



# WSO2 API Manager 4.2.0 Fundamentals

Getting Started with Developer Portal

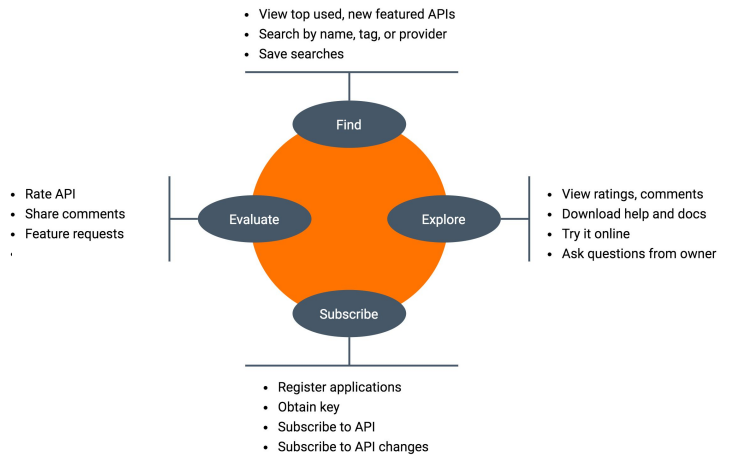


WSO2 Training

CC by 4.0

## What is API Developer Portal?

- React Web Application
- API Publishers can
  - ◉ Host APIs
  - ◉ Advertise APIs
  - ◉ Enable Recommendation
- API Consumers can
  - ◉ Self Register
  - ◉ Discover APIs
  - ◉ Evaluate APIs
  - ◉ Subscribe to APIs
  - ◉ Consume APIs



API Developer Portal is a Web Application which enables API consumers to register and evaluate APIs.

## Discover APIs

- Anonymous (Public) View
- Logged in View
- Requirements for an API to be displayed in the Developer Portal
  - ⦿ API should be in Published state
  - ⦿ API should be indexed in solr
  - ⦿ API should be either a public API or the logged in user should have permission to view the API
- Search APIs by
  - ⦿ API Name, Provider, Version, Context, Status, Description, Tags, API Category and Custom Property. (prefix based search e.g., name:Pizza)
  - ⦿ Unified content search (default)



## The API search facility

Search for an API by,

- API name
- API version
- API provider
- Context
- API status
- Description
- API Category
- Tag
- Custom property

Link - [API Search](#)



# Discover APIs

DEVELOPER PORTAL

API MANAGER

APIs

Applications

All

Search APIs

ADMIN

APIs

Cu

customer-info

By: admin

1.0.0

Version

/customerinfo

Context

★★★★★

0.0/5.0 (0users)

Cu

CustomerLeasing

By: admin

1.0.0

Version

/customerleasing

Context

★★★★★

0.0/5.0 (0users)

leasing

By: admin

1.0.0

Version

/leasing

Context

★★★★★

0.0/5.0 (0users)

Search Options

1. Content [ Default ]

2. Name [ Syntax - name:xxxx ]

3. By API Provider [ Syntax - provider:xxxx ]

4. By API Version [ Syntax - version:xxxx ]

5. By Context [ Syntax - context:xxxx ]

6. By Description [ Syntax - description:xxxx ]

7. By Tags [ Syntax - tags:xxxx ]

8. By API Category [ Syntax - api-category:xxxx ]

9. By API Properties [Syntax - property\_name:property\_value]

The search feature can be used to search APIs, API documentation, tags etc.



# APIs and Applications



# API Listing

WCO API MANAGER

DEVELOPER PORTAL

APIs

Applications

All

Search APIs

ADMIN

APIs

Cu

customer-info

By: admin

1.0.0

Version

/customerinfo

Context

★★★★★

0.0/5.0 (Users)

Cu

CustomerLeasing

By: admin

1.0.0

Version

/customerleasing

Context

★★★★★

0.0/5.0 (Users)

Le

leasing

By: admin

1.0.0

Version

/leasing

Context

★★★★★

0.0/5.0 (Users)



# API Overview

The screenshot shows the 'API Overview' page in the API Manager. The page has a blue header with the API Manager logo, navigation tabs (Overview, Subscriptions, Try Out, Comments, Documentation, SDKs), and a search bar. The main content area displays information for the 'customer-info' API, including its version (1.0.0), author (admin), and a 'TRY OUT' button. A 'Business Plans' section shows a 'Silver' plan with a limit of 2000 requests per minute. A 'Subscriptions' table lists the 'DefaultApplication' with a 'Silver' throttling tier and an 'UNBLOCKED' status. A 'Comments' section shows a comment from 'admin' stating 'Excellent!' with options to delete or reply. The left sidebar contains navigation links for Overview, Subscriptions, Try Out, Comments, Documentation, and SDKs. The bottom right corner features a yellow circle with the number 8.

API Manager Overview

customer-info

Version: 1.0.0 | By: admin

URL: <https://localhost:8243/customerinfo/1.0.0> [TRY OUT](#)

Business Plans

Silver

2000 Requests/min

Subscriptions

Application Name	Throttling Tier	Application Status
DefaultApplication	Silver	UNBLOCKED

Comments

[Write A New Comment](#)

admin  
2 minutes ago  
Excellent!  
[DELETE](#) [REPLY](#)

Tags: customer

Source: [Download Swagger](#)

Clicking on an API Thumbnail will take you to the API Overview.

This view shows the basic information about the API, Ratings, the deployed environments, tags etc.

- **Subscriptions** Information on the existing subscriptions and subscribe to new Application
- **Try Out** Test the API with the integrated swagger/ GraphQL consoles
- **Comments** View and comment on the API
- **Documentation** View the API Documentation
- **SDKs** Download Client SDKs for the API



## Applications

- Logical representation of an application such as a mobile app, webapp, device, etc.
- Generate and use a single key for multiple APIs.
- Subscribe multiple times to a single API with different Service Level Agreements (SLAs)/ business plans which operate on per access token basis
- Comes with a pre-created default application, which allows unlimited access by default.





# Application Creation



Applications

ADD NEW APPLICATION

An application is a logical collection of APIs. Applications allow you to use a single access token to invoke a collection of APIs and to subscribe to one API multiple times and allows unlimited access by default.

Search						SEARCH
Name ↑	Owner	Policy	Workflow Status	Subscriptions	Actions	
DefaultApplication	admin	Unlimited	ACTIVE	1	 	



# Application Creation

## ● Create Application Form

### Create an application

Create an application providing name and quota parameters. Description is optional.  
Required fields are marked with an asterisk ( \* )

Application Name \*

PizzaShackApp

Enter a name to identify the Application. You will be able to pick this application when subscribing to APIs

Shared Quota for Application Tokens \*

10PerMin

Assign API request quota per access token. Allocated quota will be shared among all the subscribed APIs of the application.

Application Description

PizzaShack Application

( 490 ) characters remaining

SAVE

CANCEL



# Application Overview

- Application Overview

DEVELOPER PORTAL

API MANAGER

APIs

Applications

All

Search APIs

ADMIN

Overview

Production Keys

OAuth2 Tokens

API Key

Sandbox Keys

OAuth2 Tokens

API Key

Subscriptions

PizzaShackApp

0 Subscriptions

EDIT

DELETE

Description	PizzaShack Application
Business Plan	10PerMin (Allows 10 request per minute)
Workflow Status	APPROVED
Application Owner	admin



# Application Listing

- Application Listing

DEVELOPER PORTAL

API MANAGER

APIs

Applications

All

Search APIs

ADMIN

Applications

ADD NEW APPLICATION

An application is a logical collection of APIs. Applications allow you to use a single access token to invoke a collection of APIs and to subscribe to one API multiple times and allows unlimited access by default.

Search						SEARCH
Name ↑	Owner	Policy	Workflow Status	Subscriptions	Actions	
DefaultApplication	admin	Unlimited	ACTIVE	1		
PizzaShackApp	admin	10PerMin	ACTIVE	0		

## Generate Keys

- Can be Production or Sandbox keys.
- Generates **consumer-key** and **consumer-secret** pair for the application.

The screenshot displays the 'API Manager' console interface. The left sidebar contains navigation links: Overview, Production Keys, OAuth2 Tokens, API Key, Sandbox Keys, OAuth2 Tokens, API Key, and Subscriptions. The main content area is titled 'PizzaShackApp' and shows the 'Production OAuth2 Keys' configuration. The 'Key and Secret' section indicates that the key and secret are not generated for this application. The 'Key Configuration' section includes fields for 'Token Endpoint' (https://localhost:8443/oauth2/token) and 'Revoke Endpoint' (https://localhost:8443/oauth2/revoke). The 'Grant Types' section lists various grant types with checkboxes: Refresh Token, SAML2, Password, Client Credentials, OAuth2/OAuth, Device Code, Code, urn:ietf:params:oauth:grant-type:token-exchange, and JWT. The 'Callback URL' field is empty. The 'Application Access Token Expiry Time' is set to 'N/A'. The 'User Access Token Expiry Time' is set to 'N/A'. The 'Refresh Token Expiry Time' is set to 'N/A'. The 'Id Token Expiry Time' is set to 'N/A'.

API MANAGER Applications All Search APIs

Overview Production Keys OAuth2 Tokens API Key Sandbox Keys OAuth2 Tokens API Key Subscriptions

PizzaShackApp

Production OAuth2 Keys

Key and Secret

Production Key and Secret is not generated for this application.

Key Configuration

Token Endpoint https://localhost:8443/oauth2/token

Revoke Endpoint https://localhost:8443/oauth2/revoke

Grant Types

☐ Refresh Token ☐ SAML2 ☒ Password ☒ Client Credentials ☐ OAuth2/OAuth ☐ Device Code ☐ Code ☐ urn:ietf:params:oauth:grant-type:token-exchange ☐ JWT

The application can use the following grant types to generate Access Tokens, based on the application requirements you can enable or disable grant types for this application.

Callback URL

Callback URL is a redirection URL in the client application which is used by the authorization server to send the client's user agent (usually web browser) back after granting access.

Application Access Token Expiry Time

N/A

Type Application Access Token Expiry Time

User Access Token Expiry Time

N/A

Type User Access Token Expiry Time

Refresh Token Expiry Time

N/A

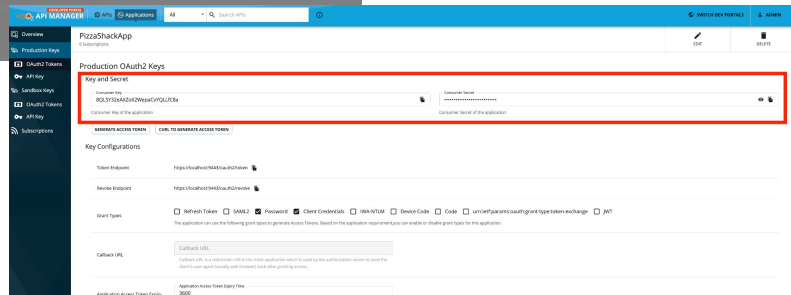
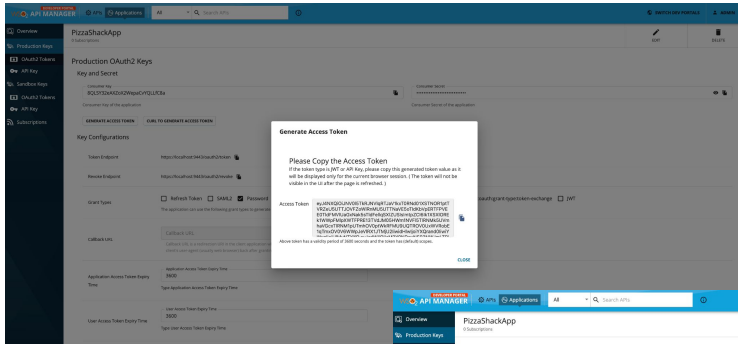
Type Refresh Token Expiry Time

Id Token Expiry Time

N/A

Type Id Token Expiry Time

# Generate Keys



## Supported Grant Types

- Supported Grant types:
  - ◉ Password Grant
  - ◉ Client Credentials Grant
  - ◉ Authorization Code Grant
  - ◉ Refresh Token Grant
  - ◉ JWT Grant
  - ◉ SAML Extension Grant
  - ◉ Kerberos OAuth2 Grant
  - ◉ NTLM Grant



When generating access tokens, select required grant type. By default all types are selected (except Authorization Code).

Note: The Implicit Grant has been removed from WSO2 API Manager 4.0.0. This has been done since the OAuth 2.1.0 has removed/discouraged the use of the implicit grant type due to security concerns.




## Supported Grant Types

**Password (legacy):** Used by first-party clients to exchange a user's credentials for an access token. Since this involves the client asking the user for their password, it should not be used by third party clients. In this flow, the user's username and password are exchanged directly for an access token.

```
POST /oauth2/token HTTP/1.1
Host: authorization-server.com

grant_type=password
&username=user@example.com
&password=1234luggage
&client_id=xxxxxxxxxx
&client_secret=xxxxxxxxxx
```



```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "access_token": "MTQ0NjJkZmQ5OTM2NDE1ZTZjNGZmZjI3",
  "token_type": "bearer",
  "expires_in": 3600,
  "refresh_token": "IwOGYzYTlmM2YxOTQ5MGE3YmNmMDFkNTVh",
  "scope": "create"
}
```




## Supported Grant Types

**Client Credentials:** Used by clients to obtain an access token outside of the context of a user. This is typically used by clients to access resources about themselves rather than to access a user's resources.

```
POST oauth2/token HTTP/1.1
Host: authorization-server.com

grant_type=client_credentials
&client_id=xxxxxxxxxx
&client_secret=xxxxxxxxxx
```



```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "access_token": "MTQ0NjJkZmQ5OTM2NDE1ZTZjNGZmZjI3",
  "token_type": "bearer",
  "expires_in": 3600,
  "refresh_token": "IwOGYzYTlmM2YxOTQ5MGE3YmNmMDFkNTVh",
  "scope": "create"
}
```



## Supported Grant Types

**Authorization Code:** Used by confidential and public clients to exchange an authorization code for an access token. After the user returns to the client via the redirect URL, the application will get the authorization code from the URL and use it to request an access token.

```
POST /oauth2/token HTTP/1.1
Host: authorization-server.com

grant_type=authorization_code
&code=xxxxxxxxxx
&redirect_uri=https://example-app.com/redirect
&client_id=xxxxxxxxxx
&client_secret=xxxxxxxxxx

HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "access_token": "MTQ0NjJkZmQ5OTM2NDE1ZTZjNGZmZjI3",
  "token_type": "bearer",
  "expires_in": 3600,
  "refresh_token": "IwOGYzYTlmM2YxOTQ5MGE3YmNmMDFkNTVh",
  "scope": "create"
}
```

Demo: <https://www.oauth.com/playground/authorization-code.html>




## Supported Grant Types

**Refresh Token:** Used by clients to exchange a refresh token for an access token when the access token has expired. This allows clients to continue to have a valid access token without further interaction with the user.

```
POST /oauth2/token HTTP/1.1
Host: authorization-server.com

grant_type=refresh_token
&refresh_token=xxxxxxxxxxx
&client_id=xxxxxxxxxxx
&client_secret=xxxxxxxxxxx
```



```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "access_token": "MTQ0NjJkZmQ5OTM2NDE1ZTZjNGZmZjI3",
  "token_type": "bearer",
  "expires_in": 3600,
  "refresh_token": "IwOGYzYTlmM2YxOTQ5MGE3YmNmMDFkNTVh",
  "scope": "create"
}
```

## Access Tokens

- Application Access Tokens : Tokens to identify and authenticate an entire application
- User Access Tokens : Tokens to identify the final user of an application.

## JWT Token



JSON Web Tokens (JWT) are self contained tokens where the user information is integrated in the token it self.

The token has 3 main sections separated by "." (fullstop) character.

- **Header** Contains algorithm, token type information
- **Body** Contains the user and token data. User claims, application and subscription details (for self contained tokens)
- **Signature** Signature is created by encrypting the header and the body with the private certificate of the server.

# API Key

The screenshot shows the AWS IAM console interface. On the left sidebar, under 'Overview', the 'Production Keys' link is selected. The main panel shows details for a user named 'PizzaShackApp'. Under 'Key Restrictions', three options are available: 'None' (selected), 'IP Addresses', and 'HTTP Referrers (Web Sites)'. A red rectangular box highlights the 'GENERATE KEY' button. Below this button, a small note states: 'Use the Generate Key button to generate a self-contained JWT token.'

A modal dialog titled 'Generate API Key' is open in the foreground. It contains the following text:

- Generate API Key**
- Please Copy the API Key
- If the token type is JWT or API Key, please copy this generated token value as it will be displayed only for the current browser session. (The token will not be visible in the UI after the page is refreshed.)

The modal displays the generated API Key:

```
API Key eyJANXQIOUJOVGRtWmpNNFpEazNOalkwWXpNUt1WmTPR  
GdSTVBfMO1XWxGORELU1VdsBfpEZrOemMOWKES9SIsI  
mtPZCIElmdhdGV3YXIY2YvdGhmaWNhdGVFYWxpPXMLECJ  
0eXAIOJKV1QLLCjhbGciOiJSUzI1Ni9.  
eyJzdWIiOiJhZG1pbkBYXUp24uc3VwZWZILCJhbHNaWWhndGVibGlyajdzZic6I  
mfKwWuliwdGlpcFibSRNWHwZSiBnNjQwLWidGlci6lEwUG
```

Below the key, a note indicates: 'Above token has a validity period of -1 seconds.' A 'CLOSE' button is located at the bottom right of the modal.

If your API is secured with API Key authentication, you can generate an API Key to invoke the api.

API Key is a JWT type self contained token which included information about the subscription, user etc.

## API Key now restricts using the following

- IP address restriction
- HTTP referer restriction

## Subscribing and Invoking the API



## Subscribing an API

There are three (03) ways an API can be subscribed.

1. Using the API Subscription and key generation wizard.
  - a. A new application will be created and subscribed
2. Subscribing to an existing application.
  - a. Select an existing application to subscribe
3. Using Try Out in overview page.
  - a. An API subscription will be added for the Default Application.





## Subscription and Key Generation Wizard.

- Navigate to Subscriptions tab.
- Click on the Subscription and Key Generation Wizard.
- This will guide you with creating a new application, subscribing to the API and generating tokens.

The screenshot shows the 'Subscription & Key Generation Wizard' interface. On the left is a dark sidebar with navigation links: Overview, Subscriptions (highlighted), Try Out, Comments, Documentation, and SDKs. The main content area has a breadcrumb trail: Subscriptions > customer-info > Subscriptions. Below this is the title 'Subscription & Key Generation Wizard'. A progress bar at the top shows five steps: 1. Create application (active), 2. Subscribe to new application, 3. Generate Keys, 4. Generate Access Token, and 5. Copy Access Token. The 'Create application' step contains the following form fields: 'Application Name \*' with a text input containing 'My Application' and a placeholder instruction 'Enter a name to identify the Application. You will be able to pick this application when subscribing to APIs'; 'Shared Quota for Application Tokens \*' with a dropdown menu set to '100PerMin' and a note 'Assign API request quota per access token. Allocated quota will be shared among all the subscribed APIs of the application.'; and 'Application Description' with a text area and a character count '(512) characters remaining'. At the bottom left are 'CANCEL' and 'NEXT' buttons.



1 Create application 2 Subscribe to new application 3 Generate Keys 4 Generate Access Token 5 Copy Access Token

Application Name \*

TestApplication

Enter a name to identify the Application. You will be able to pick this application when subscribing to AWS

Shared Quota for Application Tokens \*

1000/Mon

Assign API request quota per

Subscriptions | Customer-Info > Subscriptions

Application Description

Test Application

Subscription & Key Generation Wizard

Create application

Subscribe to new application

Generate Keys

Generate Access Token

Copy Access Token

The screenshot shows the 'Subscription and Key Generation Wizard' in the AWS IAM console. The wizard is at the 'Create application' step. The application is named 'Silver' and is a 'SaaS' application. The wizard steps are: 1. Create application (current), 2. Subscribe to new application, 3. Generate keys, 4. Generate Access Token, 5. Copy Access Token. The 'Key Configuration' section shows a table with one key named 'Key String' and a 'Key Manager' of type 'AWS Key Management Service'.

**Subscriptions** | [Customer-only](#) > Subscriptions

Subscription & Key Generation Wizard

1 Create application 2 **Subscribe to new application** 3 Generate Keys 4 Generate Access Token

Generate Access Token for SANSIDON environment

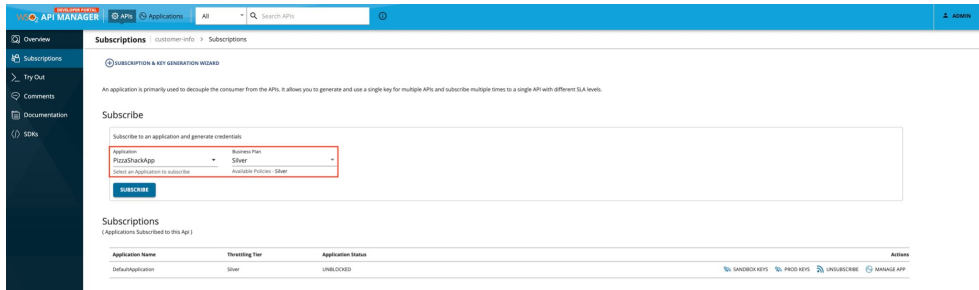
**Please Copy the Access Token**  
If the token type is JWT or API Key, please copy this generated token value as it will be displayed only for the current browser session. (The token will not be visible in the UI after the page is refreshed.)

[illegible]

Below token has a validity period of 3600 seconds and the token has default scope.

## Subscribing to an Existing Application

- Open an API, and go to the Subscriptions page by clicking the Subscriptions item in the left menu.
- Under the Subscribe section, select the Application which is created before and click Subscribe.



The screenshot shows the 'Subscriptions' page in the API Manager. The left sidebar contains navigation links: Overview, Subscriptions (active), Try Out, Comments, Documentation, and SDKs. The main content area has a header 'Subscriptions' with a search bar and a 'Subscribe' button. Below this, there is a section titled 'Subscribe' with a form to select an application and a plan. The form shows 'PizzaShopApp' selected for the application and 'Silver' for the plan. A 'SUBSCRIBE' button is at the bottom of the form. Below the form, there is a table titled 'Subscriptions' showing the list of subscriptions.

Subscriptions

Subscribe to an application and generate credentials

Application: PizzaShopApp Plan: Silver

SUBSCRIBE

Subscriptions

Application Name	Throttling Tier	Application Status	Actions
DefaultApplication	Silver	UNBLOCKED	<a href="#">GENERATE KEYS</a> <a href="#">PREVIEW KEYS</a> <a href="#">UNSUBSCRIBE</a> <a href="#">MANAGE APP</a>



## Add Subscription to an API in the Application

The screenshot displays the API Manager interface. The top navigation bar includes 'API MANAGER', 'APIs', 'Applications', and a search bar. The left sidebar lists navigation options: Overview, Production Keys, OAuth2 Tokens, API Key, Sandbox Keys, OAuth2 Tokens, API Key, and Subscriptions. The main content area shows the 'TestApplication' overview with a 'Subscription Management' tab highlighted. A 'SUBSCRIBE API' button is visible. A modal titled 'Subscribe APIs' is open, showing a search bar and a table of APIs.

**Subscribe APIs** Search APIs

Displaying all APIs

Name	Version	Subscription Status
customer-info	1.0.0	Subscribed
CustomerLeasing	1.0.0	Unlimited <span>SUBSCRIBE</span>
leasing	1.0.0	Silver <span>SUBSCRIBE</span>

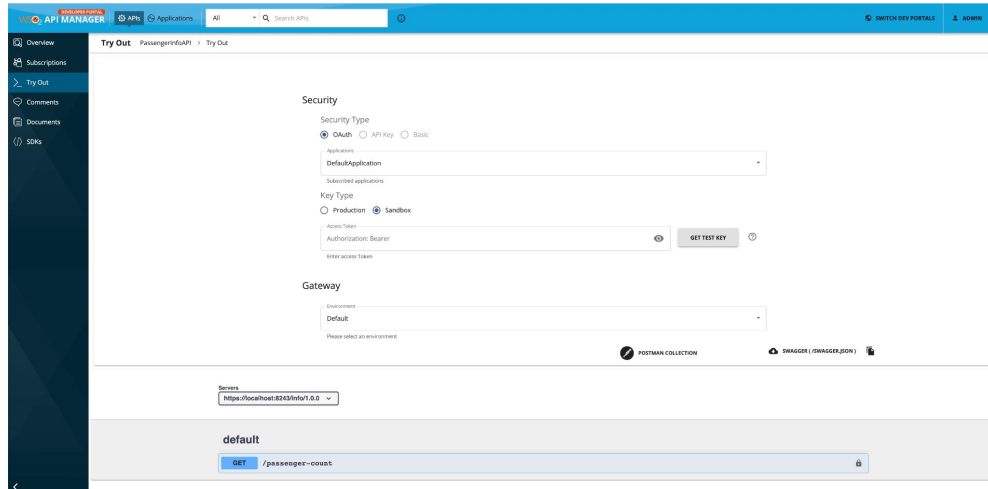
Items per page 10 1-3 of 3 < >

To add a subscription to the application,

- In the Applications tab, click on the Application Name and go to the Application overview page.
- Go to Subscriptions page by clicking the Subscriptions menu item.
- Click on Subscribe APIs
- From the APIs list, select a subscription policy and click on Subscribe to subscribe to the API.

## Invoke API

- API should be subscribed to an Application.
- Access token needs to be generated to access the API resources.



To Invoke the API,

- Click on Try Out menu item in the left menu.
- Paste the access token generated in the key generation step or click on get test key.

## Invoke an API

GET /menu

Return a list of available menu items

Parameters

No parameters

Execute

Responses

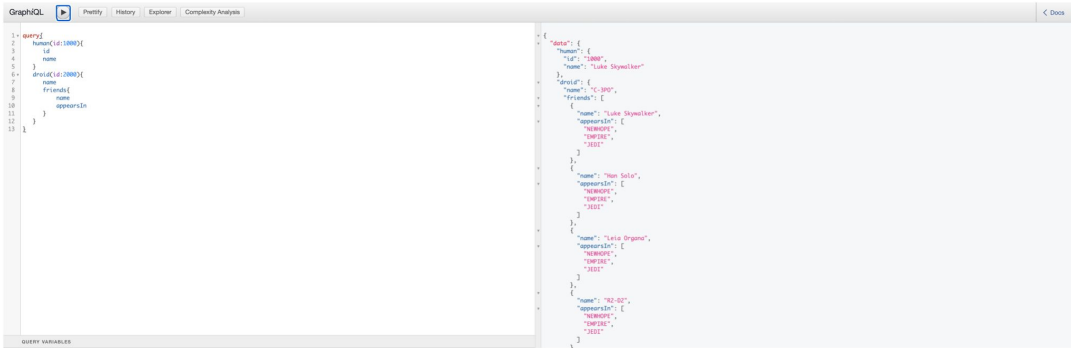
Code	Description	Links
200	OK. List of APIs is returned.	No links
Media type: application/json		
Example Value: Schema		
<pre>{   "id": "string",   "description": "string",   "name": "string",   "image": "string" }</pre>		
406	Not Acceptable. The requested media type is not supported	No links
Media type: application/json		
Example Value: Schema		
<pre>{   "message": "string",   "error": {     "message": "string",     "code": 0   },   "description": "string",   "code": 0,   "hostname": "string" }</pre>		

- Click on the API resource which needs to be invoked.
- Provide any parameters required (path, query or body, etc)
- Click Execute to send the request.

## Invoke an API

[illegible]

## Tryout GraphQL APIs



The screenshot shows the GraphQL tryout console with a query on the left and its JSON response on the right. The query is a nested query for a character's details. The response is a JSON object with a 'data' field containing the query results.

```
1 query {
2   character(id: 1000) {
3     id
4     name
5   }
6   character(id: 1000) {
7     name
8     friends {
9       name
10      appearsIn
11    }
12  }
13 }
```

```
{
  "data": {
    "character": {
      "id": "1000",
      "name": "Luke Skywalker"
    },
    "characters": {
      "id": "1000",
      "friends": [
        {
          "name": "Luke Skywalker",
          "appearsIn": [
            "THE FORCE AWAKENS",
            "THE LAST JEDI"
          ]
        },
        {
          "name": "Han Solo",
          "appearsIn": [
            "THE FORCE AWAKENS",
            "THE LAST JEDI",
            "THE FORCE AWAKENS"
          ]
        },
        {
          "name": "Leia Organa",
          "appearsIn": [
            "THE FORCE AWAKENS",
            "THE LAST JEDI",
            "THE FORCE AWAKENS"
          ]
        },
        {
          "name": "R2-D2",
          "appearsIn": [
            "THE FORCE AWAKENS",
            "THE LAST JEDI",
            "THE FORCE AWAKENS"
          ]
        }
      ]
    }
  }
}
```

API Manager 4.0.0 Developer Portal is featured with an integrated GraphQL try out console where you can provide the query, invoke and test the API.

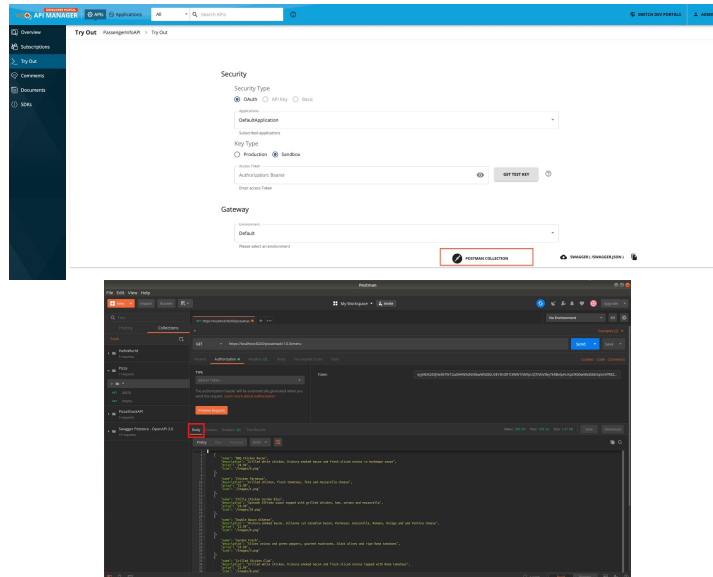
As in previous section, Create an application, subscribe and generate an access token for the GraphQL API.

Select the API and go to the Tryout page.

Using the integrated GraphQL tryout console, enter the GraphQL query and click Play.



## Try Out Using Postman



<https://apim.docs.wso2.com/en/4.2.0/consume/invoke-apis/invoke-apis-using-tools/try-out-using-postman/#try-out-using-postman>

# Application Sharing



## Application Sharing

- Enable application sharing in configure deployment.toml to use the feature.

```
[apim.devportal]
```

```
enable_application_sharing = true
```

- Register users with Organizations

WSO2 API MANAGER

### Create New Account

Fill in the form below to complete registration

First Name *	Last Name *
<input type="text" value="Jane"/>	<input type="text" value="Doe"/>
Password *	Confirm password *
<input type="password" value="*****"/>	<input type="password" value="*****"/>
Email *	
<input type="text" value="jane@test.com"/>	
Organization	
<input type="text" value="org1"/>	
Telephone	
IM	
Country	
Mobile	
URL	

When you sign in, we use a cookie in your browser to track your session.  
You can read our [Cookie Policy](#) for more information.

☒ I hereby confirm that I have read and understood the [Privacy Policy](#) \*

[Cancel](#) [Register](#)

Already have an account? [Sign in](#)

Application sharing enables subscribers to use the same OAuth application without creating new applications for their own.

## Application Sharing

- Create an Application with Application Groups defined.

### Create an application

Create an application providing name and quota parameters. Description is optional.  
Required fields are marked with an asterisk (\*)

Application Name \*

SharedApp

Enter a name to identify the Application. You will be able to pick this application when subscribing to APIs

Shared Quota for Application Tokens \*

10PerMin

Assign API request quota per access token. Allocated quota will be shared among all the subscribed APIs of the application.

Application Description

Shared Application

( 494 ) characters remaining

Application Groups

org1

Type a group and enter

SAVE

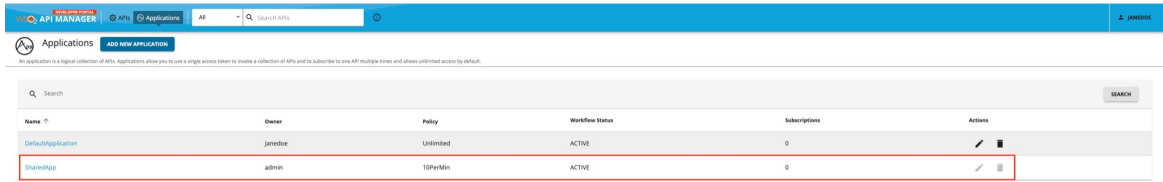
CANCEL







When creating the application, provide the organization/ list of organizations which the application needs to be shared.

## Application Sharing

- Users who belong to that specific organization can use that Application.
- Only owner of that application can generate Application keys.



The screenshot shows the 'Applications' section of an API Manager interface. At the top, there's a blue header with 'API MANAGER' and 'Applications' tabs. Below the header, a search bar and a 'SEARCH' button are visible. The main content area contains a table with columns: Name, Owner, Policy, Workflow Status, Subscriptions, and Actions. Two applications are listed: 'DefaultApplication' and 'SharedApp'. The 'SharedApp' row is highlighted with a red border. The 'Actions' column for 'SharedApp' contains a pencil icon and a trash can icon.

Name	Owner	Policy	Workflow Status	Subscriptions	Actions
DefaultApplication	jared@acme	Unlimited	ACTIVE	0	 
SharedApp	admin	10PerMin	ACTIVE	0	 

## Client SDKs

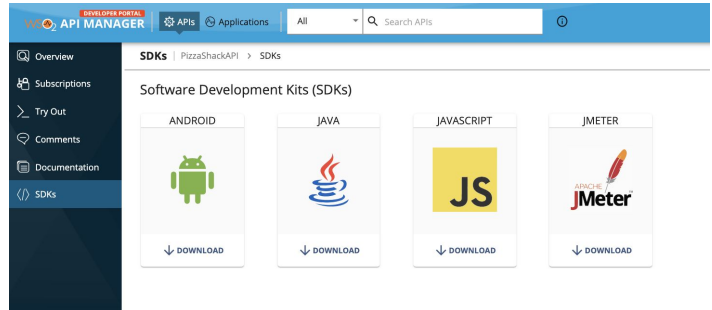


## Client SDKs

- With SDKs, developers can implement API Clients which can be used to invoke the particular API from different platforms (eg: android, web etc).
- The supported SDK platforms can be configured in the deployment.toml file.

```
[apim.sdk]
supported_languages = ["android",
"java", "scala", "csharp",
"dart", "flash", "groovy",
"javascript"]
```

Link - [Download Client SDK](#)





## Community Features





## Engage with the Community

The Developer Portal provides several useful features to build and nurture an active community of users as well as a developer community, for your APIs.

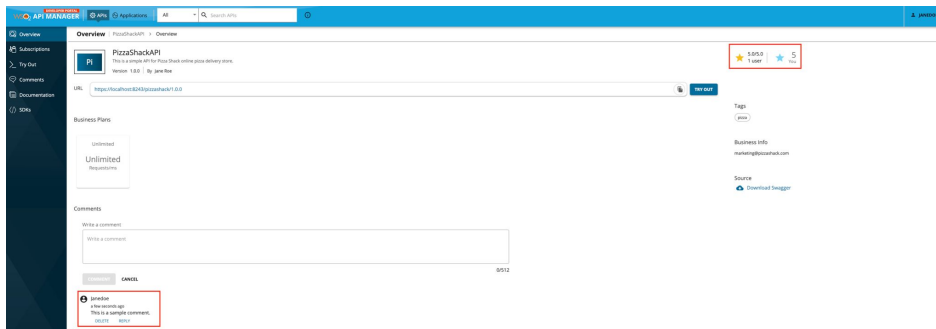
- Rate and comment
- Search facility

Link - [Using the Community Features](#)



## Rate and comment

Give insights to potential API consumers on the quality and usefulness of an API



Link - [Rate and Comment](#)





**Let's try it out!**

**Subscribing to APIs**  
**Invoking the API**

