documentation to add to the REST API, proceed to Chapter 6, “Managing Documentation for the APIs,” on page 33.

Creating and Publishing a SOAP API

Secure API Manager supports both REST and SOAP APIs. It allows you to create APIs using an existing SOAP endpoint. You must understand SOAP to create a SOAP API. This assumes that you understand and know how SOAP works. For more information, see “Required Knowledge” on page 8.

You must use valid WSDL endpoints when you create a SOAP API or the Publisher does not create a working SOAP API. The Publisher validates the WDSL endpoints. If the Publisher cannot not validate the WSDL endpoints, it displays an error `Failed to process the WSDL` when you click **Next: Implement**. Ensure that you have valid WSDL endpoints before you try to create a SOAP API.

To create a SOAP API:

1 Log in to the Publisher using the account your Secure API Manager administrator gave you.

`https://dns-name-lifecycle-manager:9444/publisher`

2 To design a SOAP API:

2a Click **Add New API**.

2b Select **I Have a SOAP Endpoint**.

2c Specify the SOAP endpoint.

2d Select **Pass Through**, then click **Start Creating**.

or

Select **Generate REST API** if you want to convert the SOAP endpoint into a REST API, then proceed to Step 2d on page 13.

2e Use the following information to create the SOAP API:

Name

Specify a name for the SOAP API that appears in the Store. No spaces are allowed.

Context

Specify the URI context path of the SOAP API. It is case sensitive.

Version

Specify the version of the SOAP API. This helps you manage the lifecycle of the API.

Access Control

Select **ALL** to allow everyone to view and modify this SOAP API or select **Restricted by roles** to restrict who can view and modify this SOAP API by specific roles. For more information, see “Understanding the Secure API Manager Roles” in the *NetIQ Secure API Manager 1.0 Administration Guide*.

Visibility

Select whether the SOAP API is **Public** or **Restricted by Roles**. For more information, see Chapter 4, “Controlling Access to the APIs through the Access Manager Roles,” on page 23.

Description

Specify a description of the SOAP API that appears in the Store. The description helps people understand the purpose of the SOAP API.

Tags

Specify tags in a comma-separated list to describe the SOAP API. Use common words or key phrases as tags.

Select Image

Upload an image to represent the SOAP API in the Store. The maximum dimensions are 100 x 100 pixels.

WSDL

Displays the SOAP endpoint you entered on the prior page. Click **Test URI** to validate that this is a valid SOAP endpoint.

3 To implement the SOAP API:

3a Click **Next: Implement**.

3b Select **Managed API**.

3c Add the production endpoints of the REST API.

Endpoint Type

Select the endpoint type of **HTTP/SOAP Endpoint**. Only use the **REST endpoint** if you used REST to create the API.

Production Endpoint

Specify the production back-end URL for the SOAP API. If you require additional configuration options, click **Advanced Options** and make the appropriate changes.

Sandbox Endpoint

Specify a test back-end URL for the SOAP API to use when testing the API. If you require additional configuration options, click **Advanced Options** and make the appropriate changes.

Endpoint Security Scheme

(Conditional) If the SOAP API requires it, add the credentials of the back-end service.

3d Select whether to enable a message mediation policy.

The message mediation policy allows you to convert the input or output of the SOAP API from XML to JSON or from JSON to XML. For more information, see “Converting JSON to XML” on page 20.

3e (Conditional) If your environment requires CORS, select **Enable API based CORS Configuration**.

CORS allows you to define additional domains that are in your environment. By default, to stop cross-site scripts, Secure API Manager does not allow multiple domain names. For more information, see “Cross-origins resource sharing Wiki”.

3f Use the following information to enable CORS.

Access Control Allow Origins

Select this option to allow all domain names that contain the origin domain name.

Access Control Allow Headers

Add any additional headers to this section to allow Secure API Manager to use additional domains.

Access Control Allow Methods

Ensure that the correct REST methods are listed.

Access Control Allow Credentials

Select this option to allow credentials from other domains.

4 Click **Next: Manage API**.

5 Configure the API as follows:

Make this the Default Version

Select this option to make the version of the published SOAP API the default version so that when you access the SOAP API though a URL you do not have to enter a specific version. For example, if the API version is 2.5 and the URL is `https://my.company.com/timesheet/2.5`, users just have to enter `https://my.company.com/timesheet/`.

Transports

Select whether you want to use HTTPS or HTTP. HTTPS is the secure transport type.

Response Caching

Select whether you want to cache the response from the SOAP API. Caching is disabled by default. Enabling this option speeds up the response of the SOAP API because Secure API Manager caches the response. If you enable this option, ensure that you define a cache timeout period.

Authorization Header

Specify a value that Secure API Manager uses as a custom authorization header to send the access token in a request to consume the SOAP. If you leave this field blank, Secure API Manager uses the default authorization header.

6 Configure the throttling options for the API as follows:

Maximum Backed Throughput

This option limits the number of calls Secure API Manager allows to the back-end. If you select **Specify**, you must specify the number of transactions per second (TPS) for the production environment and the sandbox environment.

Subscription Tiers

Select the appropriate tier that allows the correct number of requests per second. When users subscribe to the SOAP APIs, the subscription tiers controls the request to the SOAP API.

Advanced Throttling Options

Select whether you want the throttling policy applied at the SOAP API level. If you select the SOAP API level, Secure API Manager ignores the other policies and does not apply them.

7 Select whether the SOAP API is used in a production environment, sandbox environment, or both types of environments.

8 Define the business information about the SOAP API. For example, specify the business owner of the SOAP API and the technical owner of the SOAP API.

9 (Conditional) If required, add additional properties for the SOAP API. You must specify a name and value for each property that you add.

10 Add scopes to limit who has access to the REST API. For more information, see Chapter 3, “Creating and Publishing APIs,” on page 11.

11 Click **Save & Publish**, then decide whether to continue editing the SOAP API, access the Store, or view an overview of the SOAP API.

You can access and test the SOAP API in the Store to ensure that it works. For more information, see “Invoking and Testing the SOAP APIs” on page 41. If you have documentation to add to the SOAP API, proceed to Chapter 6, “Managing Documentation for the APIs,” on page 33.

Creating and Publishing a WebSocket API

Secure API Manager allows you to create WebSocket APIs. WebSocket is a protocol similar to HTTP that is part of the HTML 5 specification. The WebSocket protocol enables real-time interaction between a web client (such as a browser) and a web server with low overheads. For more information, see “Required Knowledge” on page 8.

A WebSocket API allows a developer to expose WebSocket services as an API while providing OAuth security, throttling, analytics, and so forth through Secure API Manager.

Secure API Manager uses the Access Manager Gateway to deploy WebSocket APIs.

To configure WebSocket reverse proxy in Access Manager:

1. On the dashboard, click the **Access Gateways** icon.
2. Next to the gateway cluster, click **Edit**.
3. Click **Reverse Proxy / Authentication**.
4. For **Identity Server Cluster**, select your identity provider cluster.
5. Under **Proxy Settings**, ensure that **Enable Via Header** is selected.
6. In the **Reverse Proxy List** section, click **New**.
7. Provide a name for the reverse proxy, then click **OK**.
8. In the **Proxy Service List**, click **New**.
9. Provide a name for the proxy service.
10. For the **Published DNS Name**, enter the gateway DNS name.
11. For the **Web Server IP Address**, enter the IP address of the Secure API Manager gateway or L4 switch.
12. For the **Host Header**, select **Forward Received Host Name**.
13. Click **OK**.
14. Select **Enable SSL between Browser and Access Gateway**.
15. Ensure that **Enable SSL with Embedded Service Provider** and **Redirect Requests from Non-Secure Port to Secure Port** are selected.
16. Click **Auto-generate Key**, then click **OK**. Click **OK** again.
17. For **Non-Secure Port**, enter the appropriate port (80).
18. For **Secure Port**, enter the appropriate port (443).
19. Under the **Web Server Addresses** column in the **Proxy Service List**, click the IP address.
20. Ensure that **Enable Session Stickiness** is selected and **Connect Using SSL** is deselected.
21. For **Connect Port**:
    - Select **9100** if you are using combined admins and gateways
    - Select **9102** if you using separate admins and gateways
22. On the **Protected Resources** tab, click **New**.
23. Provide a name for the protected resource, then click **OK**.
24. (Optional) Provide a description for the protected resource.
25. For **Authentication Procedure**, select **OAuth Token**.
26. In the **URL Path List** section, click **New**.
27. For **URL Path**, enter `/wss*`, then click **OK**.

28 Deselect the **/wss** path and select the **/*** path, then click **Delete**. Click **OK**.

29 Click **OK** four times until you are on the Access Gateway Servers page.

30 Click **Security > Trusted Roots**.

31 Click **Auto-Import From Server.**

32 For **Server IP/DNS**, provide the IP or DNS for the Secure API Manager gateway (L4 switch or each Secure API Manager gateway).

33 For **Server Port**, enter 9443.

34 Provide a name for the certificate and click **OK**.

35 Click **OK**. With the new trusted root selected, click **Add Trusted Roots to Trust Stores**.

36 Click the pencil next to **Trust store(s)**.

37 Select all(?) the trust stores and click **OK**, then click **OK** again.

38 Click **OK** to add the trusted roots to the trust stores.

39 Click **Close**.

40 Update the Identity Servers and Access Gateways.

You can test and ensure that the WebSocket API works by accessing the new WebSocket API in Store. For more information, see "Invoking and Testing the WebSocket APIs" on page 42. If you have documentation to add to the API, proceed to Chapter 6, "Managing Documentation for the APIs," on page 33.

Converting JSON to XML

Secure API Manager allows you to convert the input messages to the APIs, the output of messages of the APIs to the back-end services, or show any exceptions that occur in the APIs from JSON to XML. You would do this if the API was written in XML and the service receiving the message excepts only JSON.

You enable Message Mediation on the Implement page in the Publisher whether you are creating a production API or editing an existing production API.

To enable Message Mediation when editing the API:

1 Log in to the Publisher using an administrator account.

`https://lifecycle-manager-dns-name:9444/publisher`

2 Find the appropriate API.

3 Click the **Edit** icon on the appropriate API to edit the API.

4 Scroll to the bottom of the page, then click **Next: Implement**.

5 Under the heading **Message Mediation Policies**, click **Enable Message Mediation**.

6 Click in the field of the appropriate option for your API.

Input Flow

Enable this option by selecting an available file listed when you click in the field or click the **Upload** icon to upload your own file.

The **Input Flow** allows you to take the incoming messages to the API and transform the JSON to XML when the message arrives at the API Gateway. If you want to revert back to XML, remove the file.

Output Flow

Enable this option by selecting an available file listed when you click in the field or click the **Upload** icon to upload your own file.

The **Output Flow** allows you to take the messages sent from the API to the back-end services and convert the messages from JSON to XML when the messages hit the API Gateway.

Fault Flow

Enable this option by selecting an available file listed when you click in the field or click the **Upload** icon to upload your own file.

By default, Secure API Manager writes error logs in XML. If an error occurs in the API, the **Fault Flow** allows to convert the XML message to JSON and then the API Gateway writes the log in JSON.

7 At the bottom of the page, click **Next: Manage**, then scroll the bottom of the next page and click **Save and Publish** to save the Message Mediation policies.

After you have added the Message Mediation policies to the API, you can test the functionality in the Store through the **API Console** tab. The API Console tab allows you to test all of the calls in the API individually through the **Try Me** option on each call.

4 Controlling Access to the APIs through the Access Manager Roles

The Secure API Manager roles 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 and Access Manager to use the Access Manager roles and scopes. Use the following information to understand this process and how to configure Secure API Manager and Access Manager.

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

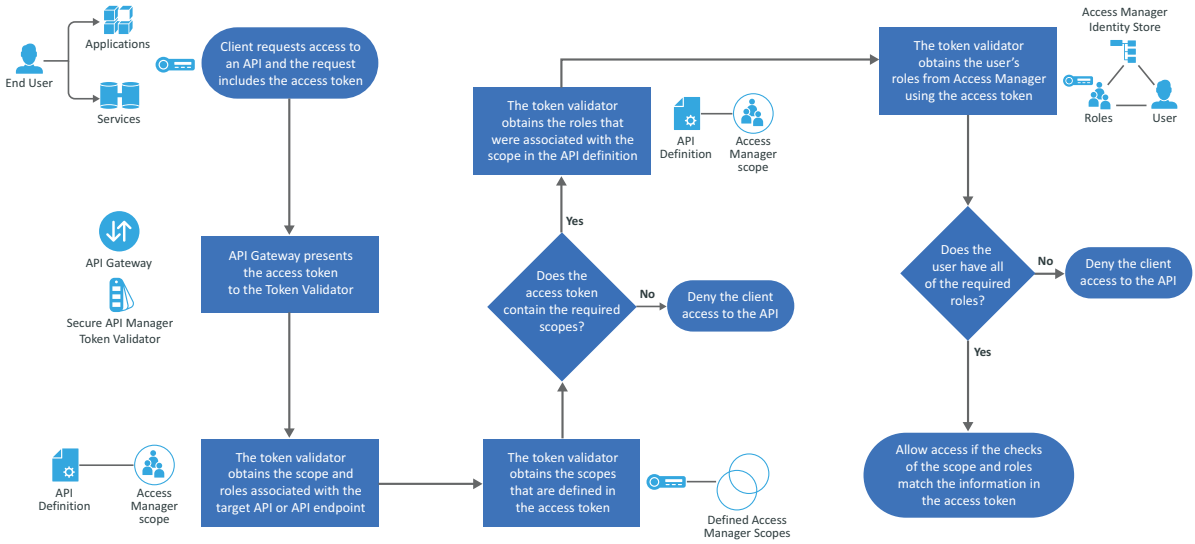
Secure API Manager allows you to make APIs accessible to the public or limits access to the APIs. To make APIs available to the public means that anyone who knows the full path to the API can call the API. Secure API Manager integrates with Access Manager to provide OAuth2 tokens for any requests to the APIs. This is a separate point of integration with Access Manager than controlling specific users to be able to access the APIs. For more information about OAuth2 authorizations, see "How Secure API Manager Works" in the *NetIQ Secure API Manager 1.0 Installation Guide*.

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 must create one scope for every API. 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. Otherwise, it is a one-to-one relationship between the API and the Access Manager scope. You cannot use the same scope for multiple APIs or for multiple API endpoints associated with different APIs. The Publisher displays a error message if you try to associate a scope already in use. 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*



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 and 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 require 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 all 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

To allow Secure API Manager to read and access the Access Manager roles and scopes is a two-step process. First, you must create an attribute map in Access Manager. Next, you must create a scope for each API or for a specific API that you want to protect. You perform the first step once but you must create a scope for each API or specific API endpoints that you want to protect.

Creating an Attribute Map in Access Manager for Secure API Manager

To control access to the APIs that are available in the Store you must create an attribute map. 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 Each API in Access Manager

You must create a scope for each API or specific API endpoints that you want to protect with Access Manager. This is a one-to-one relationship. You cannot reuse a 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 name of the identity server that you associated with Secure API Manager during the deployment. For more information, see "Completing the Integration Between Secure API Manager and Access Manager" in the *NetIQ Secure API Manager 1.0 Installation Guide*.
2. Click the **OAuth and OpenID Connect** tab.
3. In the menu, click **Resource Servers**.
4. Create a new resource server for each API or each specific API endpoint as follows:
   - **4a** Click **New.**
   - **4b** Specify a name for the resource server that represents s the API or the specific API endpoint 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 each API or specific API endpoint you want to protect.

5 Add a scope to each resource server for the API or the specific API endpoint 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 API or the specific API endpoint.

5e Click **User Attributes**, then select the **Require user permission** option. Ensure that you select this option or 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 25.

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 Log in to the appliance administration console as `vaadmin`.

`https://appliance-dns-name:9443`

7b Click **Deployment Manager**.

7c Click the **Access Manager Integration** tab.

7d Click **Save**.

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 in Access Manager. For more information, see “Role Policies” in the *NetIQ Access Manager 4.5 Administration Guide*.

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 only a single scope with a single API, all of the endpoints in an API, or with a specific API endpoint. Attempting to reuse a scope that is already in use causes the Publisher to display an error message. 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.

Every API has at least one API endpoint. You must assign the scope to the API, which does not limit any access to the API. Next, you must assign the scope to the API endpoint or to multiple API endpoints to control access to the API.

To associate the scope with the API or specific API endpoints:

1. Log in to the Publisher using an administrative account.

   `https://lifecycle-manager-dns-name:9444/publisher`

   The *dns-name* is the fully qualified hostname of the appliance running the Lifecycle Manager component.
2. On an API to which you want restrict access, click **Edit**.
3. Scroll to the end of the page, then click **Next: Implement**.
4. Scroll to the end of the page, then click **Next: Manage**.
5. Assign a scope to the API using the following steps:
   - **5a** Under **Resources**, click **Add Scopes**.
   - **5b** In the **Scope Key** field, select the associated Access Manager scope from the drop-down list.
   - **5c** Specify a display name for the scope that appears in the Publisher.
   - **5d** In the **Roles** field, select the Access Manager role or roles that controls access to this API or the additional API endpoints.
   - **5e** Specify a description for this list of roles. Assigning a description to the list of roles helps you to know what roles are in this list.
   - **5f** Click **Add Scope**.
6. Assign a scope to the API endpoint or assign the scope to multiple API endpoints, using the following steps:
   - **6a** Under **Resource**, on the single API endpoint, click **+scope** to limit the access to the API.
   - **6b** Click the drop-down menu, then select the appropriate scope.
   - **6c** Click the check mark to save the assignment and limit access to this API endpoint.
   - **6d** (Conditional) If you have multiple API endpoints, and you are using the same scope, repeatStep 6a through Step 6c for each API endpoint.
   - **6e** (Conditional) If you want to have different roles control access to different API endpoints in the same API, repeat Step 5a through Step 5e to add the new scope, then repeat Step 6a through Step 6c to assign the different scopes to the API endpoints.
7. Click **Save** to continue making changes to the API at a later time or click **Save and Publish** to make the API and the API endpoints available for users to access.

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.

5 Managing Connections to the APIs with Throttling Policies

Secure API Manager allows you to control the number of calls and authorizations to the APIs for a certain period through throttling policies. You might want to use throttling policies for several different reasons:

- Protect the APIs from denial of service (DOS) attacks
- Control traffic due to infrastructure availability
- Provide APIs, applications, and resources to users at different service levels

There are many different components involved when a user accesses an application or service that makes a call to an API stored in Secure API Manager.

- The back-end services
- The APIs in the API Gateway that act as proxies to the back-end services
- The applications and resources that use the APIs in the API Gateway

Several throttling policies are available for the different components and you create the different throttling policies for different reasons. Use the following information to determine when to use the different throttling policies and how to create them.

- “Back-End Services Throttling Policy” on page 29
- “API Throttling Policies” on page 30
- “Application Throttling Policies” on page 32

Back-End Services Throttling Policy

Use the following information to determine why and when you would use a back-end service throttling policy and how to configure the policy.

- “Understanding Why and When to Use a Back-End Services Throttling Policy” on page 29
- “Configuring a Back-End Services Throttling Policy” on page 30

Understanding Why and When to Use a Back-End Services Throttling Policy

The APIs stored in the API Gateway act as proxies to the back-end services in your company. These back-end services are the services within your company or your company’s services hosted in the cloud. These back-end services have a physical capacity limit of the load that they can process. As the number of APIs in the API Gateway increase, the number of applications in your environment increase which in turn increases the load on these back-end services.

Secure API Manager allows you to create **defined controls (subscription limits)** that limit access to the APIs. Even though each API might not surpass the subscription limits, this increases the total load on the back-end services and you might overload these back-end services.

Secure API Manager provides a back-end service throttling policy through the **Maximum Backend Throughput** setting when you create an API or edit an API. This setting provides a defined limit of how many requests can be made per API over a certain amount of time to the back-end service. Secure API Manager maintains this setting when evaluating the maximum back-end throughput that is shared across all nodes of the API Gateway cluster and applies across all users using any application that accesses that particular API.

By default, this setting is set to **Unlimited** to allow unlimited connections to a specific back-end service. You would change this setting only if you knew that a specific back-end service could not handle the requests from multiple APIs.

Configuring a Back-End Services Throttling Policy

The back-end services throttling policy is a **Throttling Setting** named **Maximum Backend Throughput** that you define when you create or edit an API. You must configure this setting for each API that connects to the same back-end service that might not be able to handle the load. This allows Secure API Manager to uniformly affect the specific back-end service.

By default this setting is set to **Unlimited** to allow unlimited connections to the back-end service. If you want to control how many requests APIs can make to a specific back-end service over time, you must change this setting.

To change the Maximum Backend Throughput setting:

1. Log in to the Publisher using an administrative account.

   `https://lifecycle-manager-dns-name:9444/publisher`

   The *dns-name* is the fully qualified hostname of the appliance running the Lifecycle Manager component.
2. On an API whose access a specific back-end service you want to limit, click **Edit**.
3. Scroll to the end of the page, then click **Next: Implement**.
4. Scroll to the end of the page, then click **Next: Manage**.
5. Under **Throttling Settings > Maximum Backend Throughput**, click **Specify**.
6. Specify the length of time in seconds for the throughput per second to the back-end service.
7. Click **Save and Publish**.
8. Repeat Step 2 through Step 7 for each API that uses the back-end service where you want to limit the requests.

API Throttling Policies

Use the following information to determine why and when you would use an API throttling policy and how to configure the policy. The API throttling policies are the subscription tier settings.

- "Understanding API Throttling Policies" on page 31
- "Using the API Throttling Policies" on page 31

Understanding API Throttling Policies

Secure API Manager uses API throttling policies to limit the number of successful requests to the API through **subscription tiers**. Developers set the subscription tiers on the API when they create new APIs. By associating the tiers to the API, you limit the number of requests that come through the API Gateway for each API. This allows you to give specific APIs a higher access rate than other APIs.

By default, Secure API Manager contains four tiers:

- **Bronze:** Allows 1000 requests per second
- **Silver:** Allows 2000 requests per second
- **Gold:** Allows 5000 requests per second
- **Unlimited:** Allows unlimited requests

It is important to note that even though you might assign an API to the **Unlimited** tier, if you set the **Maximum Backend Throughput** option to something other than **Unlimited**, the **Maximum Backend Throughput** setting takes precedence over the tiers option.

Using the API Throttling Policies

The API throttling policy is a **Throttling Setting** named **Subscription Tiers** that you define when you create an API or edit. You must configure this setting for each API that you create or import.

You must select one of the available options to finish creating or importing an API.

To set the Subscription Tiers setting:

1. Log in to the Publisher using an administrative account.

   `https://lifecycle-manager-dns-name:9444/publisher`

   The *dns-name* is the fully qualified hostname of the appliance running the Lifecycle Manager component.
2. On an API for which you want to change the number of calls to an API per minute, click **Edit**.
3. Scroll to the end of the page, then click **Next: Implement**.
4. Scroll to the end of the page, then click **Next: Manage**.
5. Under **Throttling Settings > Subscription Tiers**, select the appropriate level of calls for your API.
6. Click **Save and Publish**.
7. Repeat Step 2 through Step 6 for each API that uses the back-end service where you want to limit the requests.

Application Throttling Policies

Use the following information to determine why and when you would use an application throttling policy and how to configure the policy.

Understanding Application Throttling Policies

Secure API Manager creates secure communication channels between the APIs and the applications or services using tokens. This process ensures that no unauthorized requests can access the APIs. This also allows you to use a single access token to invoke a collections of APIs and to subscribe to one API multiple times with different service levels.

To use an API or a group of APIs you must create an application for the APIs in the Store. When you create an application, you define an API request per access token. The application shares the quota to all of the APIs you assign to the application. The application throttling options allows you set a maximum number of request to the application within a defined time period.

There are four application throttling policies:

- **Unlimited:** Allows unlimited requests for this application
- **10PerMin:** Allows 10 requests per minute for this application
- **20PerMin:** Allows 20 requests per minute for this application
- **50PerMin:** Allows 50 requests per minute for this application

Using the Application Throttling Policies

You use the application throttling policies to limit the number of request accessing the application. You add the application throttling policy to the application.

To set the Subscription Tiers setting:

1. Log in to the Store with the account your administrator gave you.

   `https://lifecycle-manager-dns-name:9444/store`

   The *dns-name* is the fully qualified hostname of the appliance running the Lifecycle Manager component.
2. Click the **Applications** tab.
3. Click the name of the application that you want to enable the application throttling policies.
4. Click **Edit** (pencil icon).
5. In the **Per Token Quote** setting, select the appropriate rate for your application.
6. Click **Update**.
7. Repeat Step 2 through Step 6 for each API that uses the back-end service where you want to limit the requests.

6 Managing Documentation for the APIs

Providing documentation for APIs helps make the APIs easier to use. It saves time and issues by providing detailed instructions on what the API does and how to implement the API without having to contact the developer of the API.

Adding your APIs to Secure API Manager provides a single repository, in your IT environment, where everyone in the company knows where to go to access any available APIs. Secure API Manager provides the ability to add documentation to the APIs in the Store. You can add, edit, and delete the documentation of each API.

Adding Documentation for the APIs

Secure API Manager allows you to add inline documentation, add a URL where the users access the documentation, or upload a file that contains the documentation. It also allows you to categorize the documentation that you are adding. You can add procedural (how to) documentation, samples, and SDK documentation, as well as provide URL links to support and developer forums.

Adding Inline Documentation for the APIs

Secure API Manager provides an embedded text editor for you to use when you add inline documentation. The text editor provides similar functionality to what you would find when you compose an email.

To add inline documentation:

1. Log in to the Publisher using an administrator account.

   `https://lifecycle-manager-dns-name:9444/publisher`

2. Find the API where you want to add the documentation.
3. Click the name of the appropriate API, then click the **Docs** tab.
4. Click **Add New Document**, then use the following information to add the inline documentation.

   Name

   Specify the name of this documentation that appears in a table on the **Docs** tab. You can add one or more documentation items an API.

   Summary

   Provide a summary of this documentation item. This information appears in the table on the **Docs** tab of the API.

Type

Select the type of documentation that you are adding. Select **How To**, **Samples**, **Public Forum**, **Support Forum**, or **Other**. If you select **Other**, you must add an additional summary of the documentation type in the field provided.

Source

For this example, select **Inline** as the documentation source.

5 Click **Add Document** to add an entry to the documentation table on the **Docs** tab of the API.

6 Click **Edit Content** on the new documentation entry in the table on the **Docs** tab.

Secure API Manager launches an embedded text editor in a separate window.

7 Add or create the appropriate documentation in the text editor. This is the inline documentation that the users see on the API.

8 Click **Save and Close**.

9 (Conditional) If you want to create additional inline documentation types, repeat Step 4 through Step 8 for each additional documentation type.

10 To see the documentation as subscribers see it:

10a Log in to the Store.

`https://lifecycle-manager-dns-name:9444/store`

10b Click the name of the appropriate API, then click the **Documentation** tab.

The Store lists all of the documentation by type.

10c Expand the documentation type, then click **View Content** to see the documentation.

Adding a URL that Contains Documentation for the APIs

You can add a URL to the API that provides a link to the documentation for the API on a Swagger server or for documentation posted online. You can also specify the URLs for your public and support forums for the developers. If you select **Public Forum** or **Support Forum** as the documentation **Type**, Secure API Manager automatically changes the Source to **URL**.

To add a URL for the documentation:

1 Log in to the Publisher using an administrator account.

`https://lifecycle-manager-dns-name:9444/publisher`

2 Click the name of the appropriate API, then click the **Docs** tab.

3 Click **Add New Document**, then use the following information to add the URL for the documentation.

Name

Specify the name of this documentation that appears in a table on the **Docs** tab. You can add one or more documentation items to an API.

Summary

Provide a summary of this documentation item. This information appears in the table on the **Docs** tab of the API.

Type

Select the type of documentation that you are adding. Select **How To**, **Samples**, **Public Forum**, **Support Forum**, or **Other**. If you select **Other**, you must add an additional summary of the documentation type in the field provided.

Source

For this example, select **URL** as the documentation source, then specify the URL in the new field that appears.

4 Click **Add Document** to add an entry to the documentation table on the **Docs** tab of the API.

5 Click **Save and Close**.

6 (Conditional) If you want to add additional URLs for the documentation, repeat Step 3 through Step 5 for each additional URL.

7 To see the documentation as subscribers see it:

7a Log in to the Store.

`https://lifecycle-manager-dns-name:9444/store`

7b Click the name of the appropriate API, then click the **Documentation** tab.

The Store lists all of the documentation by type.

7c Expand the documentation type, then click **View Content** to see the documentation.

Uploading a File that Contains Documentation for the APIs

If your documentation for the API already exists in a file, Secure API Manager allows you to upload that file to the API. The document is stored on the API and the developers that use the API can download the file. Secure API Manager supports the following file types of any size:

- `.doc`
- `.html`
- `.pdf`
- `.txt`

To upload a file that contains the documentation:

1 Log in to the Publisher using an administrator account.

`https://lifecycle-manager-dns-name:9444/publisher`

2 Click the name of the appropriate API, then click the **Docs** tab.

3 Click **Add New Document**, then use the following information to upload the file that contains the documentation.

Name

Specify the name of this documentation that appears in a table on the **Docs** tab. You can one or more documentation items added to an API.

Summary

Provide a summary of this documentation item. This information appears in the table on the **Docs** tab of the API.

Type

Select the type of documentation that you are adding. Select **How To**, **Samples**, **Public Forum**, **Support Forum**, or **Other**. If you select **Other**, you must add an additional summary of the documentation type in the field provided.

Source

For this example, select **File** as the documentation source, then browse to and select the appropriate file that contains the documentation.

4 Click **Add Document** to add an entry to the documentation table on the **Docs** tab of the API.

5 Click **Save and Close**.

6 (Conditional) If you want to add additional URLs for the documentation, repeat Step 3 through Step 5 for each additional URL.

7 To see the documentation as subscribers see it:

7a Log in to the Store.

`https://lifecycle-manager-dns-name:9444/store`

7b Click the name of the appropriate API, then click the **Documentation** tab.

The Store lists all of the documentation by type.

7c Expand the documentation type, then click **View Content** to see the documentation.

Editing Documentation for the APIs

Secure API Manager allows you to edit each documentation entry added to the API. This is how you edit the inline documentation, change the URLs, or upload an updated file that contains the documentation.

To edit the documentation for an API:

1 Log in to the Publisher using an administrator account.

`https://lifecycle-manager-dns-name:9444/publisher`

2 Click the name of the appropriate API, then click the **Docs** tab.

3 In the documentation table, click **Edit Content** on the appropriate documentation item.

4 Change the appropriate information for the documentation source.

Inline

The Publisher launches the embedded text editor. Make the appropriate changes to the inline documentation, then click **Save and Close**.

URL

Change the URL to the new URL.

File

Upload a new documentation file by clicking **Browse** and selecting the new file.

5 Click **Save and Close** to save the changes.

Deleting Documentation for the APIs

Secure API Manager allows you to delete any documentation items from the APIs as long as you are an administrator.

To delete the documentation for an API:

1 Log in to the Publisher using an administrator account.

`https://lifecycle-manager-dns-name:9444/publisher`

2 Click the name of the appropriate API, then click the **Docs** tab.

3 In the documentation table, click **Delete** on the appropriate documentation item, then click **Yes**.

7 Using APIs

To use APIs you must have access tokens to authenticate to and use the API. Instead of generating an access token each time you use an API, Secure API Manager generates a single access token for each API you use by having you create an application. The application allows you to subscribe to one or more APIs. Secure API Manager generates the access token through the application.

Secure API Manager provides a Swagger interface that allows you to test and invoke the APIs to ensure that they work. You access the API Console through each available API in the Store. The API Console helps you to test and invoke the APIs through the application.

To use the APIs, you must ensure that you have created an application that calls the appropriate API or APIs that you want to use. You can use more than one API in an application. To use the APIs:

1. Create an application on the **Applications** tab in the Store. For more information, see “Creating Applications” on page 37.
2. Access the application through the appropriate API, then subscribe to the application. For more information, see “Subscribing to APIs” on page 39.
3. Test or invoke the API through the API Console. Each API type requires different steps to test or invoke the API. For more information, see “Invoking or Testing the APIs” on page 40.

Use the following information to subscribe to and use the appropriate API type for your project.

Managing Applications

Secure API Manager requires you to create an application in the Store to use the APIs. You can have one or more APIs in an application. Secure API Manager uses applications that you create in the Store to manage subscriptions, throttling, and access to the APIs. It also allows you to combine multiple APIs to provide additional functionality through the API.

Creating Applications

You create an application to be able to use one or more APIs that are available in the Store. You or someone else must have created the API and published the API in the Store for you to use it. By default, Secure API Manager creates a DefaultApplication with unlimited access for you to use.

1 Log in to the Store using the account your Secure API Manager administrator gave you.

`https://lifecycle-manager-dns-name:9444/store`

2 Click the **Applications** tab, then click **Add Application**.

3 Use the following information to create the application:

Name

Specify a name for your application. It is the name that appears in the table on the **Applications** tab.

Per Token Quota

Select the quota of access tokens that Secure API Manager allows for the API. Secure API Manager shares the allocated quota among all the subscribed APIs of the application. You can have:

- **Unlimited**
- **50PerMin**
- **20PerMin**
- **10PerMin**

Description

Specify a description of what the application does. This description appears in the table on the **Applications** tab.

Token Type

Secure API Manager only supports OAuth tokens at this time.

4 Click **Add** to create the application.

5 View the details of the new application, then close the window.

To add APIs to the application, you must access the appropriate API and subscribe to the application. For more information, see “Subscribing to APIs” on page 39.

Editing Applications

You can edit applications any time you need to make changes to them.

1 Log in to the Store using the account your Secure API Manager administrator gave you.

`https://lifecycle-manager-dns-name:9444/store`

2 Click the **Applications** tab.

3 Find the appropriate application in the table, then click the name of the application.

4 Click the **Edit** icon next to the name of the application.

5 Make the appropriate changes, then click **Update**.

Deleting Applications

You can delete applications at any time.

1 Log in to the Store using the account your Secure API Manager administrator gave you.

`https://lifecycle-manager-dns-name:9444/store`

2 Click the **Applications** tab.

3 Find the appropriate application in the table.

4 Click the **Delete** icon in the row for the application, then confirm the deletion.

Managing Subscriptions

Secure API Manager allows you to subscribe to applications that contain the API or APIs you want to use. It also allows you to remove subscriptions if you no longer want to use the API or the API is deprecated. Also, if you want to delete an API, you must remove all subscriptions from the API before you can delete it.

Subscribing to APIs

To use any of the published APIs in the Store, you first must create an application and subscribe to the application through the appropriate API. Subscribing to the API allows you to combine multiple APIs to create applications with new functionality. Subscribing to the application also allows you to receive the access token for the API to authenticate you to the API so that you can invoke the API. If you have used an API in multiple applications, you can generate one authentication token to use in all of the applications.

To subscribe to an API:

1. Log in to the Store using the URL you received from your administrator.

   `https://lifecycle-manager-dns-name:9444/store`

2. Click the **Applications** tab.
3. Create a new application or ensure that the application you want to use already exists. For more information, see "Creating Applications" on page 37.
4. Click on the **APIs** tab, then click the name of the API you want to subscribe to and use.
5. In the **Applications** field, select the application you want to use.
6. In the **Tiers** field, select the appropriate tier to use for this subscription, then click **Subscribe**.
7. When prompted, click **View Subscriptions**.
8. Click the **Production Key** tab, then use the following information to define the appropriate information for your environment:

   Grant Type

   Select the **Client Credentials** grant type to use with the selected API. The grant types generate the access tokens for the API.

   IMPORTANT: By default, Secure API Manager uses the **Client Credentials** grant type to generate the access token. Ensure that you always select the **Client Credentials** grant type.

   Callback URL

   (Conditional) Specify a callback URL that sends a callback to a specific server or program soon after Secure API Manager sends your application request.

   Scopes

   (Conditional) Select the appropriate scope to limit the access for the API. For more information, see Chapter 4, "Controlling Access to the APIs through the Access Manager Roles," on page 23.

Access token validity period

Specify how long the access token is valid in seconds. The default value is 3600 seconds which equals one hour. If you do not want the access token to expire, specify -1.

9 Click **Generate keys**.

After you subscribe to the API, you are able to use the API with the correct access token. The steps to use each API type are different. For more information, see “Invoking or Testing the APIs” on page 40.

Deleting Subscriptions

If you are no longer using an API and want to delete it or deprecate it, you must remove all subscriptions from the API before you can delete the API.

1 Log in to the Store using the URL you received from your administrator.

`https://lifecycle-manager-dns-name:9444/store`

2 Click the **Applications** tab.

3 Click the name of the appropriate application.

4 On the overview page of the application, click the **Subscriptions** tab.

5 Click **Unsubscribe**, then verify that you want to unsubscribe.

Invoking or Testing the APIs

Secure API Manager provides an integrated Swagger editor that allows you to test and invoke the APIs before users or services access them. This ensures that the APIs work correctly. The steps to test and invoke the APIs are different depending on the type of API subscriptions you have.

- “Invoking and Testing the REST APIs” on page 40
- “Invoking and Testing the SOAP APIs” on page 41
- “Invoking and Testing the WebSocket APIs” on page 42

Invoking and Testing the REST APIs

Secure API Manager allows you to test and invoke the REST APIs through the built-in Swagger tools. To invoke and test the REST API, you must know and understand REST and the Swagger tools. For more information, see “Required Knowledge” on page 8.

1 Log in to the Store using the URL you received from your administrator.

`https://lifecycle-manager-dns-name:9444/store`

2 Ensure that you have created an application and that you have subscribed to the appropriate API or APIs. For more information, see “Creating Applications” on page 37 and “Subscribing to APIs” on page 39.

3 Generate an access token for the API.

3a On the **Applications** tab, click the name of the appropriate application.

3b On the overview page, click the **Product Keys** tab, then specify the appropriate information for your API.

3c Click **Generate keys**.

3d Click **Close** to close the overview page.

4 Click the **APIs** tab, then click the name of the appropriate API.

5 Click the **API Console** tab.

6 Click the appropriate REST call to expand it, then click **Try it out**.

7 Click **Execute**, then view the response.

8 Ensure that the response contains the expected response.

Invoking and Testing the SOAP APIs

Secure API Manager allows you to test and invoke the SOAP APIs through the built-in Swagger tools. To invoke and test the SOAP API, you must know and understand SOAP and the Swagger tools. For more information, see “Required Knowledge” on page 8.

1 Log in to the Store using the URL you received from your administrator.

`https://lifecycle-manager-dns-name:9444/store`

2 Ensure that you have created an application and that you have subscribed to the SOAP API or APIs. For more information, see “Creating Applications” on page 37 and “Subscribing to APIs” on page 39.

3 Generate an access token for the API.

3a On the **Applications** tab, click the name of the appropriate application.

3b Click the **Product Keys** tab, then specify the appropriate information for your API.

3c Click **Generate keys**.

or

(Conditional) If you have generated keys before, click **Regenerate**.

By default, access tokens expire after one hour.

3d Click **Copy** to copy the access token.

NOTE: Ensure that you have the latest flash plug-in installed in your browser to have the **Copy** button work for the access token in the Store.

3e Click **Close** to close the details page.

4 Invoke the SOAP API with the embedded Swagger API Console.

4a Click the name of the SOAP API, then click the **API Console** tab.

4b Expand the POST method, then click **Try it out**.

4c Enter the SOAP Request and SOAP Action information for your API.

4d Click **Execute**.

4e View the response of the SOAP API in the console.

You can also invoke the SOAP API using any SOAP client. You must enter the WSDL of the SOAP API and generate an access token that you copy and paste in the Authorization header of the SOAP client.

Invoking and Testing the WebSocket APIs

Secure API Manager allows you to test and invoke the WebSocket APIs. You must have a WebSocket client that works with the API to invoke and test the API. You must also know and understand the WebSocket protocol to invoke and test the API. For more information, see “Required Knowledge” on page 8.

1 Log in to the Store using the URL you received from your administrator.

`https://lifecycle-manager-dns-name:9444/store`

2 Ensure that you have created an application and that you have subscribed to the WebSocket API or APIs. For more information, see “Creating Applications” on page 37 and “Subscribing to APIs” on page 39.

3 Generate an access token for the API.

3a On the **Applications** tab, click the name of the appropriate application.

3b Click the **Product Keys** tab, then specify the appropriate information for your API.

3c Click **Generate keys**.

or

(Conditional) If you have generated keys before, click **Regenerate**.

By default, access tokens expire after one hour.

3d Click **Copy** to copy the access token.

NOTE: Ensure that you have the latest flash plug-in installed in your browser to have the **Copy** button work for the access token in the Store.

3e Click **Close** to close the overview page.

To invoke the WebSocket API you can use any WebSocket client. Use the instructions that come with the WebSocket client to invoke the WebSocket API. You use the access token from the Store as the **Authorization Bearer token** in the WebSocket client. Also, you use the **Production Endpoint** or **Sandbox Endpoint** from the API as the String URL that you define in the WebSocket client.

8 Managing the Lifecycle of an API

As you create APIs you want to be able to test, temporarily make an API unavailable, or deprecate the API as it becomes outdated. Secure API Manager provides a tool in the Publisher that allows you to view and manage the lifecycle of an API. This tool shows the six possible states of an API: created, prototyped, published, blocked, deprecated, and retired.

You can view the lifecycle of the API and manage the different stages through the Publisher. When you view an API, there is a **Lifecycle** tab that you use to manage the different lifecycle stages. Use the following information to help you manage your APIs with the lifecycle tool.

Creating a Prototype API and Publishing an API

Secure API Manager automatically assigns three of the lifecycle stages. Those stages are created, published, and prototyped. These stages occur while you creating the APIs. You create a prototype API to provide early access and test the API.

When you create a REST API, the Publisher gives you the choice to deploy the REST API as a prototype or to import an existing REST API. If you have a SOAP API or a WebSocket API, you can define sandbox endpoints to create a prototype an API. For more information see, Chapter 3, "Creating and Publishing APIs," on page 11.

Viewing the Details of an API

You can view the details of an API through both the Publisher and the Store. To view the details of an API, you click the name of the API and the Overview page appears. You manage your subscriptions and the lifecycle of an AP, as well as other tasks, through the Overview page.

Changing the Version of an API

You change the version of an API if you want to provide new functionality in the API, change who can access the API, or change throttling tiers, and so forth. By creating a new version of an API, you can deploy a prototype of the modified API and test the changes to ensure that they function as expected.

We recommend that you change the version of an API and deploy a prototype of the modified API instead of editing an API and updating the API while you have subscriptions assigned to the API.

IMPORTANT: Never modify an API that has subscriptions assigned to it. Doing so can cause issues with the API.

To change the version of an API:

1. Log in to the Publisher using the account your Secure API Manager administrator gave you.

   `https://lifecycle-manager-dns-name:9444/publisher`

2. Click the name of the appropriate API to view the details of the API.
3. In the upper-right corner, click **Create a New Version** (+).
4. Specify the new version of the API.
5. (Conditional) If you want this API to become the default URL that the applications use to the access the API, select **Make this the default version**.
6. Click **Done,** then click **OK**.
7. Click the **Edit** icon on the new API.
8. Make the appropriate changes to the new API.
9. Click **Save**.
10. Click the **APIs** tab, then click the name of the API.
11. Click the **Lifecycle** tab.
12. (Conditional) To test the API, click **Deploy as a Prototype**.
13. (Conditional) To publish the API to the Store select one or more of the following options:
    - **13a** (Conditional) If you want all users subscribed to the prior version of the API to keep their subscription, ensure that you deselect **Require re-subscription when publish the API**.
    - **13b** (Conditional) If you want to deprecate the prior version of the API, select **Deprecate old version after publish the API**. For more information, see “Deprecating APIs” on page 44.

      IMPORTANT: You can move an API to the retired state only after you deprecate an API. You cannot move a deprecated API back to a published state.

    - **13c** Click **Publish** to publish the API and make it available for use.

Deprecating APIs

Secure API Manager allows you to deprecate APIs so that no new subscriptions can be made to the deprecated APIs. People that currently have subscriptions to the deprecated APIs still see the deprecated APIs in the Store. The deprecated APIs can stay in this state until you are ready to retire them.

You can depreciate an API when you publish a new version of the API or after you have published the API. For more information on how to deprecate an API when you create a new version, see “Changing the Version of an API” on page 43.

To deprecate the API after you have published a new version:

1. Log in to the Publisher using the account your Secure API Manager administrator gave you.

   `https://lifecycle-manager-dns-name:9444/publisher`

2. Click the name of the API you want to deprecate to see the details of the API.

3 Click the **Lifecycle** tab, then click **Deprecate** and read the warning.

IMPORTANT: You can move an API to the retired state only after you deprecate an API. You cannot move a deprecated API back to a published state.

4 Click **Yes** to finish the process.

You can still view the API in the Publisher. You can see the state of each API when you first log into the Publisher.

Retiring APIs

Secure API Manager allows you to retire APIs after you have deprecated the APIs. When you retire an API, Secure API Manager removes the API from the Store. Secure API Manager keeps the API in the Publisher to keep a record of the older APIs.

After you deprecate an API, the next and only option is to retire the API.

1 Log in to the Publisher using the account your Secure API Manager administrator gave you.

`https://lifecycle-manager-dns-name:9444/publisher`

2 Click the name of the deprecated API you want to retire to see the details of the API.

3 Click the **Lifecycle** tab, then click **Retire**.

Blocking APIs

Sometimes you might have a need to not allow anyone to use a published API but you do not want to deprecate the API. You might not want someone to use an API due to a security issue or a bug. You can block a published API so that it cannot be used until the issue is resolved.

When you block a published API, the lifecycle tool moves the API back to the created state. You can make the necessary changes to the API and then republish the API.

1 Log in to the Publisher using the account your Secure API Manager administrator gave you.

`https://lifecycle-manager-dns-name:9444/publisher`

2 Click the name of the published API you want to block.

3 Click the **Lifecycle** tab, then click **Block**.

4 Make the appropriate changes to the API or keep the API unavailable for as long as necessary.

5 When you are ready to make the API available again, in the Publisher, click the blocked API.

6 Click the **Lifecycle** tab, then click **Re-Publish** to re-publish the API.

9 Using Analytics

Secure API Manager provides an analytics tool that allows you to run reports to see the usage of the APIs. You can see which APIs customers access the most, whether customers are accessing the APIs more than what is allowed, to see the throughput for the APIs, to see if the throttling must be upgraded, and so forth.

The global administrator must deploy and configure Analytics for the reports to appear in the Publisher and the Store. You can view the reports in the Publisher and the Store.

Viewing the Analytics Reports in the Publisher

You can view detailed reports generated by the Analytics component in the Publisher. The Publisher divides the reports into three categories: APIs, Applications, and Subscriptions.

1. Log in to the Publisher using the account your Secure API Manager administrator gave you.

   `https://lifecycle-manager-dns-name:9444/publisher`

2. Click the **Analytics** tab.

   NOTE: If the **Analytics** tab does not display any reports, the global administrator has not deployed or enabled the Analytics component.

3. Select the appropriate report you want to view.
4. Select what you want to view and the appropriate date range for the report. The UI automatically updates when you make a change.

NOTE: For this release, Secure API Manager does not support managed alerts.

Viewing the Analytics Reports in the Store

The Analytics reports available in the Store contain information about API usage, the top users accessing the APIs, the resource usage, and whether there are any faulty invocations of the subscribed APIs.

1. Log in to the Store using the account your Secure API Manager administrator gave you.

   `https://lifecycle-manager-dns-name:9444/store`

2. Click the **Analytics** tab, then select the appropriate report.

NOTE: If the **Analytics** tab does not display any reports, the global administrator has not deployed or enabled the Analytics component.

3 Select what you want to view and the appropriate date range for the report. The UI automatically updates when you make a change.

NOTE: For this release, Secure API Manager does not support managed alerts.