# **Practical Exercises: Publishing Integrations to the API Manager**

Training Objective

Learn how to expose an integration service as a managed API by publishing an integration to the API Manager.

Business Scenario

PizzaShack needs to expose an integration service as a managed API in the API Manager to make it discoverable to a wider community of consumers.

High-Level Steps

* Develop the integration service via WSO2 Integration Studio
* Configure service metadata
* Configure the Micro Integrator
* Package the artifacts
* Build and run the service
* Expose the integration service as a managed API via WSO2 API Manager

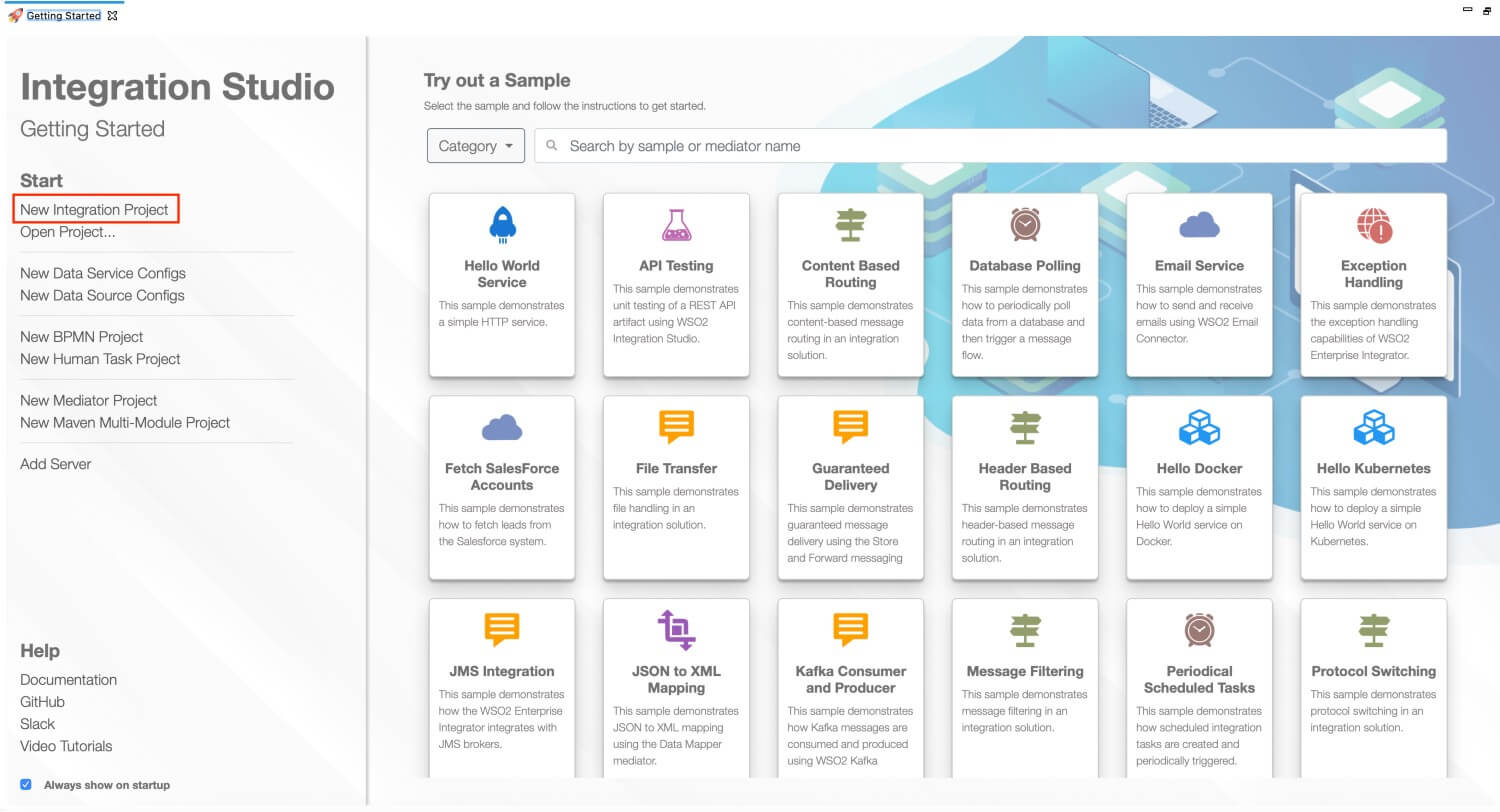
Detailed Instructions

Follow the steps below to develop an integration service via WSO2 Integration Studio and expose it as a managed API to the API Manager.

### **Step 1 - Develop the integration service**

Follow the instructions given in this section to create and configure the required artifacts.

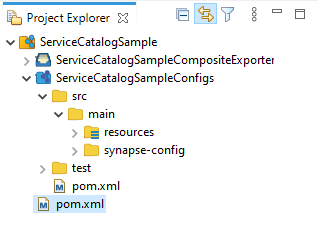
1. Download the relevant [WSO2 Integration Studio](https://wso2.com/api-management/tooling/) based on your operating system.
2. Open **WSO2 Integration Studio**.
3. Click **New Integration Project** in the **Getting Started** tab as shown below.

[](https://apim.docs.wso2.com/en/latest/assets/img/integrate/tutorials/common/create-integration-project.jpg)

This will open the **New Integration Project** dialog box.

1. Enter ServiceCatalogSample as the project name and select the following check boxes to create the required modules.
   * **Create ESB Configs**
   * **Create Composite Exporter**
2. Click **Finish**.

You can see the projects listed in the **Project Explorer** as shown below:

[](https://apim.docs.wso2.com/en/latest/assets/img/integrate/tutorials/service-catalog/project-explorer-service-catalog.png)

**Note**

A **resources** folder is created in the ServiceCatalogSampleConfigs project. This folder holds the Swagger and metadata YAML files. These YAML files will be uploaded to the service catalog later in this tutorial.

1. Create an **Endpoint** artifact.
   1. Right-click **ServiceCatalogSampleConfigs** in the project explorer and click **New** -> **Endpoint**.
   2. Ensure that **Create a New Endpoint** is selected and click **Next**.
   3. Enter the information given below to create the new endpoint.

|  |  |
| --- | --- |
| **Property** | **Value** |
| Endpoint Name | PizzaEP |
| Endpoint Type | HTTP Endpoint |
| URI Template | **Backend Service UR**L that gives the menu as a response - You can use an HTTP mocky endpoint for this |
| Method | GET |
| Save Endpoint in | ServiceCatalogSampleConfigs |

* 1. Click **Finish**.

1. Create a **REST API** artifact.
   1. In the project explorer, right-click **ServiceCatalogSampleConfigs** and click **New -> REST API**.
   2. Ensure **Create A New API Artifact** is selected and click **Next**.
   3. Enter the details given below to create a new REST API.

|  |  |
| --- | --- |
| **Property** | **Value** |
| Name | PizzaShackAPI |
| Context | /pizzashack |
| Save location | **ServiceCatalogSampleConfigs** |

* 1. Click **Finish**.

1. Open the **Source** view of the PizzaShackAPI that you created and apply the following.

<?xml version="1.0" encoding="UTF-8"?>

<**api** context="/pizzashack" name="PizzaShackAPI" xmlns="http://ws.apache.org/ns/synapse">

<**resource** methods="GET" uri-template="/menu">

<**inSequence**>

<**log** description="Request Log" level="custom">

<**property** name="Log Property message" value="&quot;Welcome to PizzaShack Service&quot;"/>

</**log**>

<**send**>

<**endpoint** key="PizzaEP"/>

</**send**>

</**inSequence**>

<**outSequence**>

<**send**/>

</**outSequence**>

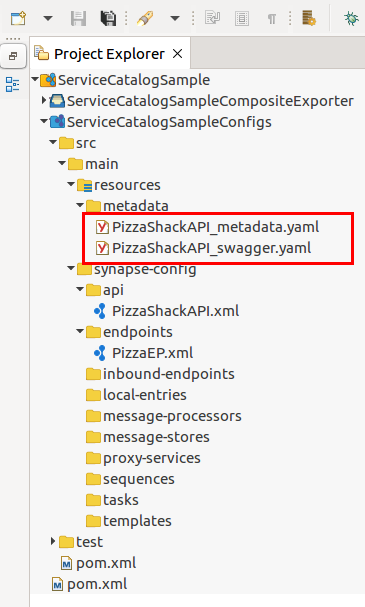
<**faultSequence**/>

</**resource**>

</**api**>

1. When the PizaShackAPI is created, the following two new files are created in the metadata folder.

* PizzaShackAPI\_metadata.yaml - This file contains the metadata of the integration service you created in the previous step.
* PizzaShackAPI\_swagger.yaml - This Swagger file contains the OpenAPI definition of the integration service.



1. Configure service metadata
   1. Open the **PizzaShackAPI\_metadata.yaml** file from the project explorer.
   2. Update the following values in the file.

|  |  |
| --- | --- |
| **Property** | **Value** |
| description | API to fetch pizza menu |
| serviceUrl | http://localhost:8290/pizzashack |

* 1. **Important**: Be sure to change the serviceUrl from HTTPS to HTTP. This is required because the PizzaShackAPI is not secured.
  2. Leave the default values for the remaining parameters.

### **Step 3 - Configure the Micro Integrator**

The Micro Integrator contains a client application, which automatically publishes artifacts to the **Service Catalog** in the **API Publisher** portal.

Let's enable this client for the embedded Micro Integrator of WSO2 Integration Studio.

1. Click the **Embedded Micro Integrator Configuration** ([](https://apim.docs.wso2.com/en/latest/assets/img/integrate/tutorials/common/server-config-64x64.png)) icon on the upper menu to open the dialog box.
2. Uncomment the [[service\_catalog]] section as shown below and change the APIM server configurations accordingly.

**Tip**

The default username and password for connecting to the API gateway is admin.

[[service\_catalog]]

apim\_host = "https://localhost:9443"

enable = true

username = "admin"

password = "admin"

1. **Optionally**, you can encrypt the username and password for better security:
   1. Update the configuration as shown below.

[secrets]

userName = "[admin]"

password = "[admin]"

[[service\_catalog]]

apim\_host = "https://localhost:9443"

enable = true

username = "$secret{username}"

password = "$secret{password}"

* 1. Click **Encrypt Secrets**.

**Tip**

See [Encrypt static (embedded) server secrets](https://apim.docs.wso2.com/en/latest/integrate/develop/using-embedded-micro-integrator/#encrypt-static-embedded-server-secrets) for details.

1. Save the configurations.
2. **Optionally**, inject environment variables to your Micro Integrator.

If you chose to parameterize the serviceUrl in the metadata file, you must inject the parameterized values as environment variables. Shown below are example placeholder values that you may have used in the serviceUrl followed by the corresponding environment variables.

{MI\_HOST} : localhost

{MI\_PORT} : 8290

{MI\_URL} : localhost:8290

**Tip**

See the instructions on [injecting environment variables to the embedded Micro Integrator](https://apim.docs.wso2.com/en/latest/integrate/develop/using-embedded-micro-integrator/#injecting-environment-variables-to-embedded-micro-integrator).

1. Package the artifacts
   1. Open the **pom.xm**l file of the **ServiceCatalogSampleCompositeExporter** module.
   2. Ensure that the following artifacts are selected in the POM file.
      1. PizzaShackAPI
      2. PizzaEP
   3. By default, the **Publish to Service Catalog** checkbox is enabled. If not, select the checkbox in the wizard so that it will include metadata files of the selected artifacts.
   4. Save the changes.
2. Start the API Manager runtime - Start the API Manager runtime before starting the Micro Integrator.

### **Step 6 - Build and run the service**

Let's deploy the [packaged artifacts](https://apim.docs.wso2.com/en/latest/tutorials/integration-tutorials/service-catalog-tutorial/#step-3-package-the-artifacts) in the embedded Micro Integrator:

**Info**

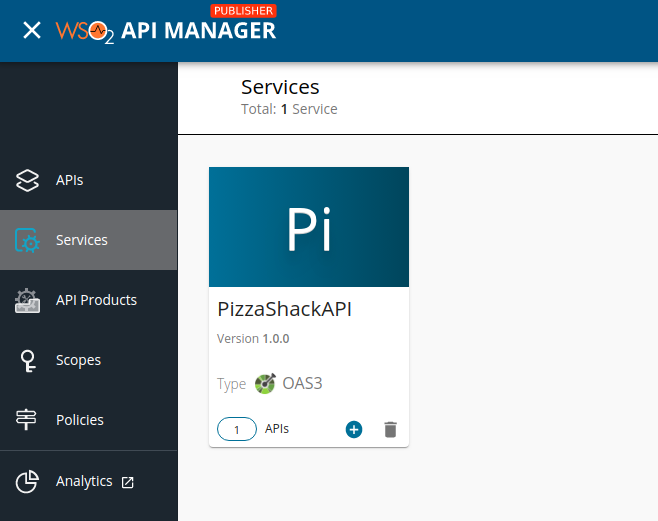
When you do this step:

1. The Micro Integrator first reads the metadata files.
2. If you used placeholders in the metadata file, they are replaced with environment variable values and a ZIP file is created.
3. Finally, it uploads the metadata to the API management runtime.
4. Right-click the composite exporter module and click **Export Project Artifacts and Run**.
5. In the dialog box that opens, confirm that the required artifacts from the composite exporter module are selected.
6. Click **Finish**.

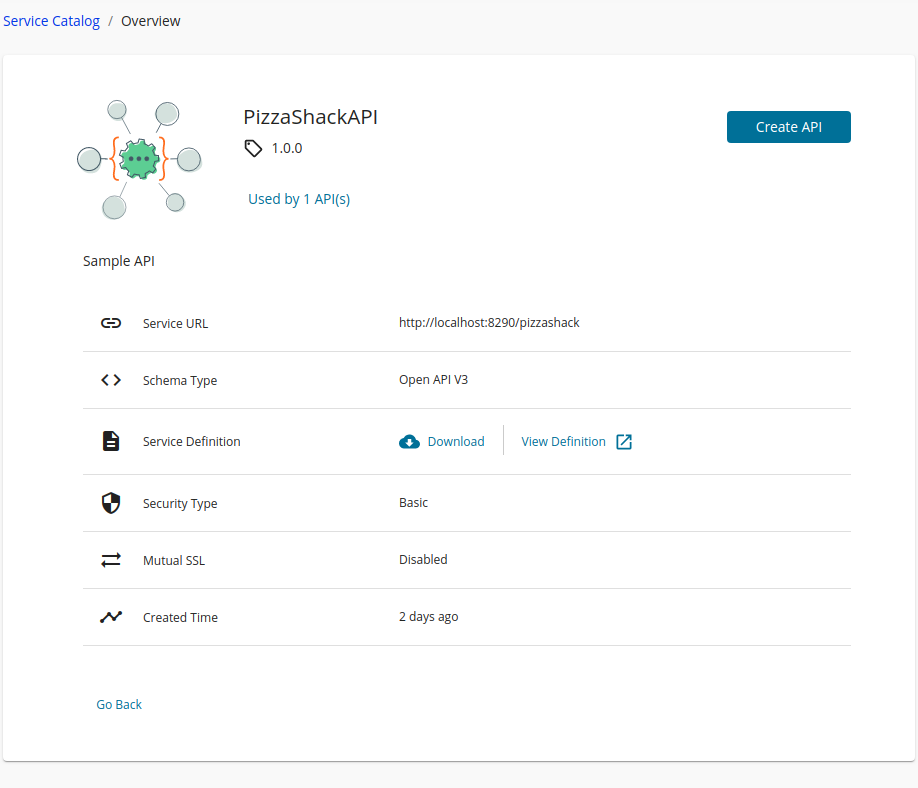
The artifacts are deployed in the embedded Micro Integrator and the Micro Integrator starts. The integration service is also deployed in the **Service Catalog** during server startup. You will see the following in the server start-up log.

Successfully updated the service catalog

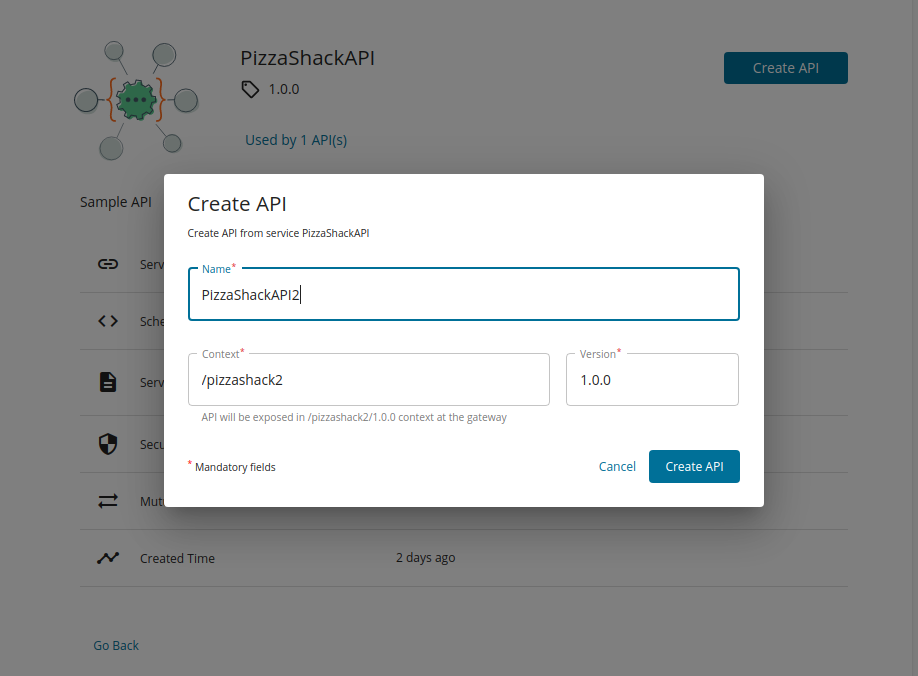
1. Create and Deploy the API
   1. Sign in to the API Publisher portal: <https://localhost:9443/publisher>
   2. You can also click the **hamburger** icon on the upper-left and click **Services** to see the available services.



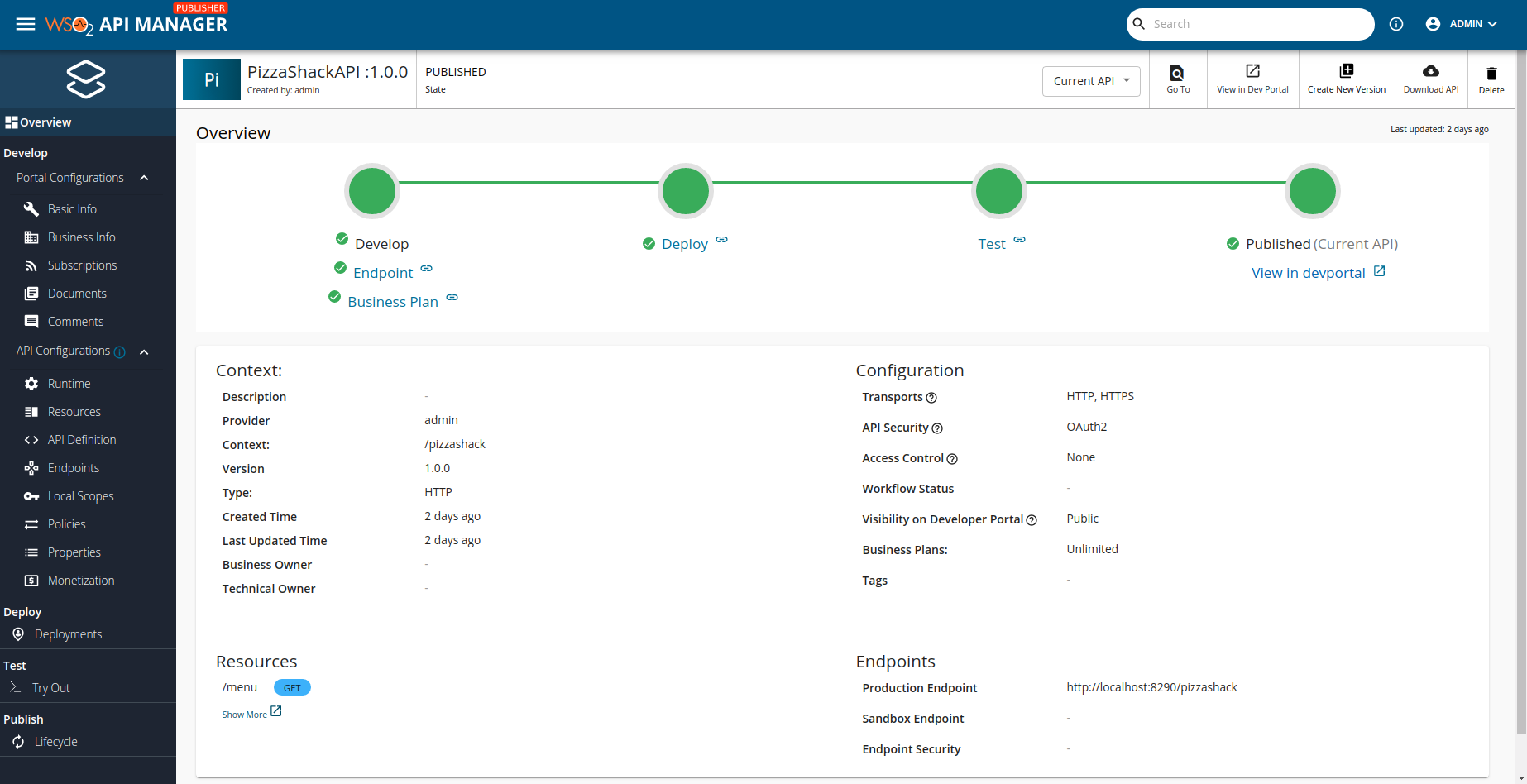
* 1. Open **PizzaShackAPI** from the above list.



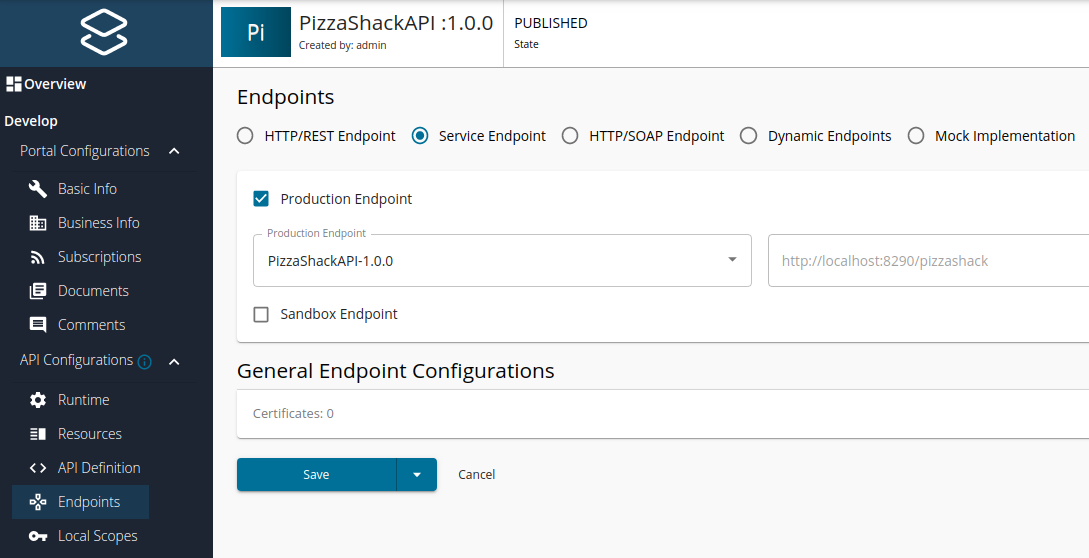
* 1. Click **Create API** in the above screen to open the Create API dialog box.



* 1. Specify an API name, context, and version, and then click **Create API**.
  2. You can now see the new API's overview page.



1. You can use the left-hand navigation to explore the new API. Click **Endpoints** in the left-hand navigator. You will see that the new API uses the integration service deployed in the Micro Integrator as the endpoint (backend).



1. Select business plans.
   1. Go to the API overview and click **Business Plan**.
   2. Select at least one business plan for the API and save.
2. Deploy API in the Gateway
   1. Go to the API overview and click **Deploy**.
   2. Click **Default** to specify the gateway environment and host.
   3. Click **Deploy**.
3. Publish the API
4. Subscribe to the API
5. Generate an access token and try out the service
6. You will get the response from the Pizza Shack backend service and in the **Console** tab of WSO2 Integration Studio and you will see the following message:

INFO - LogMediator message = "Welcome to PizzaShack Service"

Expected Outcome

As a result of exposing the integration service as a managed API in the API Manager in this exercise, the integration service is available for a wider community of consumers.