Microgateway Quick Start

Before you begin,

1. Download the version 2.5.0 of the WSO2 API Microgateway distribution and extract it.
2. Append the full path of the /bin folder of the extracted Microgateway distribution to the PATH environment variable.

If you are unable to append the path, you can alternatively run the Microgateway commands by navigating to the <MICROGW_HOME>/bin folder and running the commands as ./micro-gw instead.

Let's go through the main use case of the Microgateway:

- Deploy and subscribe to the sample API
- Generate a distribution for a single API
- Generate a JWT token and invoke the API

Deploy and subscribe to the sample API

1. Open the API Publisher (https://<hostname>:9443/publisher) and sign in with admin/admin credentials.
2. Close the interactive tutorial that starts automatically if you are a first-time user.
3. Click the Deploy Sample API button. It deploys a sample API called PizzaShackAPI into the API Manager.

The Deploy Sample API option is available only when there are no APIs in API Publisher. If you have already created an API, this option is not available.

4. Click PizzaShackAPI to open it.
5. Go to the Lifecycle tab and note that the State is PUBLISHED. The API is already published to the API Store.
6. Sign in to the API Store (https://<hostname>:9443/store) with the admin/admin credentials and click the PizzaShackAPI API.
7. Select the default application and an available tier, and click Subscribe.
8. When the subscription is successful, click View Subscriptions on the information message that appears.

You have now successfully subscribed to an API. Let's generate a distribution for this API.

Generate a distribution for a single API

It is possible to create a Microgateway distribution for a single API that is in the Published state.

For details on how to create a Microgateway distribution for a group of APIs, see Grouping APIs with Labels.

1. Navigate to a preferred workspace folder using the command line.
   This location is used to run the Microgateway commands and to generate Microgateway artifacts.
2. Set up a project using the command given below,
2.

micro-gw setup <project_name> -a <API_name> -v <version>

When the tool requests for the username and password, use admin in both instances. Use the default values for the other parameters by pressing Enter.

Here is an example:

```
$ micro-gw setup pizzashack-project -a PizzaShackAPI -v 1.0.0
Enter Username: admin
Enter Password for admin: 
Enter APIM base URL [https://localhost:9443/]: 
Enter Trust store location: [lib/platform/bre/security/ballerinaTruststore.p12]
Enter Trust store password: [ use default? ]
Setting up project pizzashack-project is successful.
```

When the above command is issued, the tool connects with API Manager REST APIs and retrieves the API specified above. The source artifacts are generated in the current folder location.

The folder structure looks similar to the following,

```
./pizzashack-project
  conf
  deployment-config.toml
  src
  extension_filter.bal
  listeners.bal
  PizzaShackAPI_1_0_0.bal
  policies
  application_10PerMin.bal
  application_20PerMin.bal
  ...
  subscription_Bronze.bal
  subscription_Gold.bal
  ...
  subscription_Unauthenticated.bal
  throttle_policy_initializer.bal
  target
```

If you re-run the setup command while you already have a project named pizzashack-project in the current working directory, you receive an error similar to the following:
3. Build the microgateway distribution for the project using the following command:

```
$ micro-gw build <project_name>
```

Here is an example:

```
$ micro-gw build pizzashack-project
Build successful for the project - pizzashack-project
```

Once the above command is executed, the generated source files are built and a Microgateway distribution is created in the `target` folder.

The name of the distribution will have the format `micro-gw-<project_name>.zip`. In the above example, the name will be `micro-gw-pizzashack-project.zip`.

4. Next, unzip the `micro-gw-pizzashack-project.zip` and run the `gateway` shell script inside the `bin` folder of the extracted zip using the following command:

```
bash gateway
```

The Microgateway will now start for the Pizzashack API. For example,
You have successfully generated and started a Microgateway distribution for the Pizzashack API.

**Generate a JWT token and invoke the API**

Once you start the Microgateway, you can use a JWT or an OAuth2 token to invoke the API. For the invocation URL, you can use either https://localhost:9095/ (HTTPS) or http://localhost:9090/ (HTTP).

A signed JWT token is a self-contained access token that can be validated by the Microgateway itself without connecting to any external party. In contrast, a typical OAuth2 token needs to be validated by connecting to the Key Manager. This is one of the key advantages of the Microgateway, if you need to run it in offline mode.

1. Log in to the API Store, click **Applications** and choose to edit the Default Application. Set the token type to JWT.

   ![Edit Application](image)

   **Name**: DefaultApplication

   **Per Token Quota**: Unlimited

   **Token Type**: JWT

   **Description**: JWT token type can be used with micro gateways only

2. Click the **Production Keys** tab and click **Generate Keys** to generate an access token to invoke the API.
3. Invoke the API using the JWT token using the following command:

```
curl -k -l -H "Authorization: Bearer <JWT_TOKEN>" https://localhost:9095/pizzashack/1.0.0/menu
```

For example,

```
curl -k -l -H "Authorization: BearereyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJodHRwczpcL1wvbG9jYWxob3N0c1wvbG9jYWxob3N0X2N0b3JpdGVcL3N0b3JpdGlvbiIsImNvdW50YWxvc1wiOiJodHRwczpcL1wvbG9jYWxob3N0c1wvbG9jYWxob3N0X2N0b3JpdGVcL3N0b3JpdGlvbiIsInN1YnNjcmliZXJUZW5hbnQIb3JpZ2luY3RvZ3MiIiwiZXhwIjoxNTI4ODAyODgwMTEwLCJpYXQiOjEwMDgyNTMzNzYsIm5vbmVsc2l0aSI6MTYwMjI4ODQ2OTB9.bXg1y67tVzO7Z8fUW8WETpHr8ZG2mgQ7OOGsS0dW4KzvZPnR2GnF2S3AhGv4DnGUJ42EiQ8dS4mXv63JjPoiw" https://localhost:9095/pizzashack/1.0.0/menu
```

You receive a response similar to the following:
Alternatively, you can invoke the API using an OAuth2 token.

Invoking the API using an OAuth2 token is similar to the usual API invocation using the standard API Manager Gateway by generating an access token from the API Store. A sample cURL command is given below:

curl -k -i -H "Authorization: Bearer 20ac019e-16a7-3ba5-8940-7d42c7e56326" https://localhost:9095/pizzashack/1.0.0/menu

You receive a response similar to the following:
HTTP/1.1 200 OK
Date: Thu, 26 Jul 2018 16:52:38 GMT
Content-Type: application/json
Transfer-Encoding: chunked
server: WSO2 Carbon Server

["name":"BBQ Chicken Bacon","description":"Grilled white chicken, hickory-smoked bacon and fresh sliced onions in barbecue sauce","price":"13.99","icon":"/images/6.png"],
"name":"Chicken Parmesan","description":"Grilled chicken, fresh tomatoes, feta and mozzarella cheese","price":"9.99","icon":"/images/1.png"],
"name":"Chilly Chicken Cordon Bleu","description":"Spinash Alfredo sauce topped with grilled chicken, ham, onions and mozzarella","price":"17.99","icon":"/images/10.png"],
"name":"Double Bacon 6Cheese","description":"Hickory-smoked bacon, Julienne cut Canadian bacon, Parmesan, mozzarella, Romano, Asiago and and Fontina cheese","price":"16.99","icon":"/images/9.png"],
"name":"Garden Fresh","description":"Slices onions and green peppers, gourmet mushrooms, black olives and ripe Roma tomatoes","price":"9.99","icon":"/images/3.png"],
"name":"Grilled Chicken Club","description":"Grilled white chicken, hickory-smoked bacon and fresh sliced onions topped with Roma tomatoes","price":"13.99","icon":"/images/8.png"],
"name":"Hawaiian BBQ Chicken","description":"Grilled white chicken, hickory-smoked bacon, barbecue sauce topped with sweet pine-apple","price":"27.99","icon":"/images/7.png"],
"name":"Spicy Italian","description":"Pepperoni and a double portion of spicy Italian sausage","price":"13.99","icon":"/images/2.png"],
"name":"Spinach Alfredo","description":"Rich and creamy blend of spinach and garlic Parmesan with Alfredo sauce","price":"27.99","icon":"/images/5.png"],
"name":"Tuscan Six Cheese","description":"Six cheese blend of mozzarella, Parmesan, Romano, Asiago and Fontina","price":"11.99","icon":"/images/4.png"]}