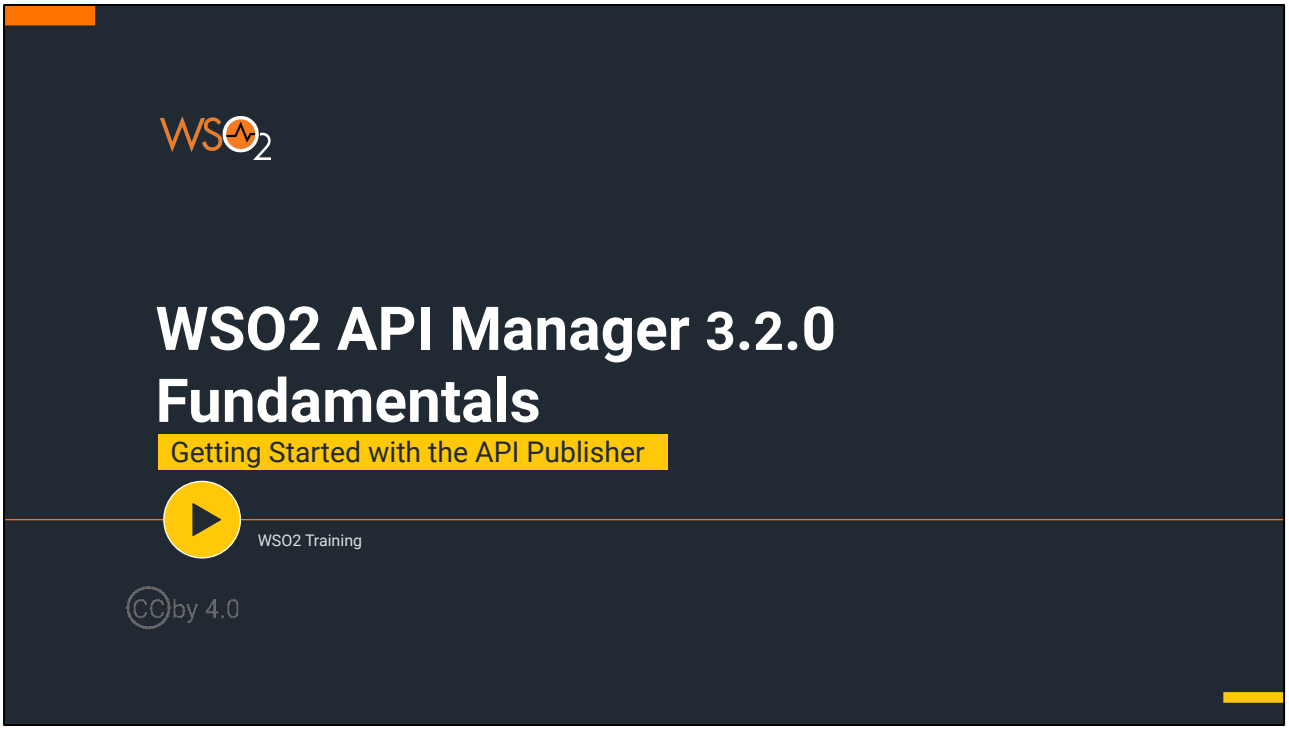
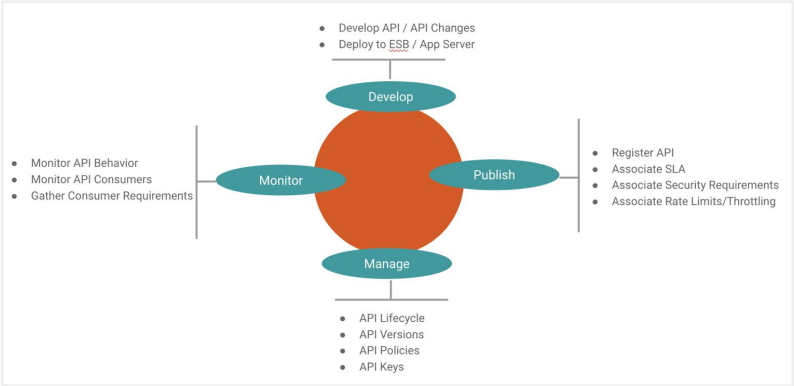

WSO2
WSO2 API Manager 3.2.0 Fundamentals
Getting Started with the API Publisher
WSO2 Training
CC by 4.0

What is the API Publisher Portal?

- React Web Application
- Story of APIs start here.
- Objectives
  - Design APIs
  - Publish APIs
  - Manage APIs
  - Monitor APIs



Creating an API

This section covers,

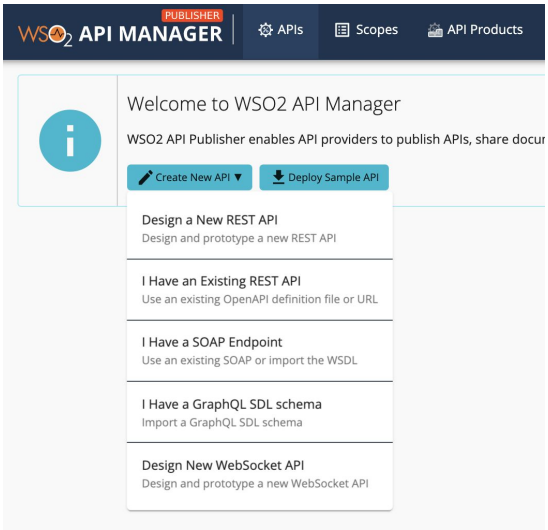
1. How to create a REST API from scratch
2. Complete UI walkthrough
3. Create other API Types

Let's try it out!

Creating and publishing an API through API Manager Publisher

Ways of Creating APIs

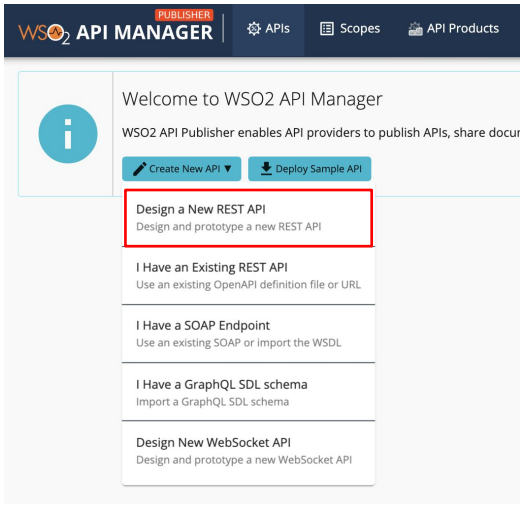
- Design a new REST API
- REST API from OpenAPI definition
- REST API from SOAP backend
- SOAP API as a REST service
- GraphQL API from SDL
- Design new Websocket API



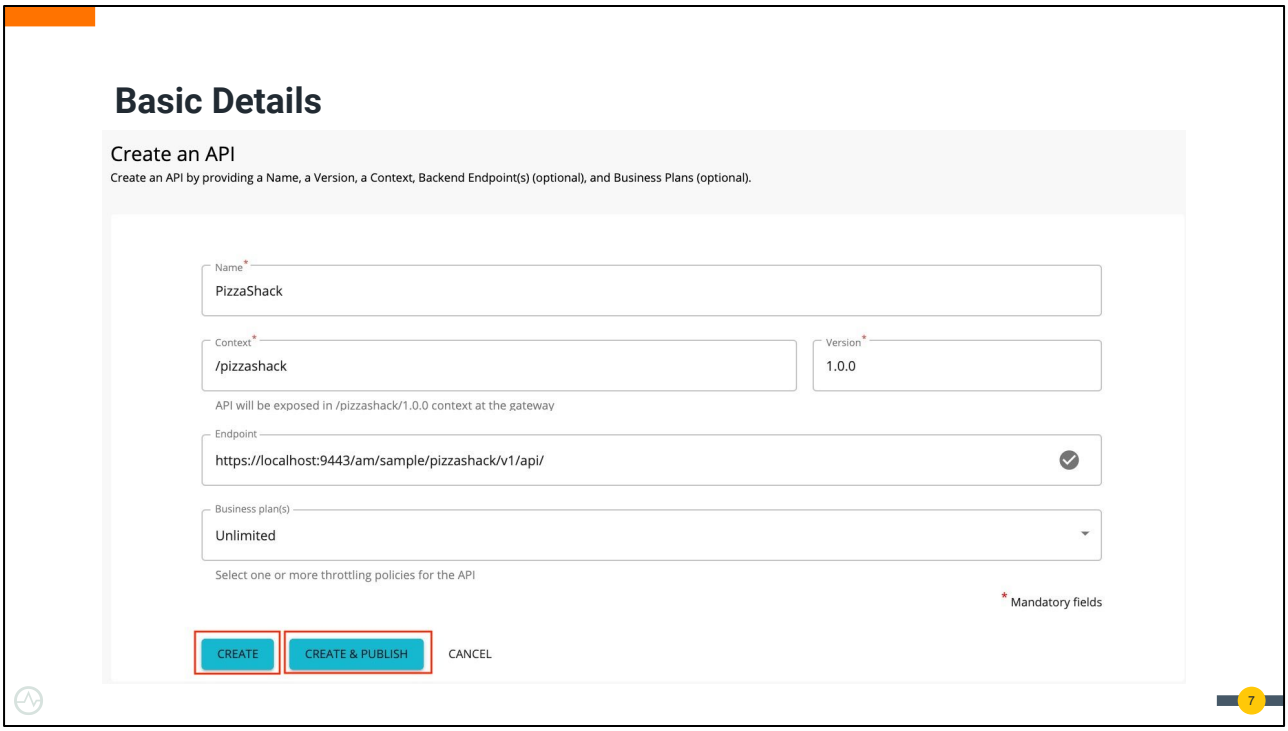
[Creating APIs]

Design a New REST API

Select the **Design a New REST API** option



This section includes how to create a REST API from scratch and a walkthrough of the API Publisher UI to identify components and their functions.

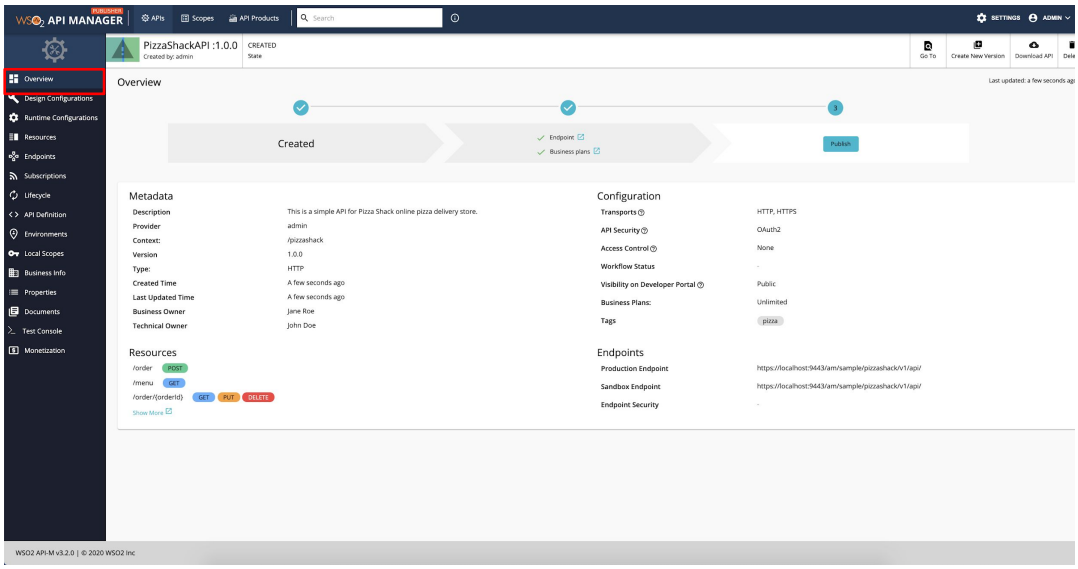


Provide the basic details for the API.

- **API Name** Name of the API is mandatory. Special characters and empty spaces are not allowed for API Name
- **Context** Root context of the API is mandatory.
- **Version** Version for the API is mandatory. For any API published via API Manager, API context + version must be unique.
- **Endpoint** Backend Endpoint of the API. This field is not mandatory.
- **Business Plan** Select the subscription policy/ policies the API should allow. This field is not mandatory

In this step, it is possible to provide all the information and publish the API in a single step, or create an API by only providing mandatory information.

API Overview



After creating the API, you will be redirected to **API Overview** page.

This is a summarized view of the details of the API.

API State Representation

- Represents the current state of the API
- Shows required actions to complete the current state.

Metadata

- Shows the basic information of the API.

Configuration

- Displays the basic configurations of the API. i.e., API security scheme, API Visibility information etc.

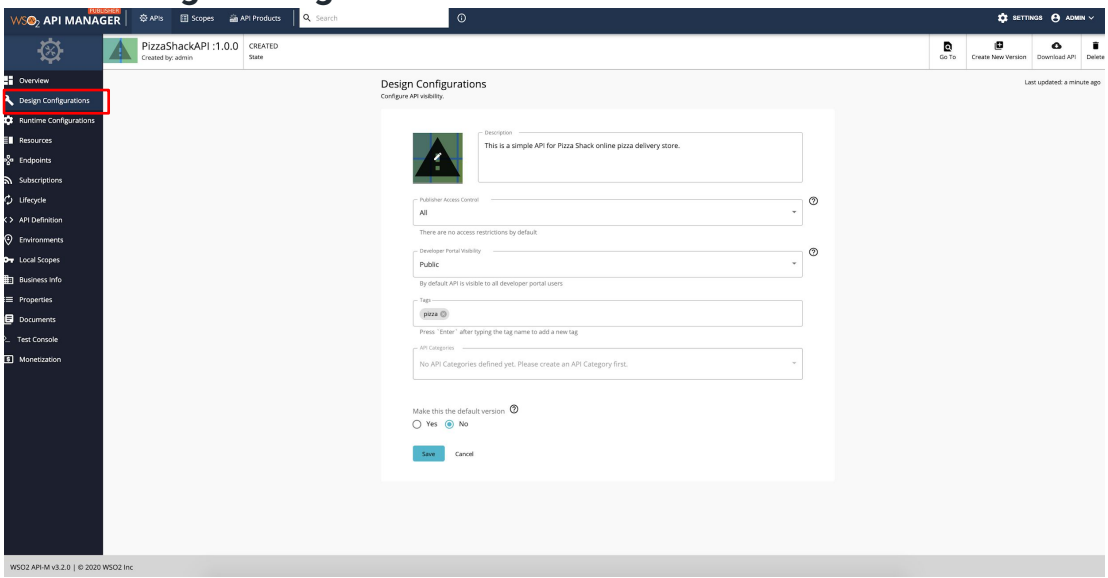
Resources

- List the Resources of the API

Endpoints

- Shows the production/ sandbox endpoints and whether they are secured.

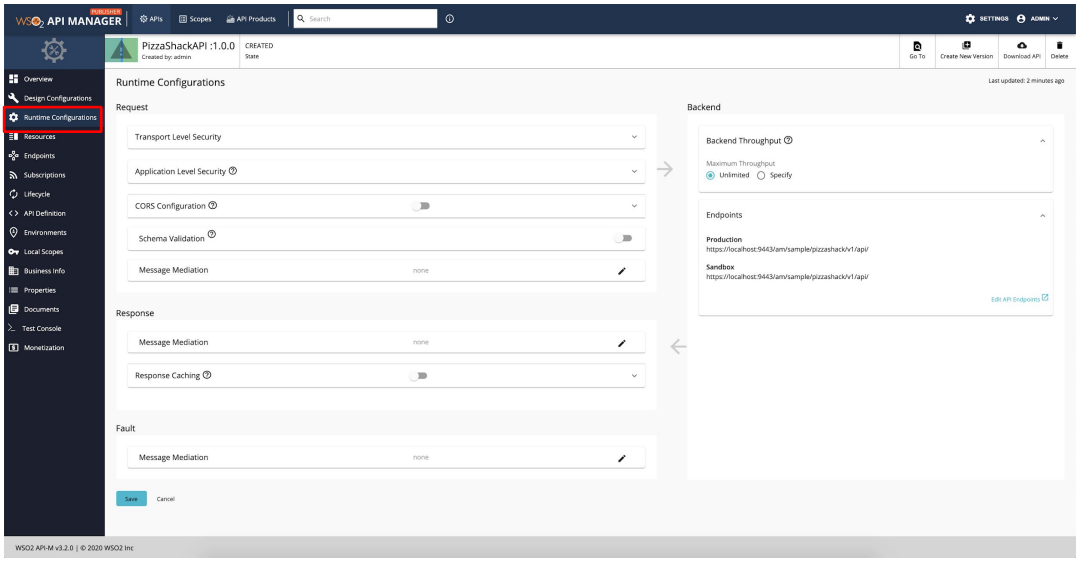
API Design Configurations



Design Configuration includes,

- **Publisher Access control** in the Publisher portal — To which roles, the API should be visible
- **Developer Portal Visibility** in the Developer Portal — To which roles the API should be visible
- **Tags** — Add tags to the API.
- **API Categories** — Set API categories
- **Make the API the default version**

API Runtime Configurations



API Runtime Configuration includes following sections

Request : How the request should be handled in the API.

- **Transport Level Security** Set the protocol type (HTTP/ HTTPS), enable Mutual SSL
- **Application Level Security** Set the application level security settings. (OAuth2, API Key)
- **CORS Configurations** Set Cross Origin Resource Sharing policies
- **Schema Validation** Set/ unset schema validation for requests
- **Message Mediation** Engage request transformation policies.

Response: How the Response flow should be handled.

- **Message Mediation** Engage Response transformation policies
- **Response Caching** Set Response caching configuration

Fault: How to handle any error situation

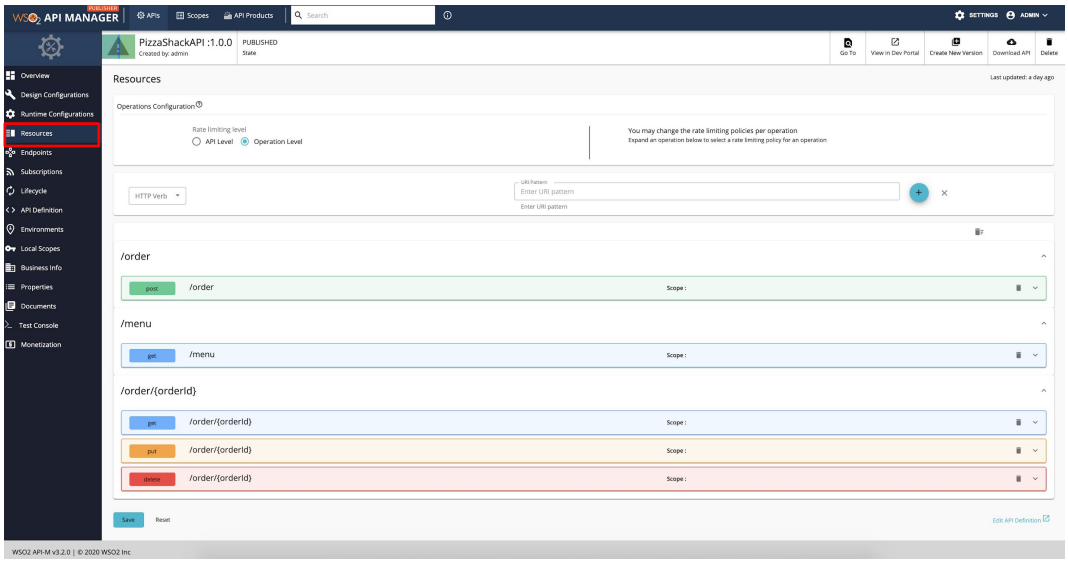
- **Message Mediation** Engage Fault Flow Mediation policy.

Backend: The configuration related to the backend endpoint

- **Backend Throughput** Set the maximum TPS that the backend could handle
- **Endpoints** Displays the Production/ Sandbox backend

- endpoints.

API Resources



Add/ edit or remove resources from the API and set API level, resource level policies.

Supported HTTP Methods

- **GET**
- **POST**
- **PUT**
- **DELETE**
- **HEAD**
- **OPTION**

Delete a Resource Click on Delete in each resource. Once clicked, it will be marked as deleted.

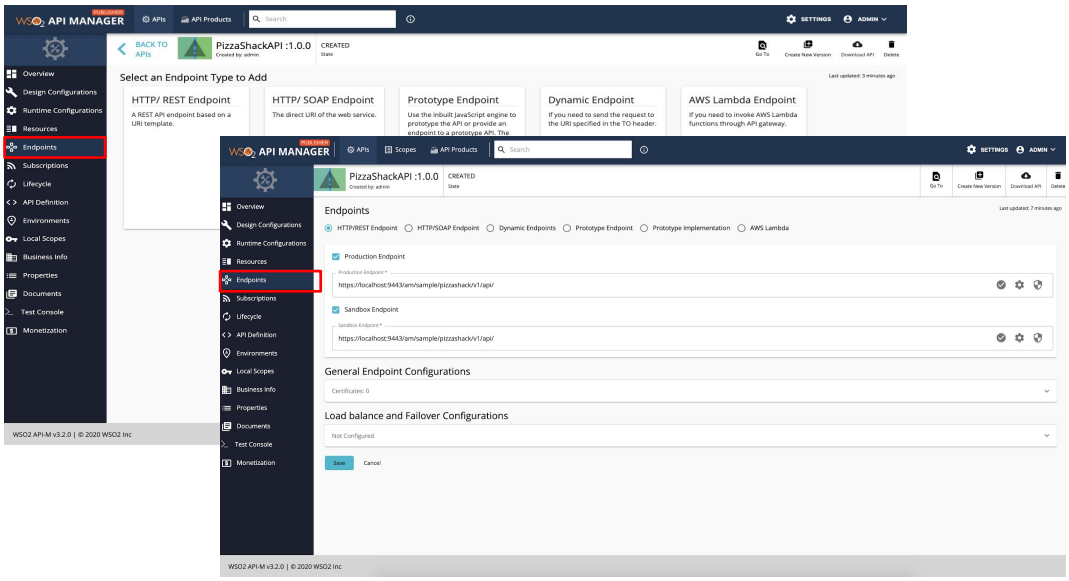
Add new resource

1. Select the HTTP method/ methods
2. Enter the resource path
3. Click “+”

Once added, it will be marked with a red dot.

To save the changes, (deletion/ addition) click Save.

Endpoints



Adding endpoints can be done via the Endpoints page.

If no endpoint is provided in the first step, this page will show an Endpoint Type selection.

- **HTTP/ REST Endpoint** HTTP endpoint based on URI Template
- **HTTP/ SOAP Endpoint** Service URL of a SOAP web service. (for SOAP APIs)
- **Prototype Endpoint**
  - **Prototype Endpoint** Mock endpoint
  - **Prototype Implementation** Use inbuilt javascript engine to mock the api response
- **Dynamic Endpoint** Route the request dynamically based on TO header.
- **AWS Lambda Endpoint** Invoke AWS Lambda functions directly through API

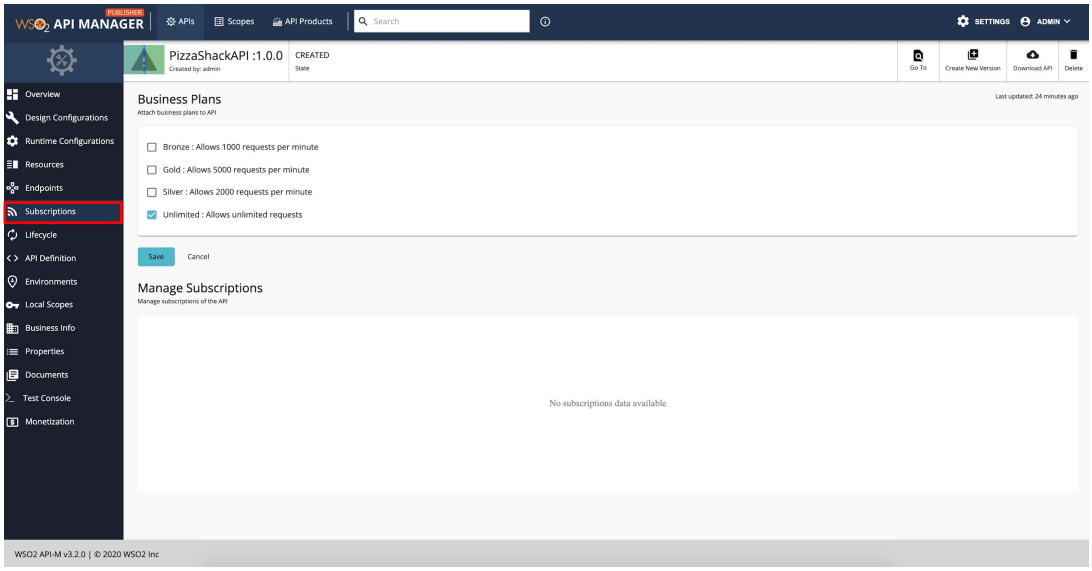
Once the endpoint type is selected, click ADD to add the endpoint to the API.

Endpoints Page

- **Endpoint Type** Change the endpoint type

- **Add Production/ Sandbox endpoints** Check the required endpoint type and enter the URL. Click **Check** in the input field to Test the health of the endpoint.
- **General Configuration** Configure Endpoint Security, and backend certificates
- **Load balance/ Failover Configuration** Add Load balance or Failover endpoints

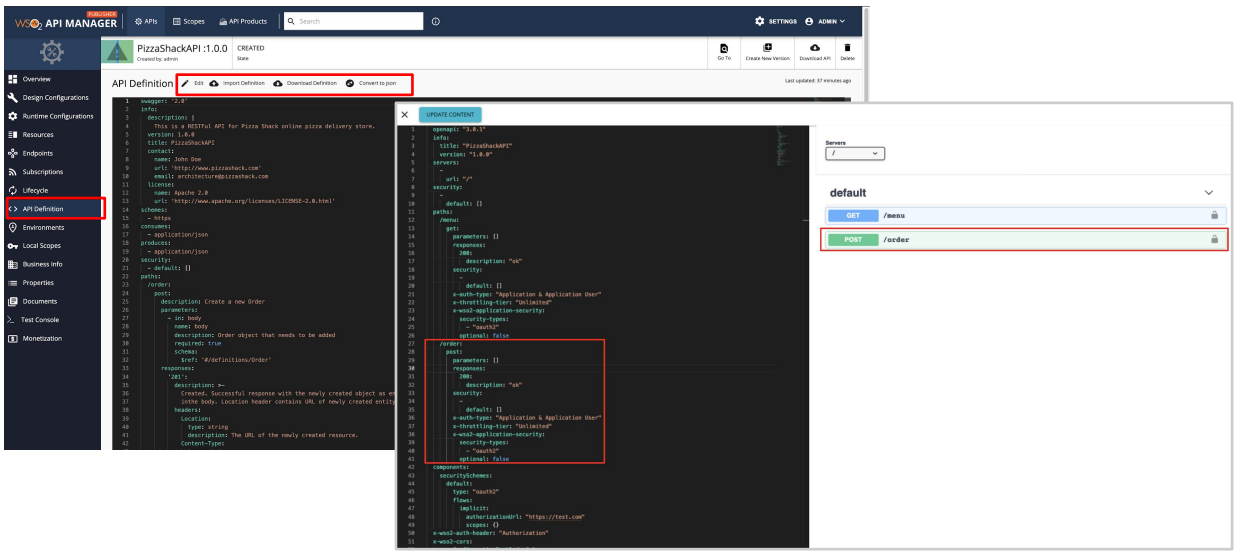
Business Plans and Subscriptions



Selecting business plan(s) for the API and view the consumer application subscriptions who has subscribed to this API

Also, this page also displays the existing API Subscriptions where we can manage them.

API Definition



View the auto-generated open API definition (Swagger) of the API.

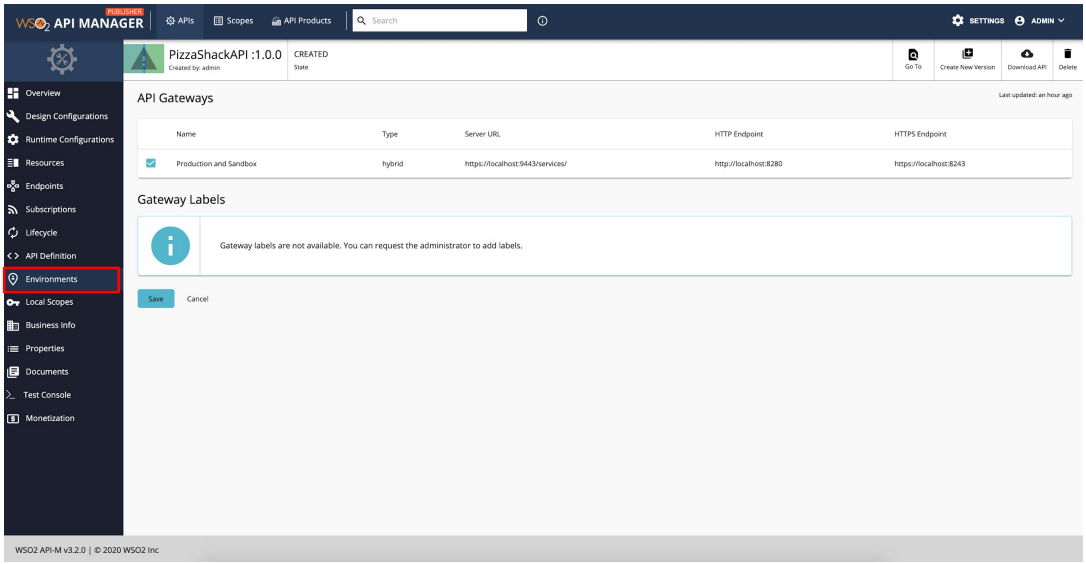
Edit the Definition

- To edit the definition in integrated Swagger Editor, click Edit.
- Do the necessary modifications. (add definition, rename existing paths, add schemas etc)
- Click Update Content to save the changes.

In this view you also can

- Download the definition in yaml or json format.
- Upload a new definition replacing the existing one
- Change the display format to json or yaml

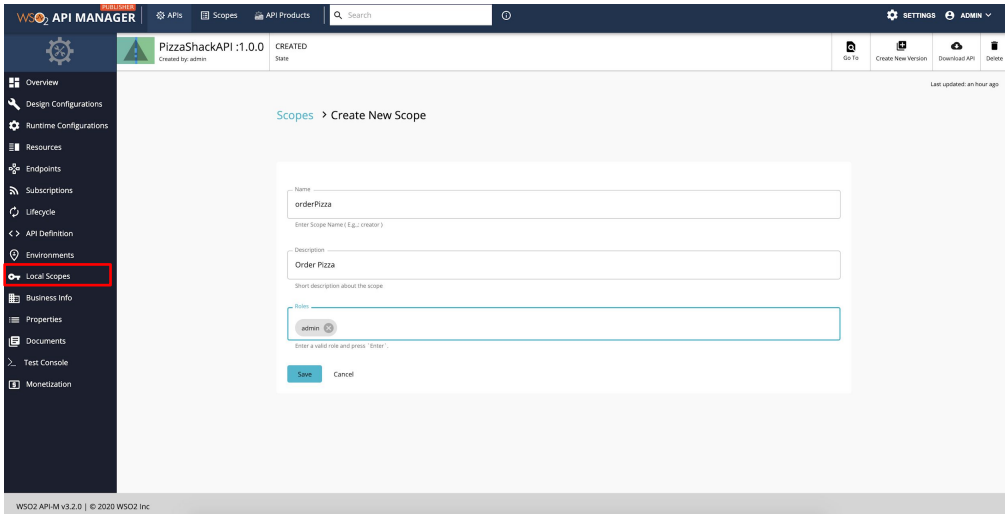
Gateway Environments



Select the required Gateway Environment this API should be deployed to.
Since there are no other environments, select the default Production and Sandbox environment.

When publishing the API, it will be published to all the selected environments.

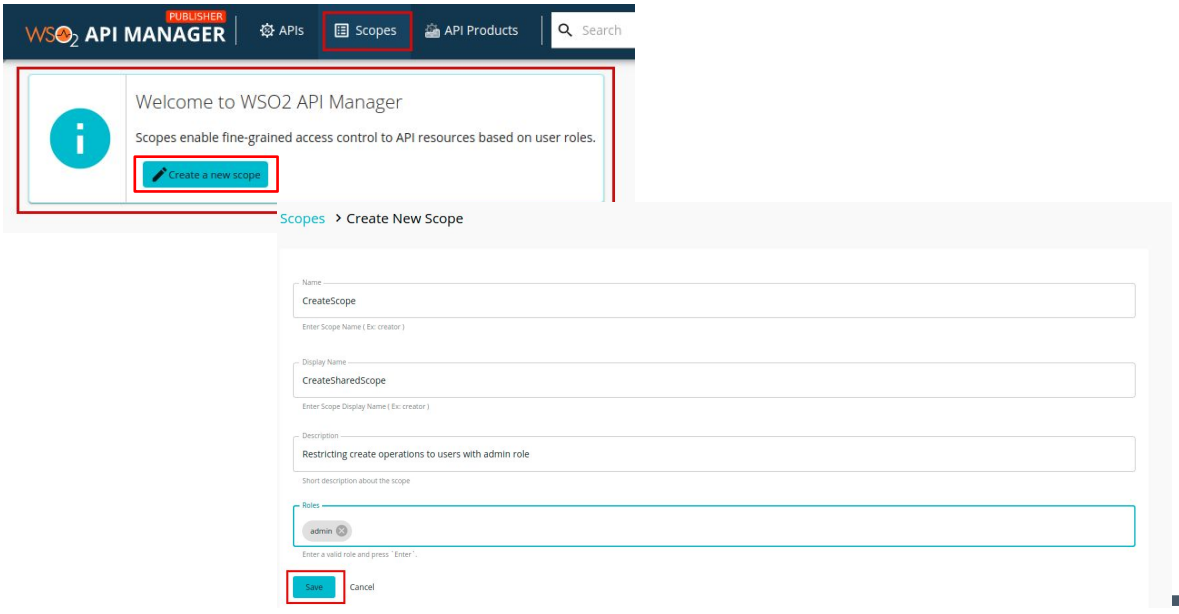
Local Scopes



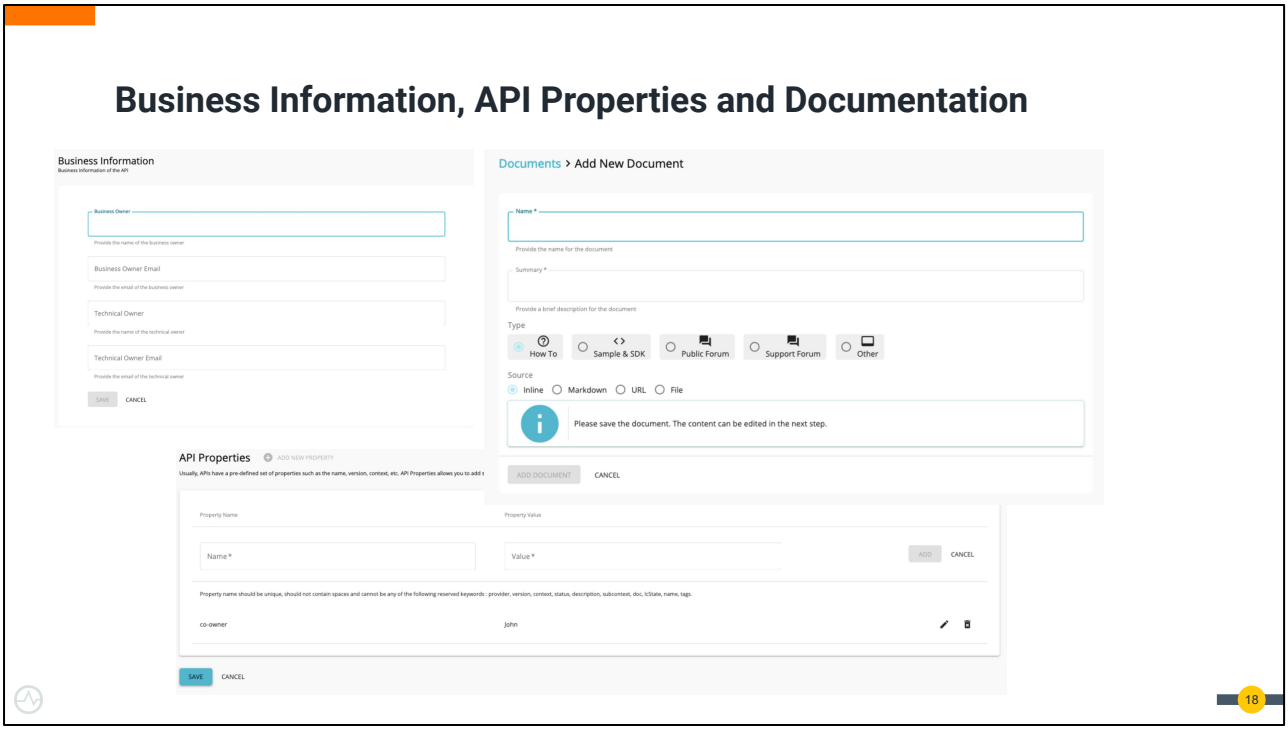
Local Scopes are a role-based access control mechanism which can be enforced to individual API resources. Local scopes are created specific to the API.

This will be discussed in detail in the API Security section.

Shared Scopes



In WSO2 API-M, an OAuth scope can be created before the API is created and shared across multiple APIs of the same tenant.
The API-M Publisher portal provides a scope management UI to view, create, edit and delete these shared scopes.
The shared scope need to be created before API creation/update time



Define various attributes of the API.

Business Information
The business information of the API. Business owner and the technical owner details.

API Properties
Define custom parameters of the API as key value pairs. This is helpful when searching APIs

API Documentation
Add documentation for the API. It can be inline, pdf format, text format etc.

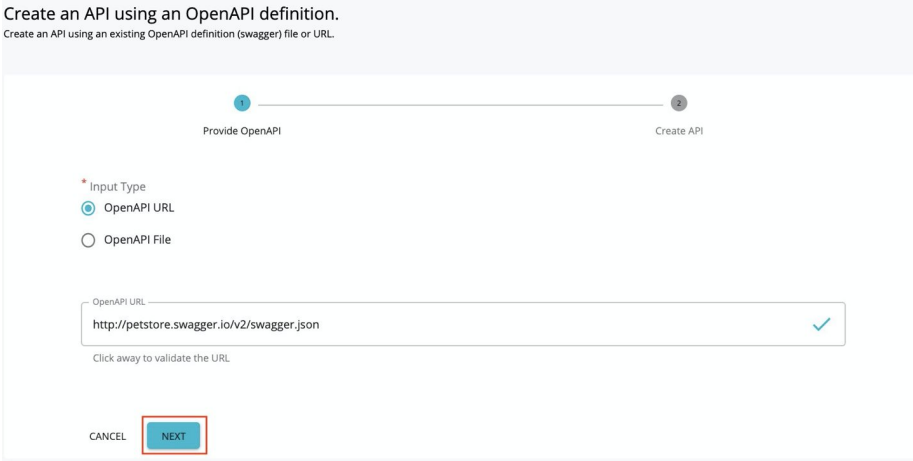
What is Swagger/OpenAPI ?

An OpenAPI file allows you to describe your entire API, including:

- Available endpoints (/users) and operations on each endpoint (GET /users, POST /users)
- Operation parameters Input and Output for each operation
- Authentication methods
- Contact information, license, terms of use and other information.
- API specifications can be written in YAML or JSON. The format is easy to learn and readable to both humans and machines.

REST API from OpenAPI(Swagger) Definition

What is Swagger/OpenAPI ?



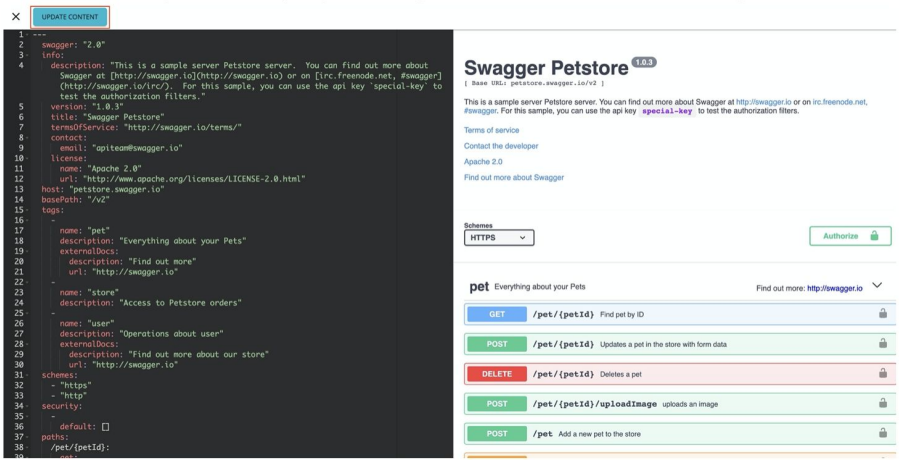
- A specification which is an API description format for REST APIs.
- Swagger 2.0 and OpenAPI 3.0
- The Specification was renamed to the OpenAPI Specification in 2015. OpenAPI 3.0 is the latest version of the specification.
- WSO2 APIM supports OpenAPI 3.0 from version 2.2.0

REST API from OpenAPI Definition

Resources Tab

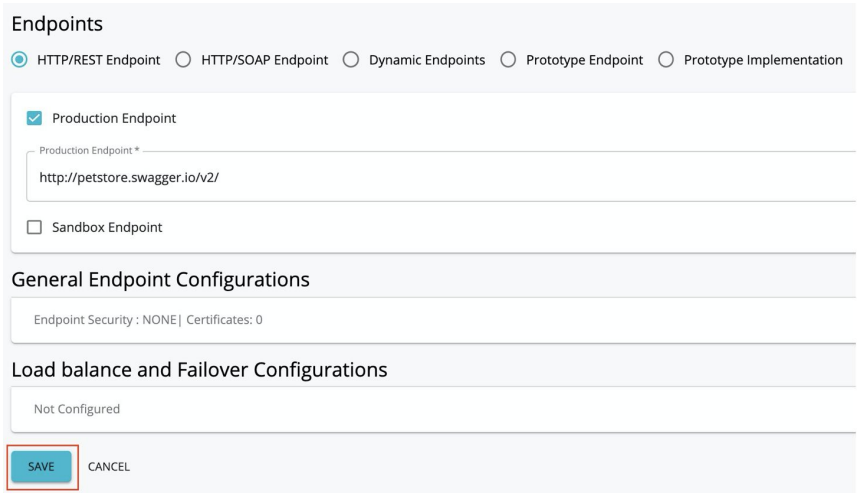
REST API from OpenAPI Definition

Resources can be edited inline



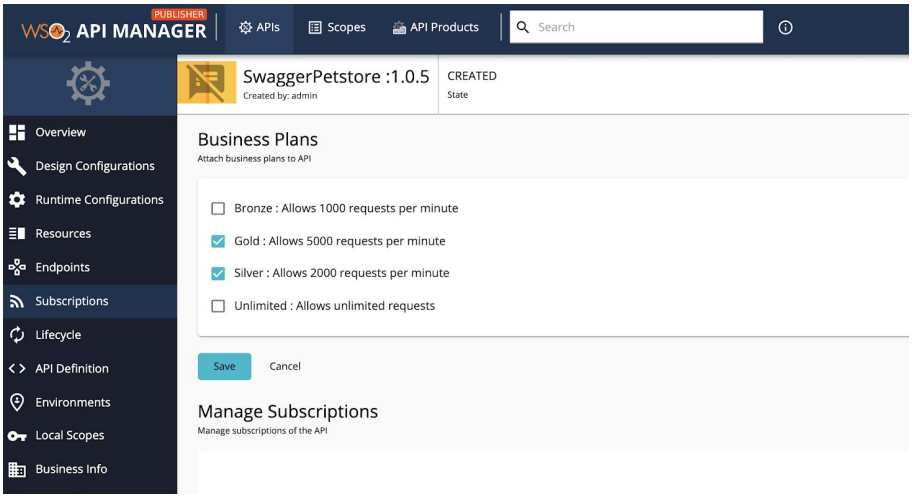
REST API from OpenAPI Definition

Configure Production and Sandbox endpoints



REST API from OpenAPI Definition

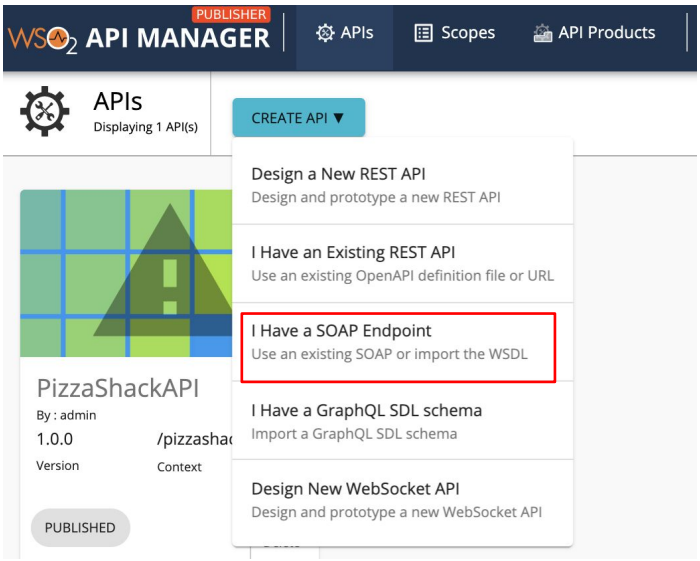
Configure Business Plans in Subscription Tab



What is SOAP (Simple Object Access Protocol)?

- SOAP is an XML based standard communication protocol that permits processes using different operating systems like Linux and Windows to communicate via HTTP and SOAP APIs are based on this protocol.
- REST API -> Swagger
- SOAP API -> WSDL
- The Swagger specification defines a set of files required to describe REST API.
- WSDL (Web Service Description Language), is an XML based definition language. It is used for describing the functionality of a SOAP based web service.

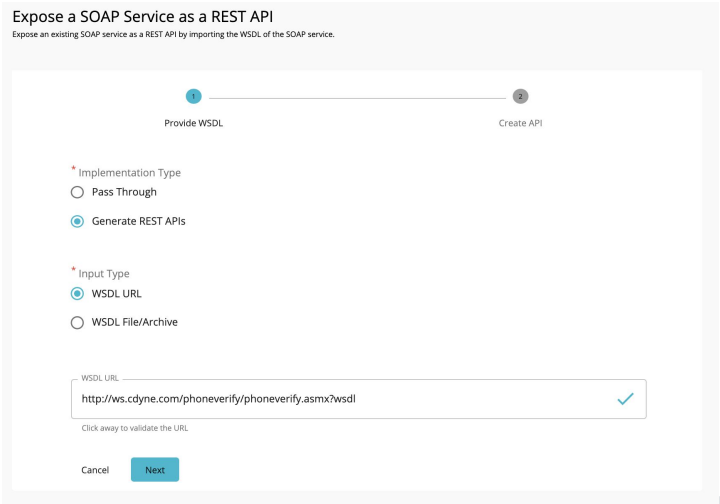
REST API from SOAP Backend



REST API from SOAP Backend

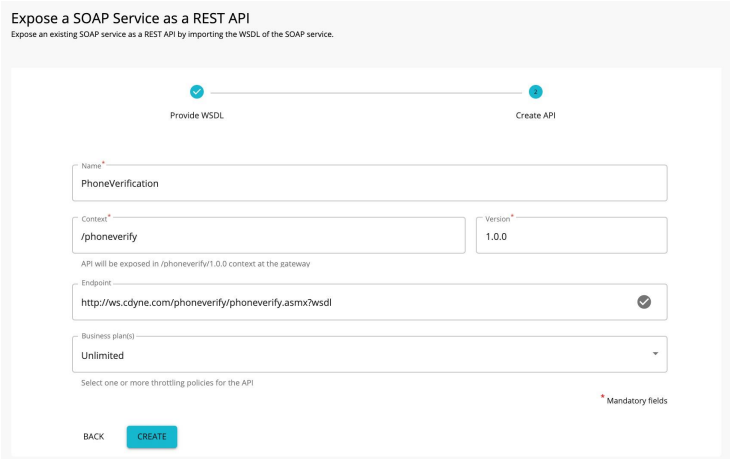
Two options to create APIs for SOAP backends

- Pass Through
- Generate REST APIs



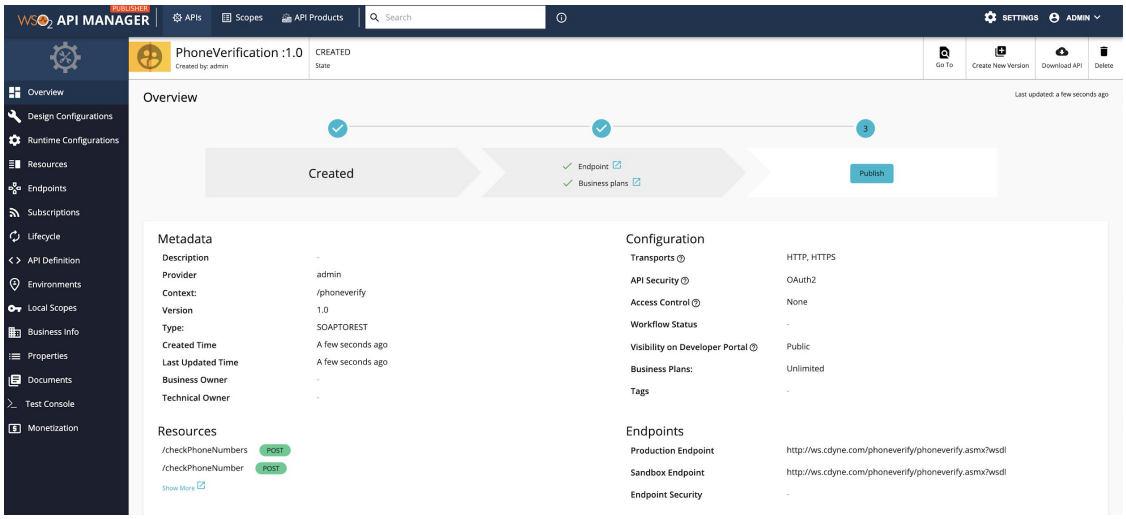
REST API from SOAP Backend

API Creation Menu



REST API from SOAP Backend

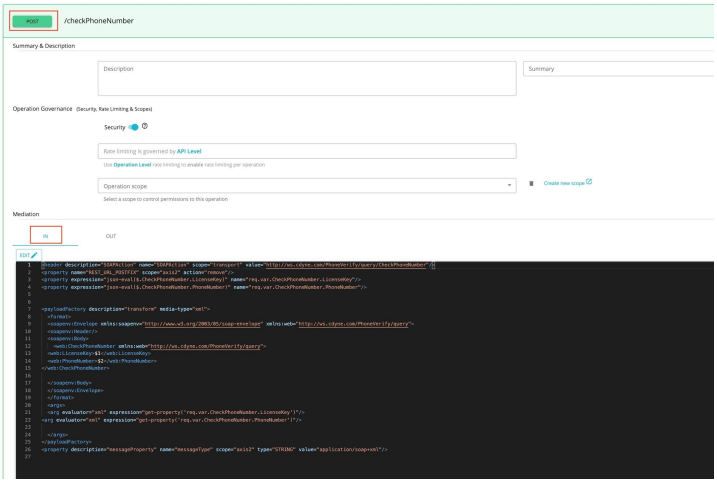
API Overview Page



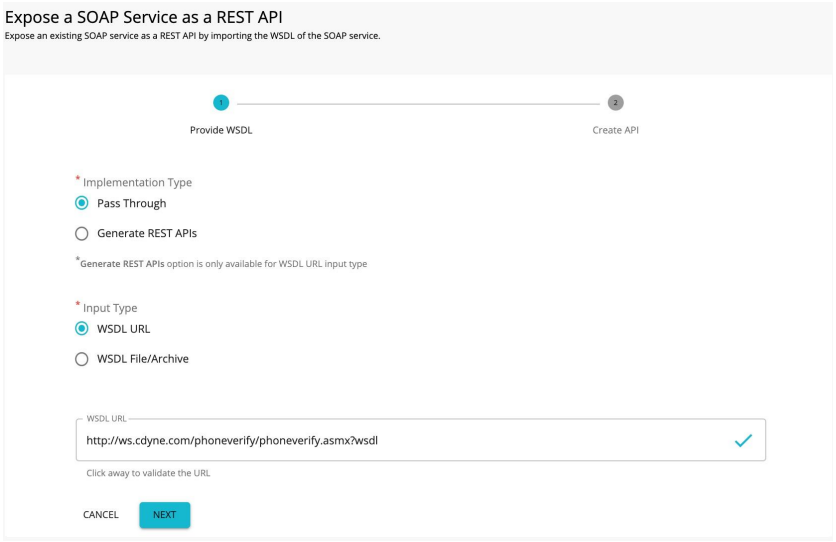
REST API from SOAP Backend

In/Out Sequences of resources

/checkPhoneNumber

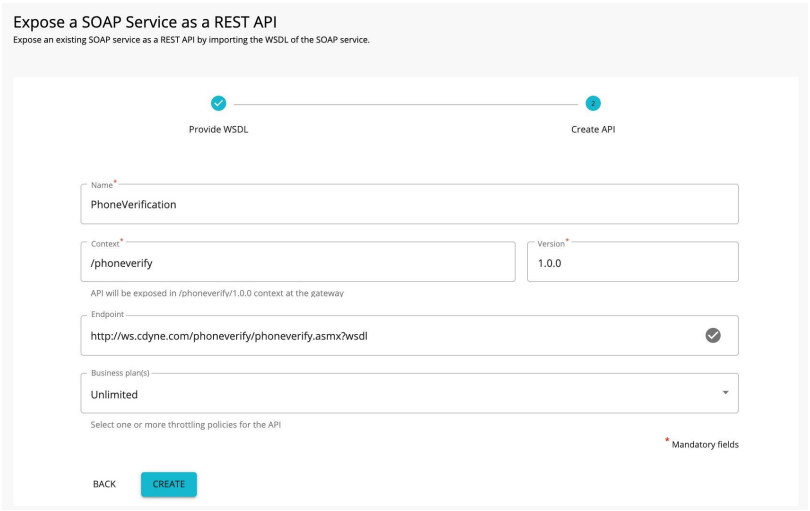


SOAP API as a REST Service



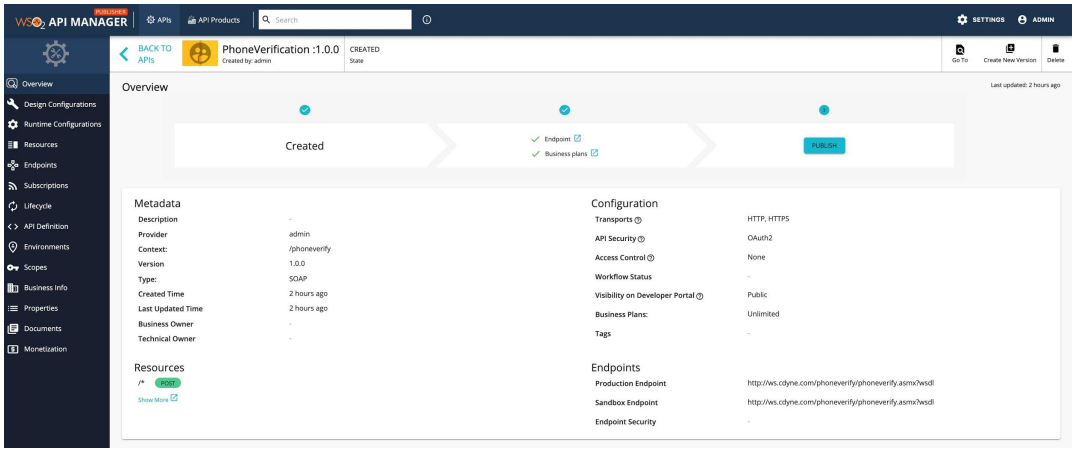
SOAP API as a REST Service

SOAP API Creation

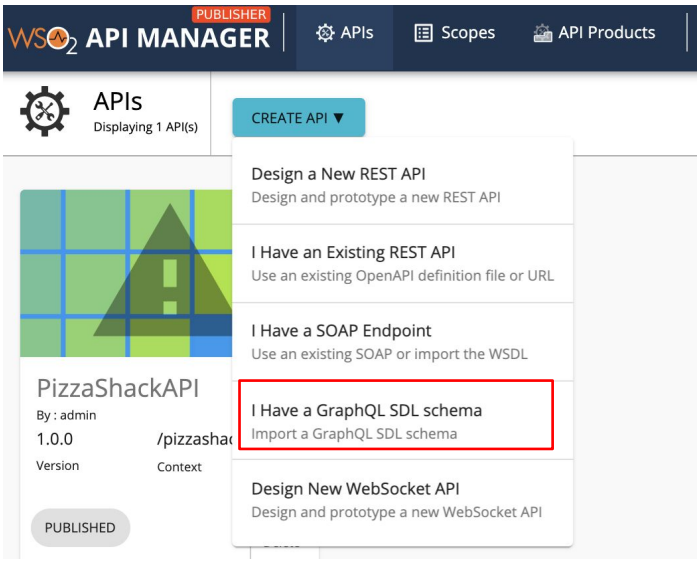


SOAP API as a REST Service

API Overview



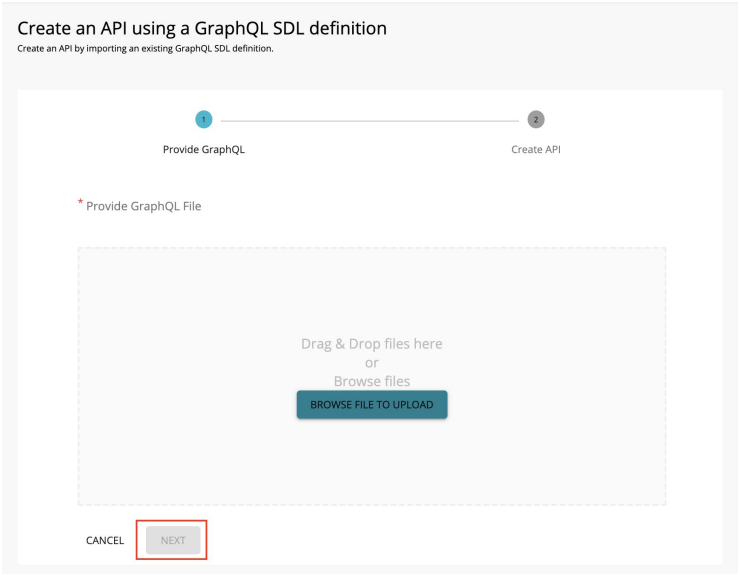
GraphQL API from SDL



GraphQL, which has been developed by Facebook, is a data query language for APIs. When using GraphQL, users can explicitly specify as to what data they need from an API. GraphQL APIs are an alternative to REST-based APIs.

GraphQL API from SDL

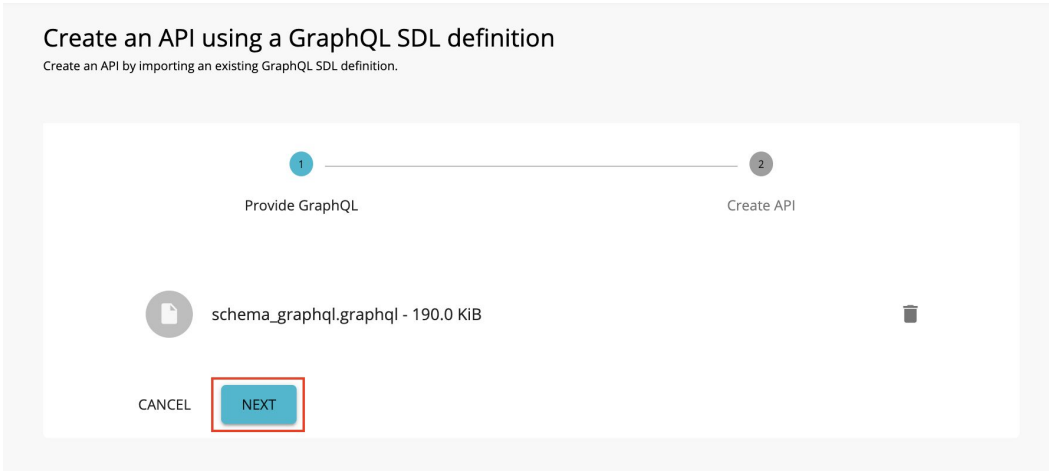
Import SDL



You can use a Schema Definition Language (SDL) schema to design a GraphQL API in WSO2 API Manager (WSO2 API-M) similar to creating SOAP APIs using WSDLs and developing REST APIs using OpenAPI Specifications (a.k.a. Swagger Definitions).

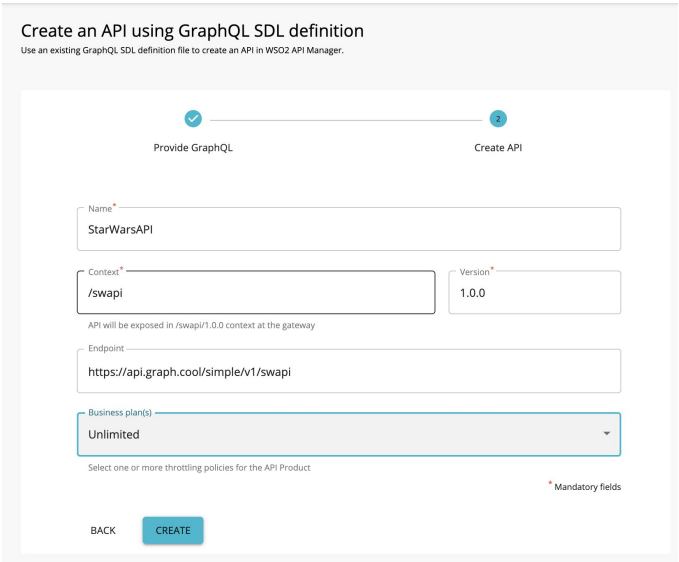
GraphQL API from SDL

Import SDL



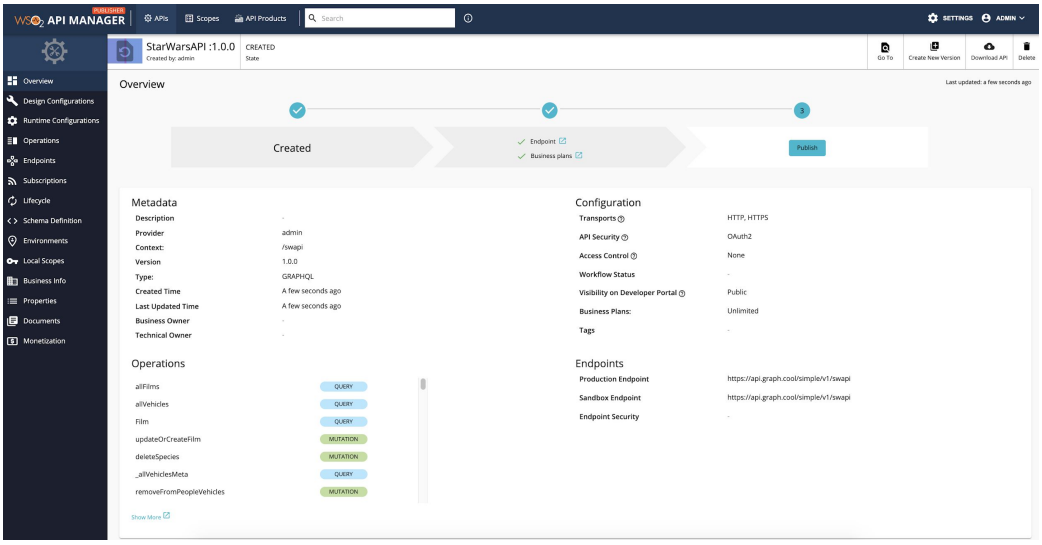
GraphQL API from SDL

Create API



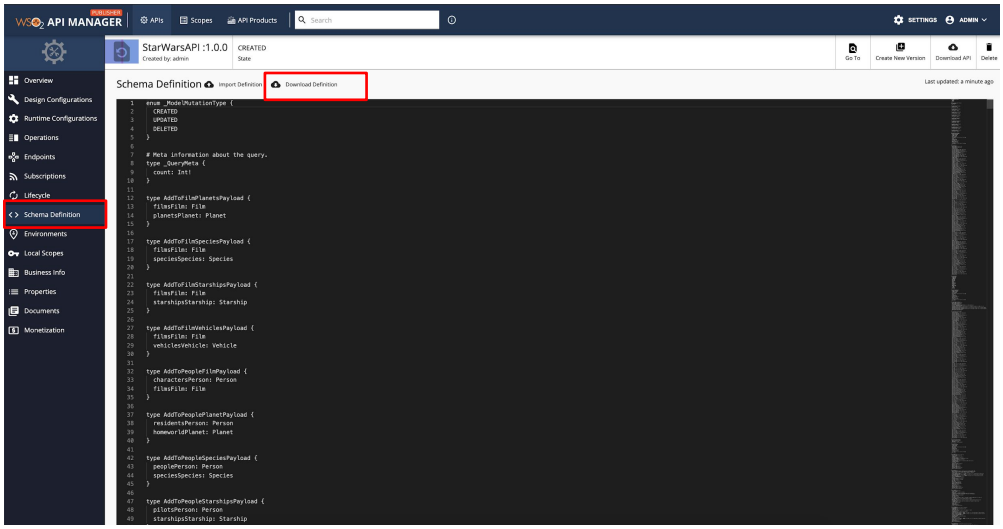
GraphQL API from SDL

API Overview

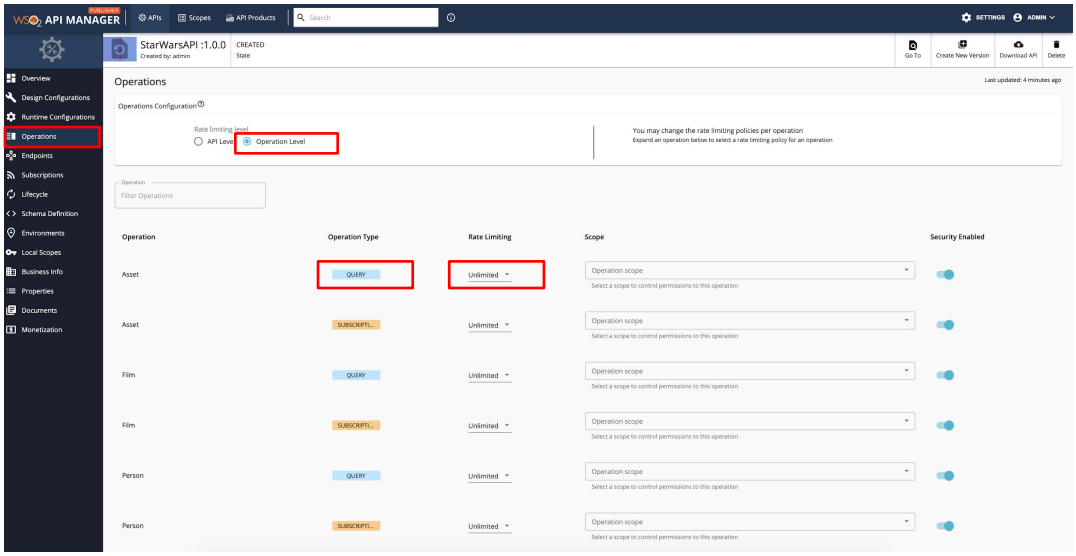


GraphQL API from SDL

Edit SDL Schema



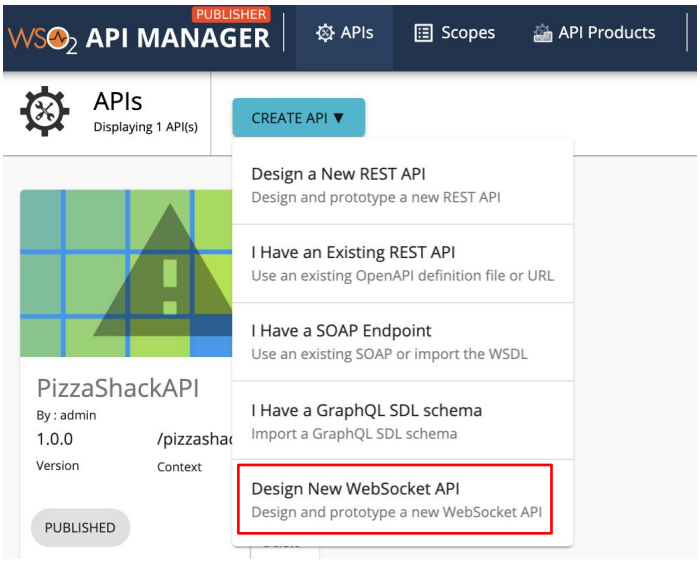
GraphQL API from SDL



WebSocket API

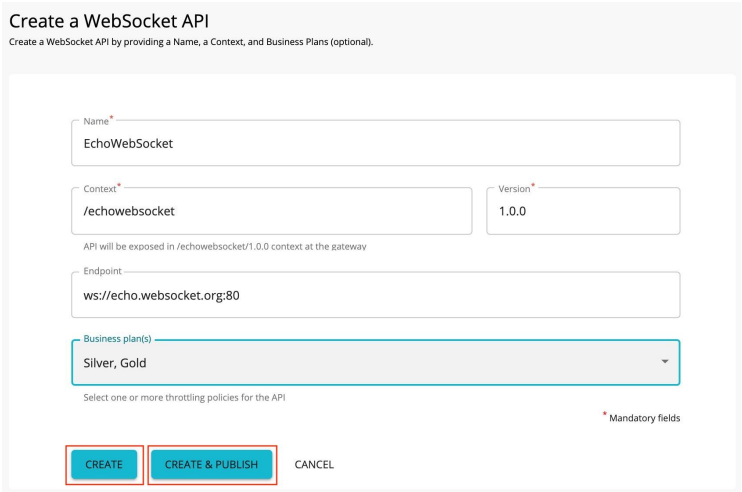
- Protocol similar to HTTP that is part of the HTML5 specification.
- Full-duplex communication between the client and the server over a single connection.
- Advantages
  - Reduce unnecessary network traffic and latency.
  - Allow streaming through proxies and firewalls while simultaneously supporting upstream and downstream communication.
  - Be backward compatible with the pre-WebSocket world by starting up as an HTTP connection before switching to WebSocket frames.

WebSocket API



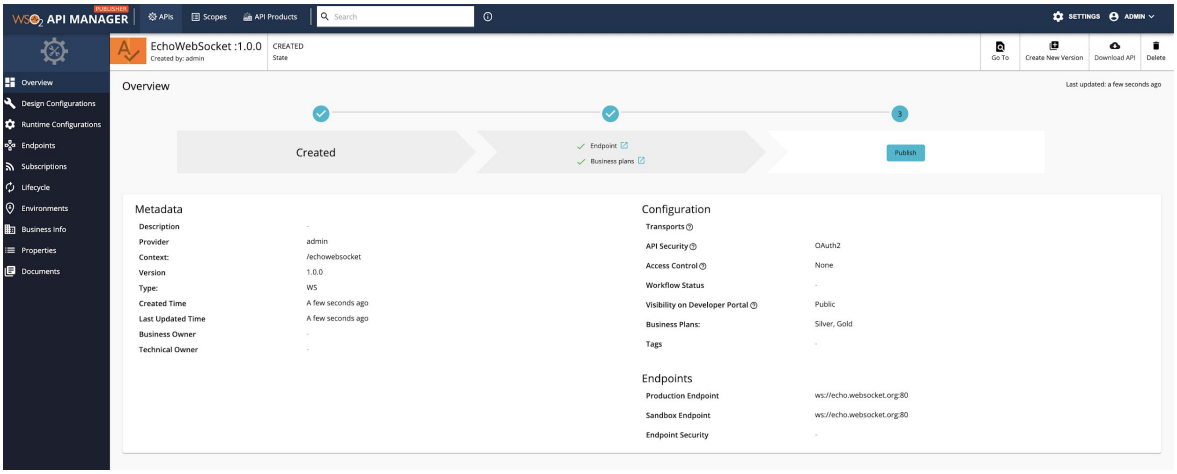
WebSocket API

Create WebSocket API



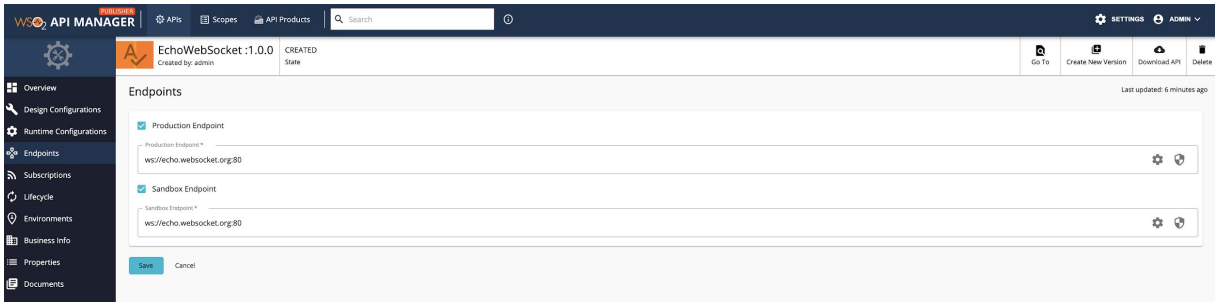
WebSocket API

Overview WebSocket API

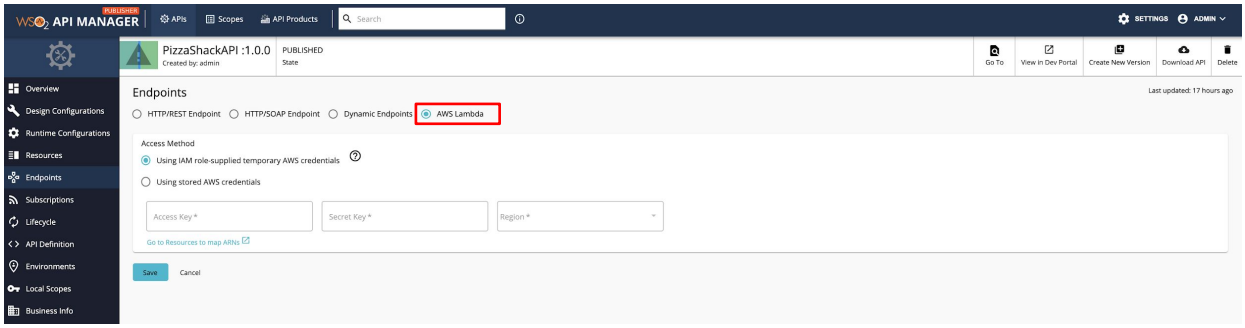


WebSocket API

Configuring endpoint



API with AWS Lambda Endpoint



Doc Link: Create and Publish an AWS Lambda API

After creating the API, select AWS Lambda as the endpoint type in Endpoints Page.

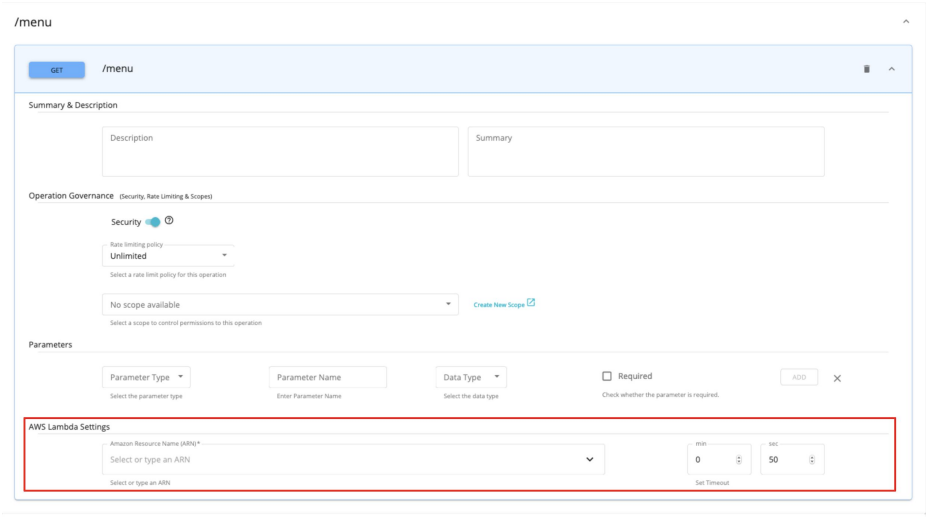
In AWS Lambda type, the credentials of the AWS API should be provided in this page.

Access Method Configuration

- **Using IAM role-supported temporary credentials** If API Manager is running on AWS EC2 instance, resolve the IAM credentials from the AWS environment.
- **Using stored AWS Credentials** Provide the Access Key, Secret and Region manually

Provide the information and save the changes.

AWS Lambda - Select Resource Level ARNs



After providing the AWS credentials in Endpoints page, Amazon Resource Names can be assigned per resource.

Once configured, when invoking the API resource, the respective AWS lambda function will be invoked.

Managing APIs

This section covers how an created API can be managed.

Also covers, UI components related to API Management

Managing APIs

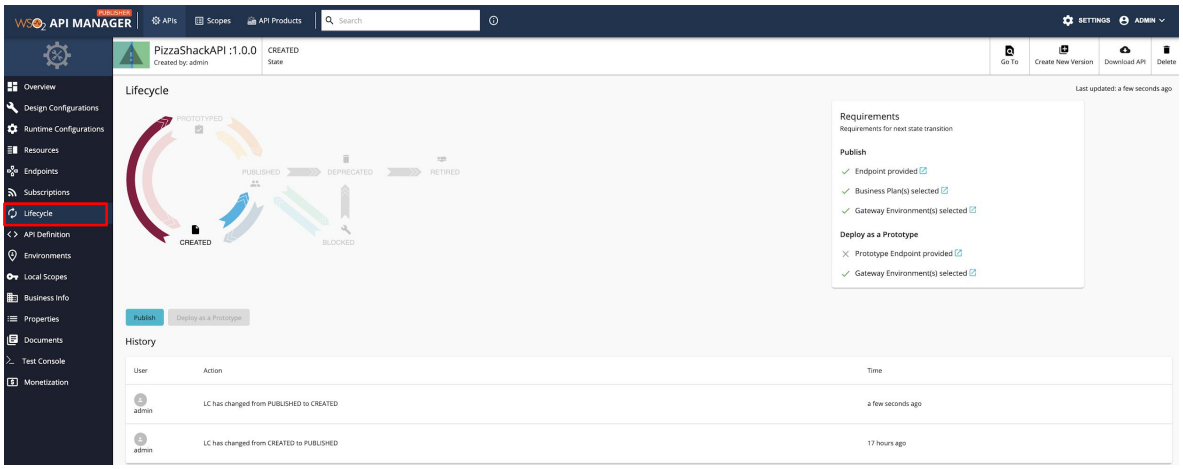
- Changing the Lifecycle State of APIs
- API Versioning
- Managing API Policies

API Lifecycle

- States which represent the stages that an API has in the process of starting to develop an API until it's retirement.
  - Created
  - Prototyped
  - Published
  - Blocked
  - Deprecated
  - Retired

Docs Link: API Lifecycle

API Lifecycle



Select a created API and click on the Lifecycle menu item from the left panel.
In the lifecycle page, the diagram shows the current lifecycle state of the API and next possible lifecycle states.

If the API is in CREATED state, it can either be PUBLISHED or PROTOTYPED.

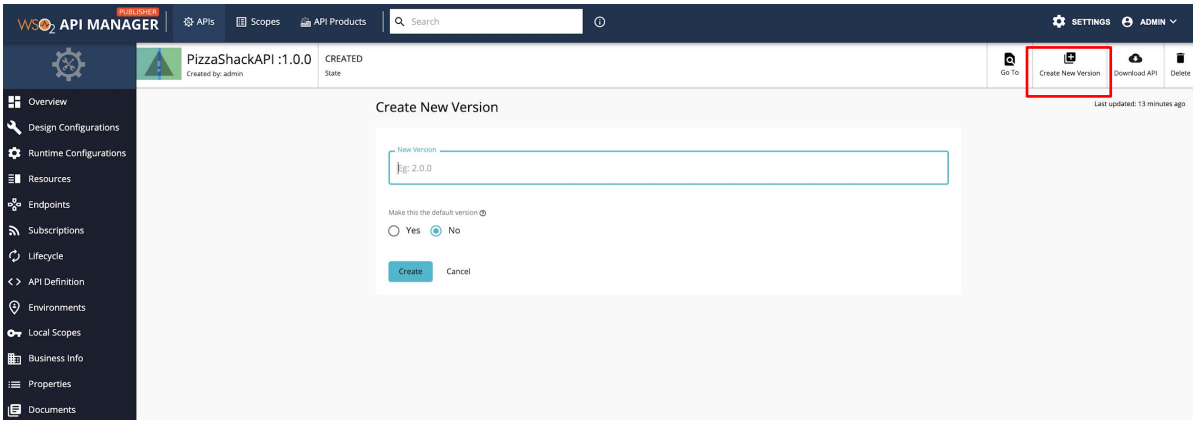
The panel in the right hand side shows the requirements to be fulfilled in order to move to the next lifecycle state.

From CREATED state to PUBLISHED state, there must be a subscription policy applied as well as Endpoints should be provided.
To PROTOTYPES state, prototype endpoints should have to be provided.

If any required requirement is not met, the respective action is not permitted.

API Versioning

Creating a new version of an existing API.



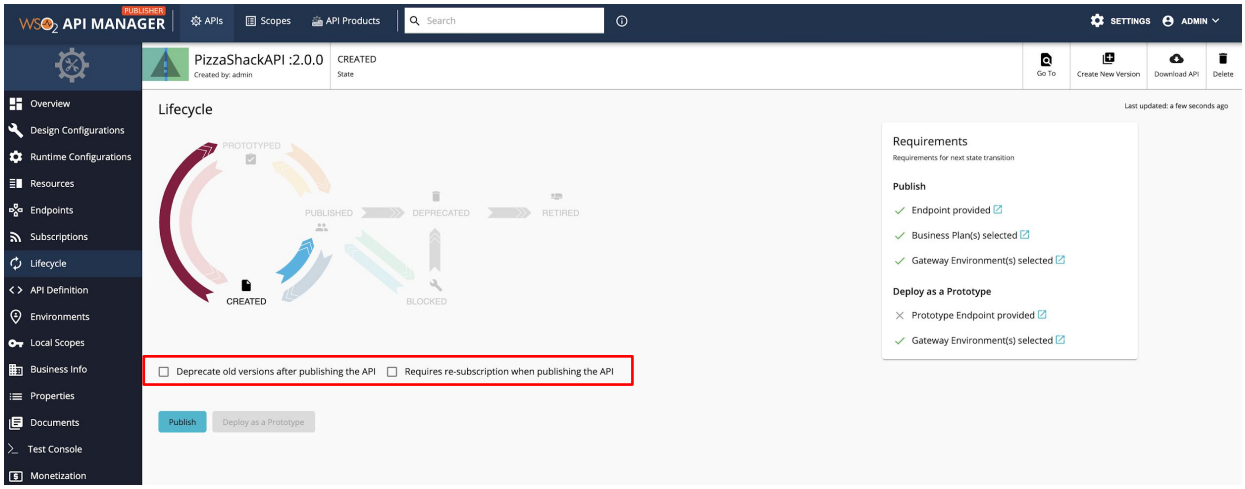
Select the API, and click “Create New Version” in the top panel.

Enter the new version.

Make this the default version: Let the API to be invoked without the version.

The new version will be in **CREATED** State. In order to publish the API, goto lifecycle page and change to **PUBLISHED**.

API Versioning

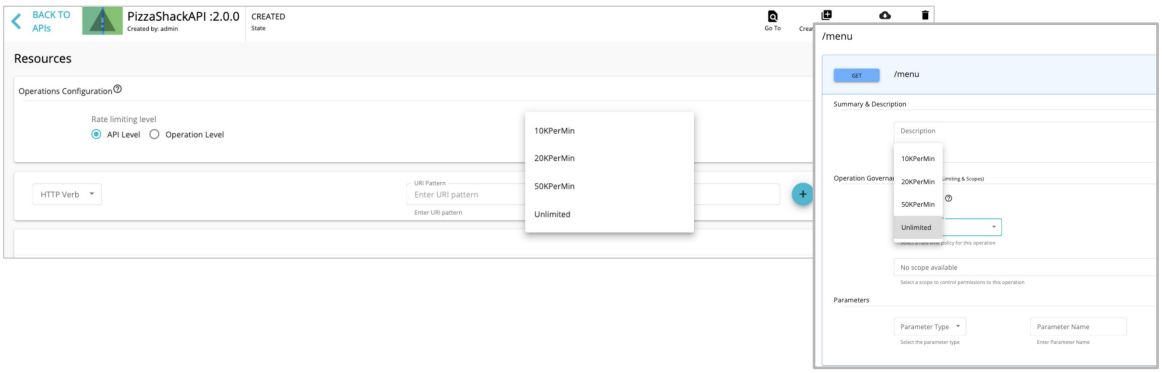


When publishing the new version following options are available.

- **Requires re-subscription when publishing the API**: If NOT CHECKED, the existing subscribers of the old version of the API will be automatically subscribed to the new version of the API. The subscribers do not need to re-subscribe. If checked, the API will be published as a new API with no subscriptions.
- **Deprecate old version after publishing the API**: Automatically deprecate the old version of the API

Managing API Policies

Enforce API Level/ resource level usage policies (Rate limiting policies) to the API.



Throttling policies can be added per API or per resource.

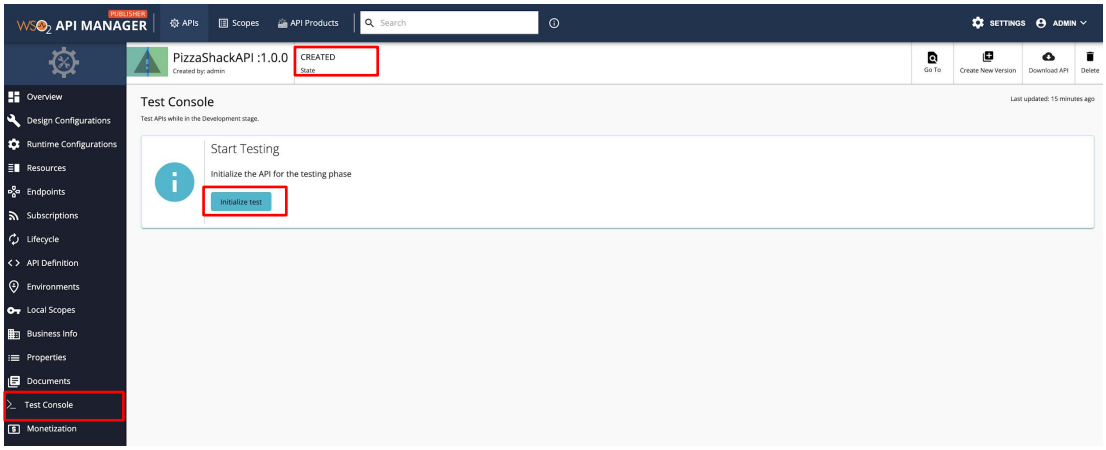
More on Throttling policies in the Throttling section

Testing APIs

This section covers how you can test a created API from the Publisher portal itself

Testing APIs

Testing APIs from the Publisher Portal



Testing APIs is trying out the APIs at the publisher itself to make sure whether the functionalities and behaviours are met before publishing to the gateway for subscribers to access

Only the API developers (creator, publisher) are allowed to test the APIs through the test console.

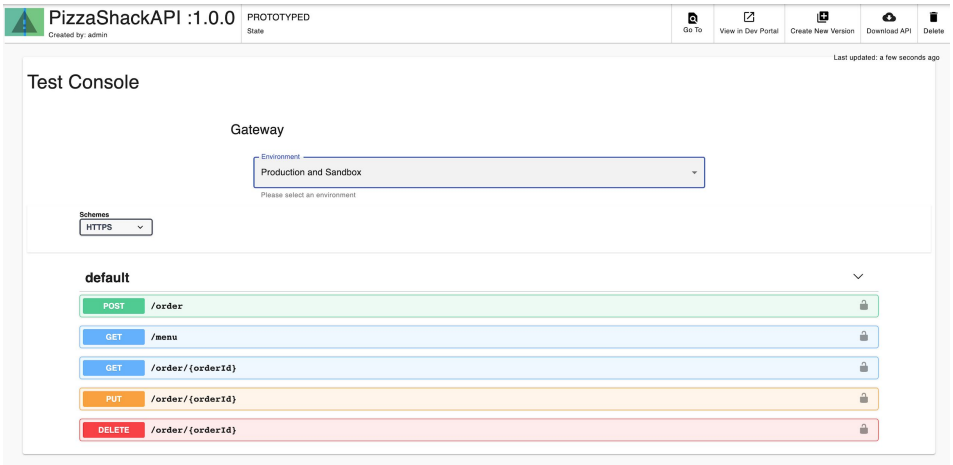
The developers can do the basic functional tests such as checking mediation policy, validating that the request is sent to the back-end and a proper responses is received as expected, request/response schema validation, and making sure that the API resources are defined correctly.

The Test Console menu item in the Publisher portal's left navigation menu can be used to initialize the test at the design phase. Once the developer initiates the test by clicking on the initialize test button, the API is transformed to the prototype(testing) state and the swagger console is populated with a test-key(token) which prevents the unauthorized users accessing the particular API.

API must be in the **created** lifecycle state, the developer can initialize the test.

Testing APIs

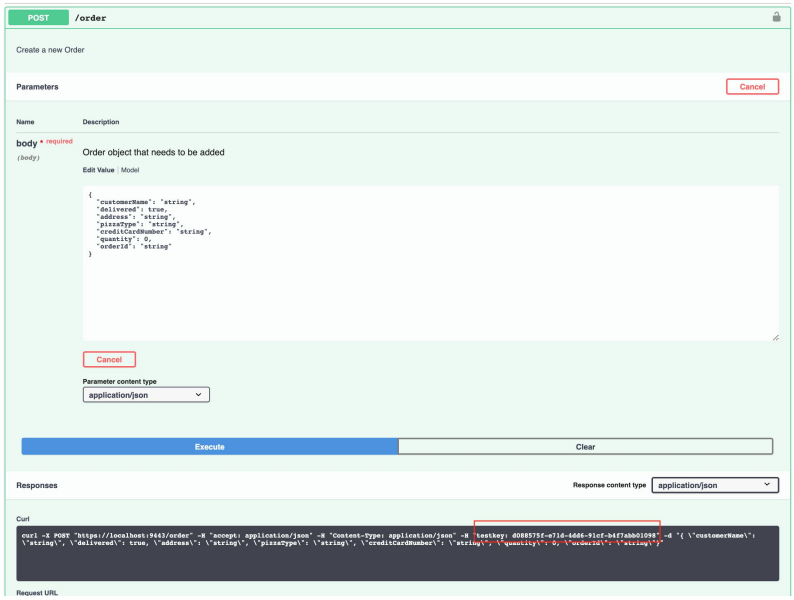
After initiating tests, the swagger UI(API Console) to test the API will be available



Click on the **Initiate Test** button. This will open the swagger UI(API Console) to test the PizzaShack API before publish to the subscribers.

Testing APIs

Test the resource



Expand the POST method and click Try it out. It will generate the test-key to invoke the API. Click Execute.

test-key token - If an API developer initiates a test, the swagger console will be automatically populated with the generated test-key. If you try the test API call through the terminal or command line, make sure you copy the generated test-key.

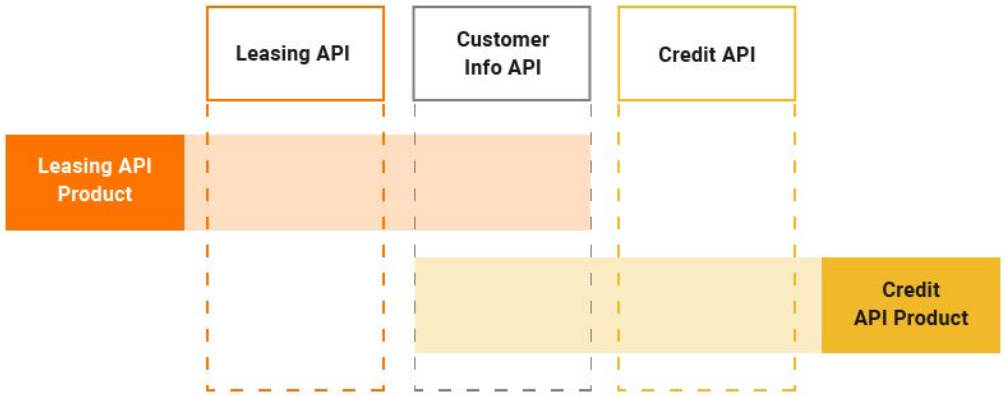
API Products

Let's try it out!

Creating and publishing an API Product with API Manager Publisher

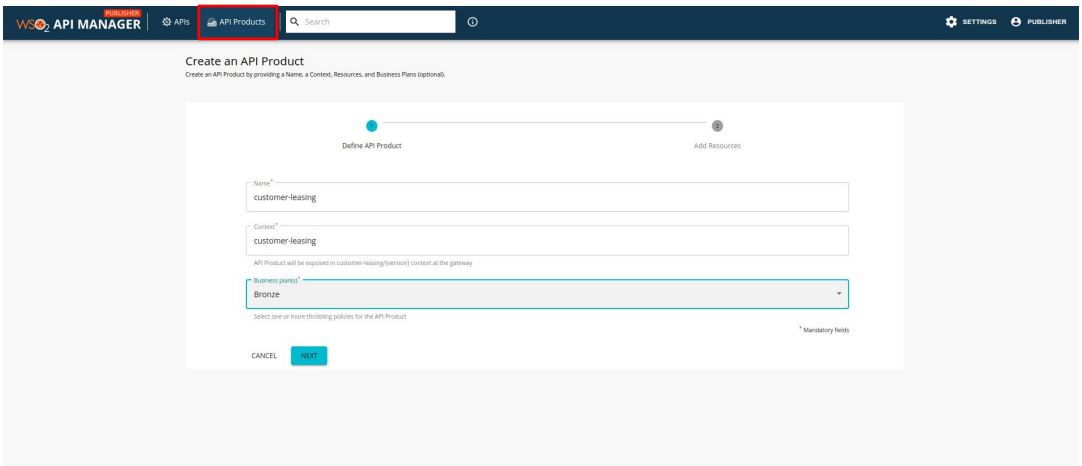
API Product

Packaging mechanism that you can use when you need to bundle a preferred set of resources from multiple APIs and expose it as a separate API interface.



Publishing an API Product

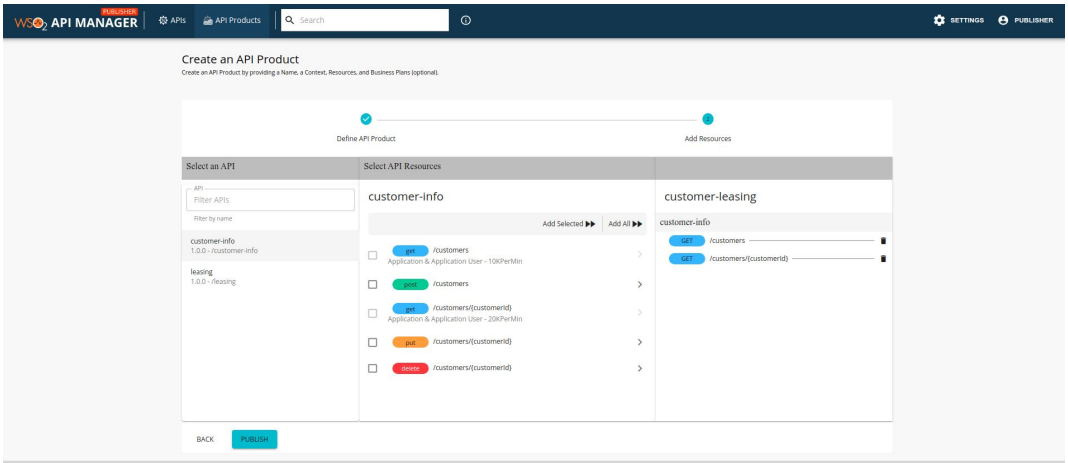
Define API Product



- Go to API Products by clicking on the API Products menu.
- Click Create and start creating the product by providing the name, context, and subscription tier.
- Click Next.

Publishing an API Product

Add Resources from APIs

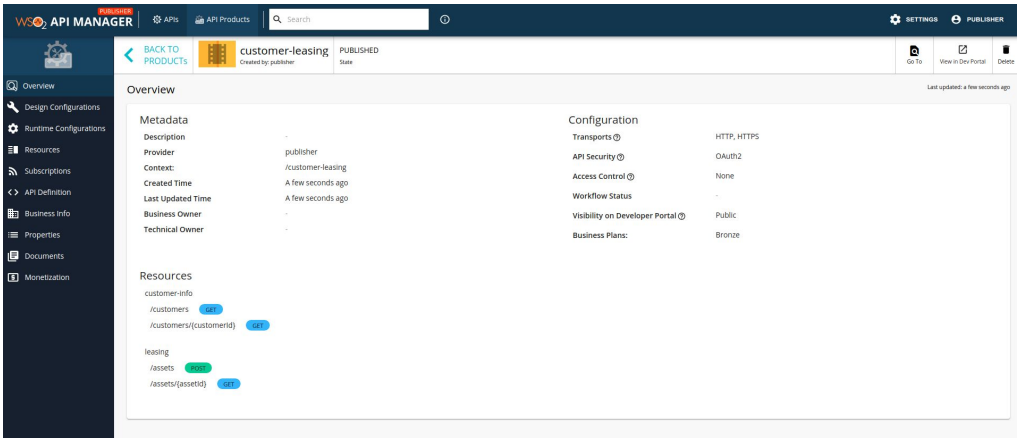


In step 2, the resources from existing APIs can be selected to create the product.

- Select the API from the left panel.
- From the resources of the selected API, check the required resources in the middle panel and click "**Add Selected**". (It is also possible to individually add resources or add all the resources)
- When done, click **Finish**.

Publishing an API Product

Created API Product Overview



Important to note:

- API Products can be only created by users with **publisher** role.
- The only state API Products could have is the **PUBLISHED** state.
- API Products can be created from APIs which are not published (in created state)
- In order to reflect the changes done in the underlying APIs, the product should be updated manually.