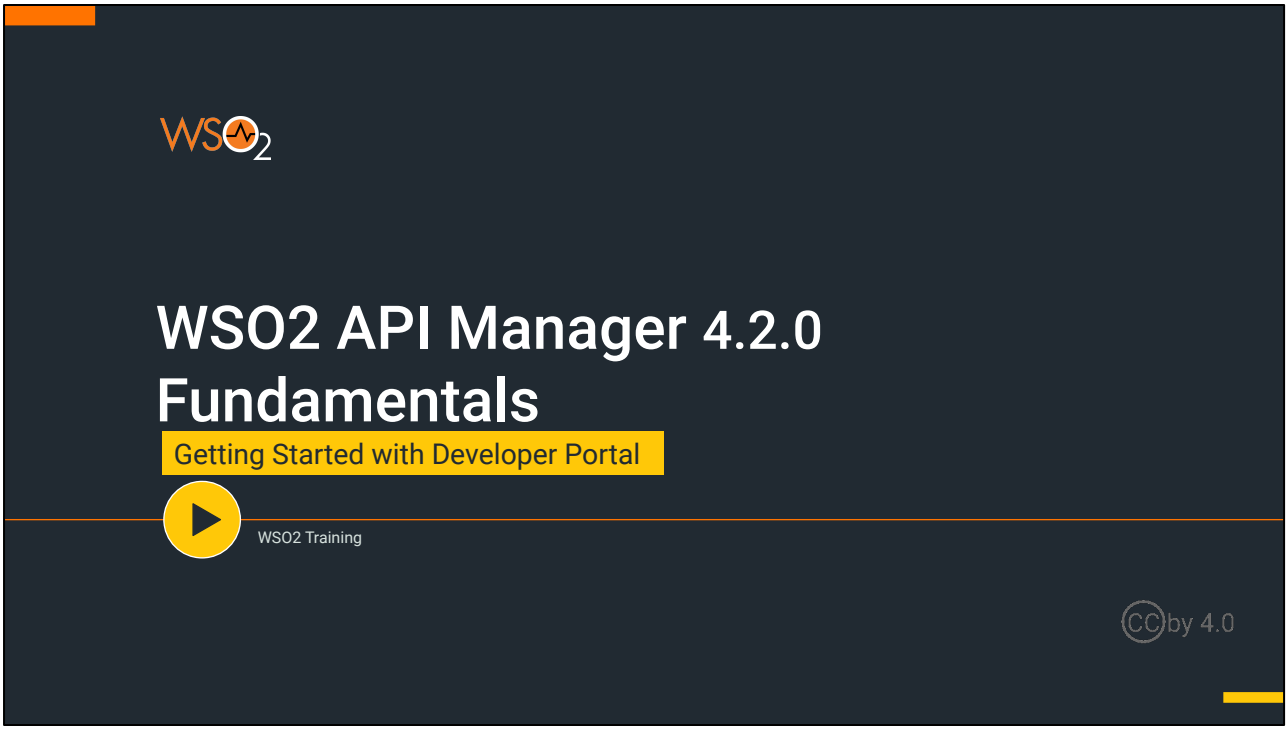
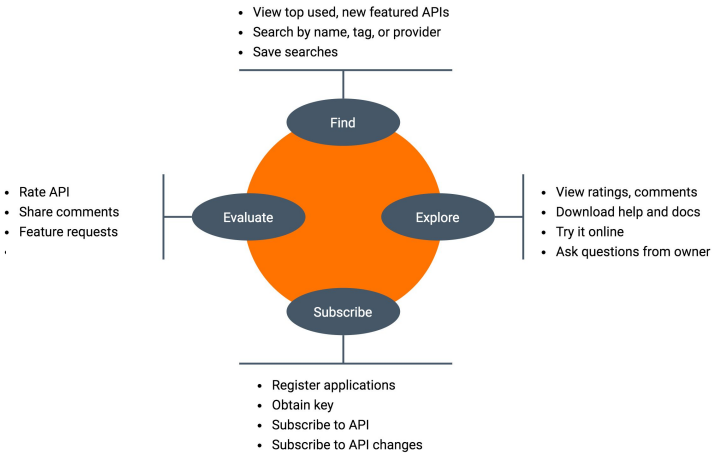

WSO2
WSO2 API Manager 4.2.0
Fundamentals
Getting Started with Developer Portal
WSO2 Training
CC by 4.0

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)

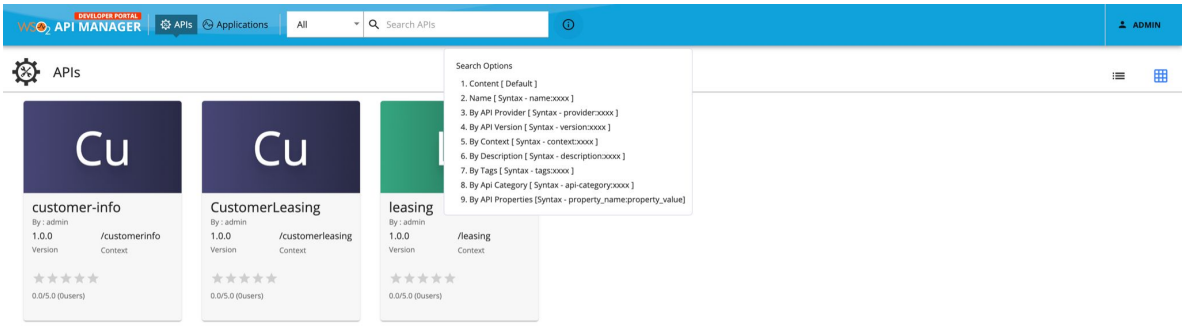
The API search facility

Search for an API by,

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

Link - API Search

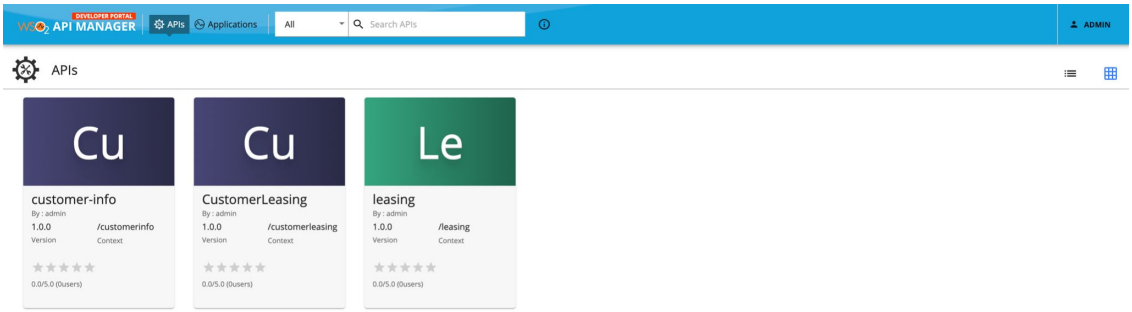
Discover APIs



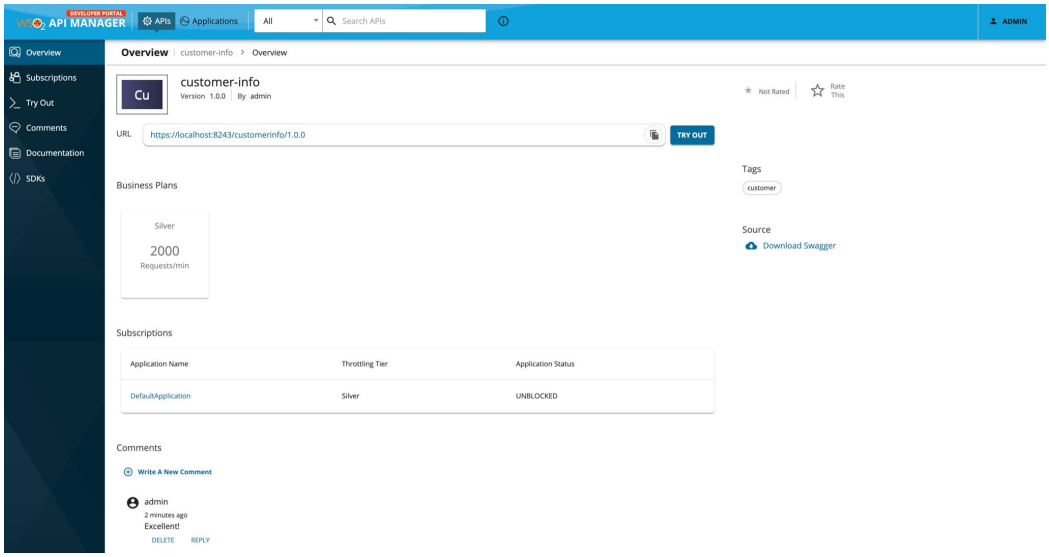
The search feature can be used to search APIs, API documentation, tags etc.

APIs and Applications

API Listing


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)
Le
leasing
By : admin
1.0.0
Version
/leasing
Context
0.0/5.0 (0users)

API Overview



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

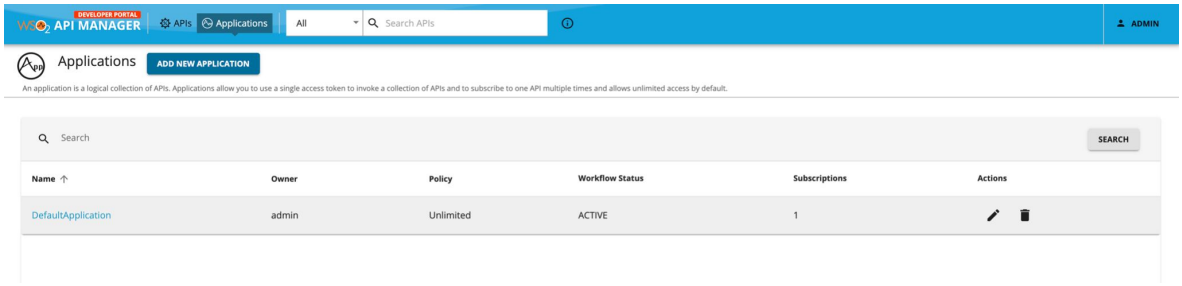
This view shows the basic information about the API, Ratings, the deployed environments, tags etc.

- **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

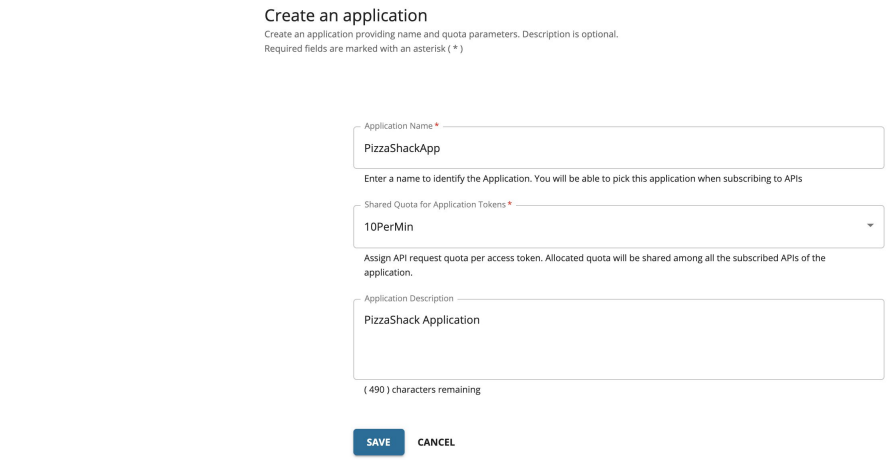
- Logical representation of an application such as a mobile app, webapp, device, etc.
- Generate and use a single key for multiple APIs.
- Subscribe multiple times to a single API with different Service Level Agreements (SLAs)/ business plans which operate on per access token basis
- Comes with a pre-created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

Application Creation



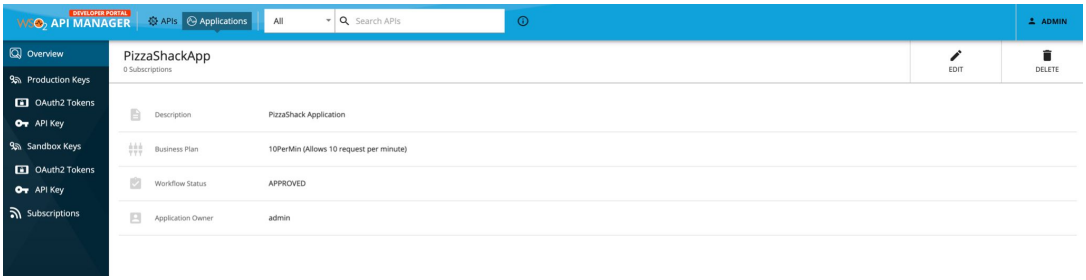
Application Creation

- Create Application Form



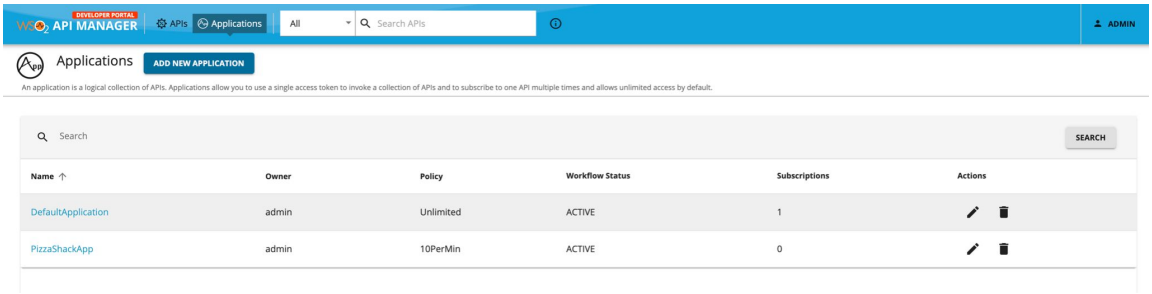
Application Overview

- Application Overview



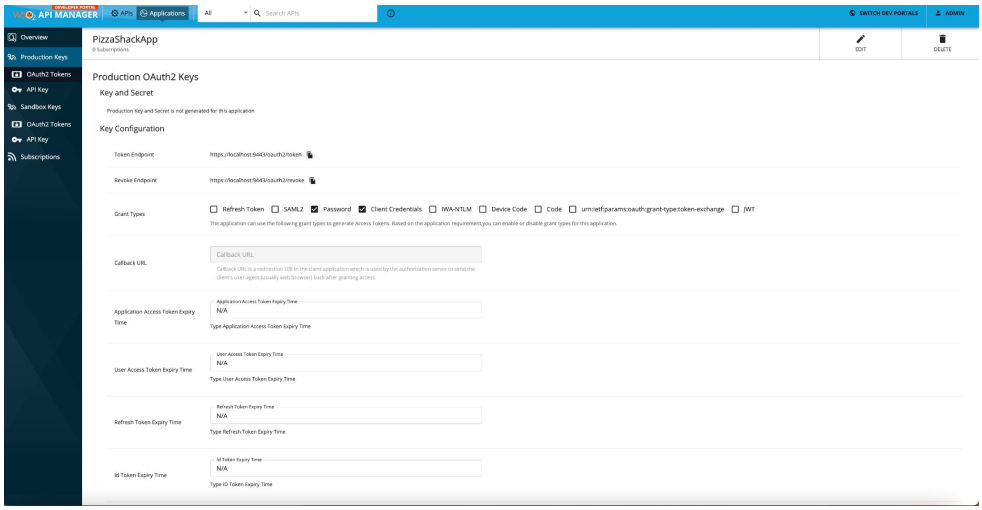
Application Listing

- Application Listing

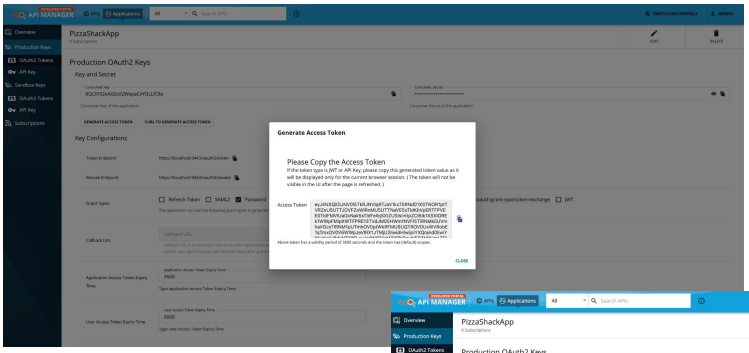


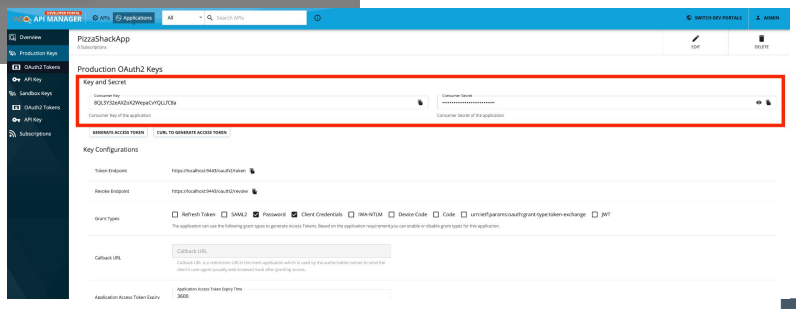
Generate Keys

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



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":"IwOGYzYTlmM2YxOTQ5MGE3YmNmMDFkNTVk",
  "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":"IwOGYzYTlmM2YxOTQ5MGE3YmNmMDFkNTVk",
  "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-ap
p.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":"IwOGYzYTlmM2YxOTQ5MGE3YmNmMDFkNTVk",
  "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=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":"IwOGYzYTlmM2YxOTQ5MGE3YmNmMDFkNTVk",
  "scope":"create"
}
```

Access Tokens

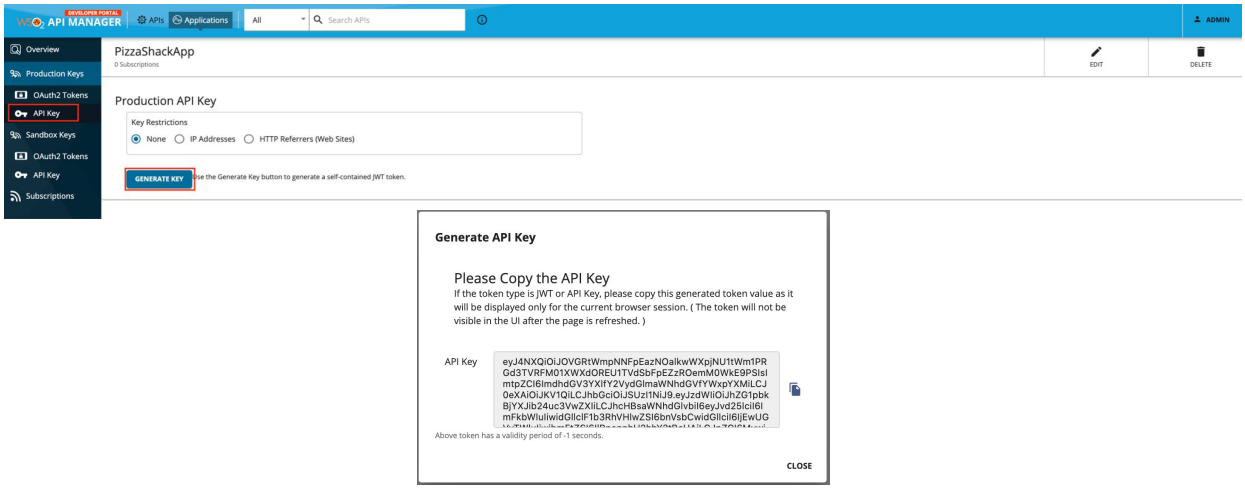
- Application Access Tokens : Tokens to identify and authenticate an entire application
- User Access Tokens : Tokens to identify the final user of an application.

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



Documentation Link - API Keys

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

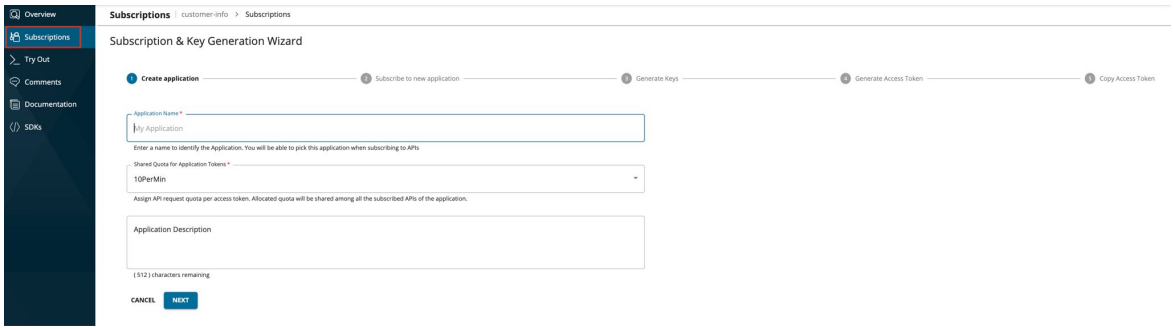
Subscribing an API

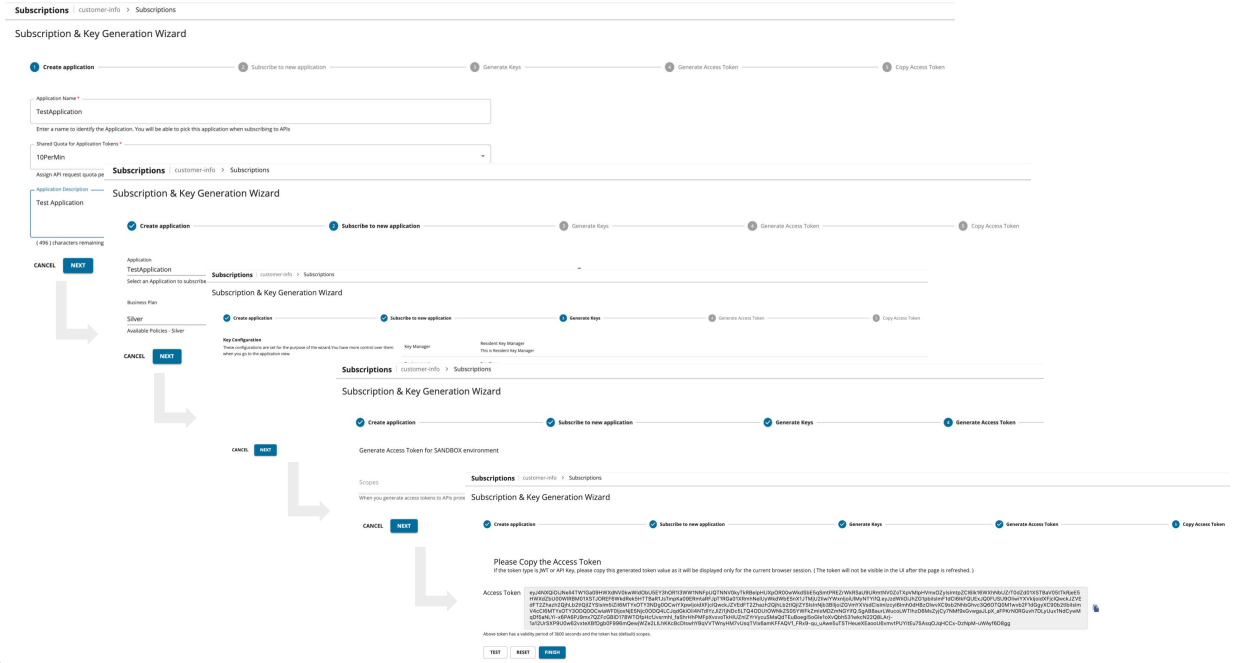
There are three (03) ways an API can be subscribed.

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.

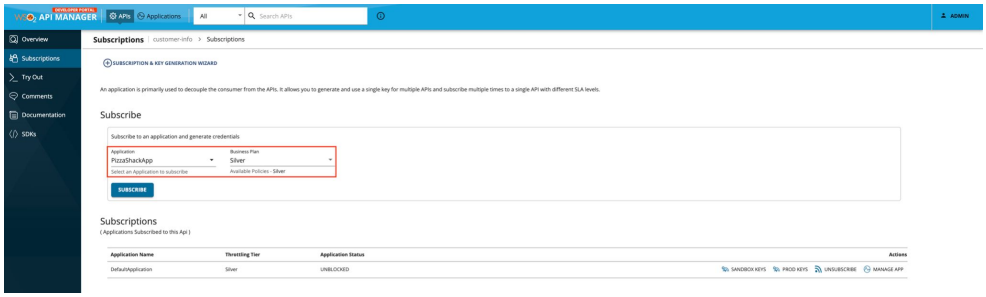
- Navigate to Subscriptions tab.
- Click on the Subscription and Key Generation Wizard.
- This will guide you with creating a new application, subscribing to the API and generating tokens.



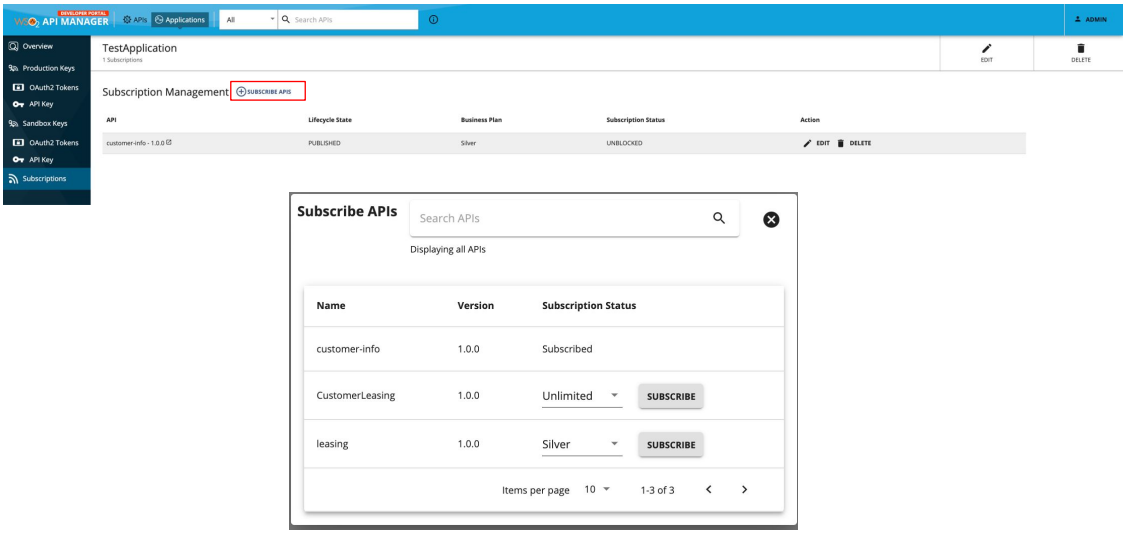

Subscriptions | customer-info > Subscriptions
Subscription & Key Generation Wizard
1 Create application
2 Subscribe to new application
3 Generate Keys
4 Generate Access Token
5 Copy Access Token
Application Name *
TestApplication
Enter a name to identify the Application. You will be able to pick this application when subscribing to APIs
Shared Quota for Application Tokens *
10PerMin
Assign API request quota pe
Application Description
Test Application
(-496) characters remaining
CANCEL
NEXT
Application
TestApplication
Select an Application to subscribe
Business Plan
Silver
Available Policies - Silver
Key Configuration
These configurations are set for the purpose of the wizard.You have more control over them when you go to the application view.
Key Manager
Resident Key Manager
This is Resident Key Manager
Generate Access Token for SANDBOX environment
Scopes
When you generate access tokens to APIs prote
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.)
Access Token
Above token has a validity period of 3600 seconds and the token has (default) scopes.
TEST
RESET
FINISH

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.



Add Subscription to an API in the Application

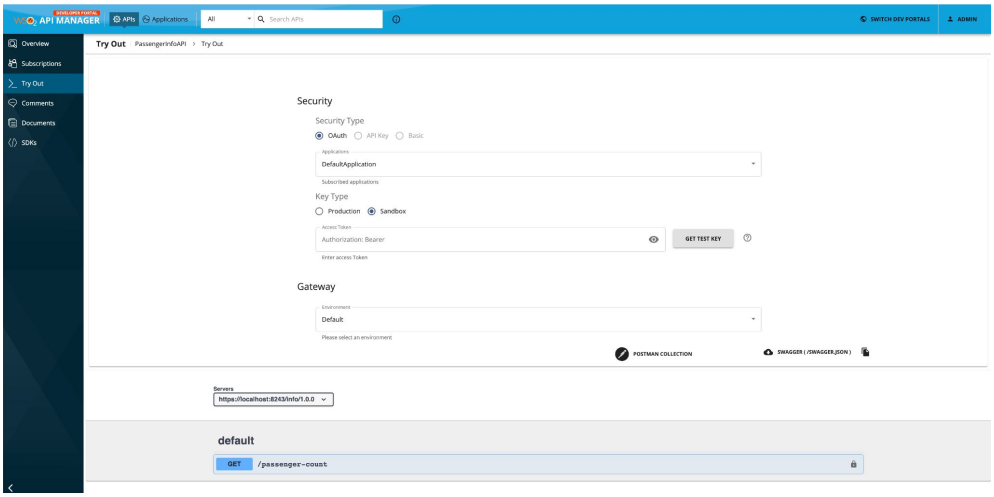


To add a subscription to the application,

- 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

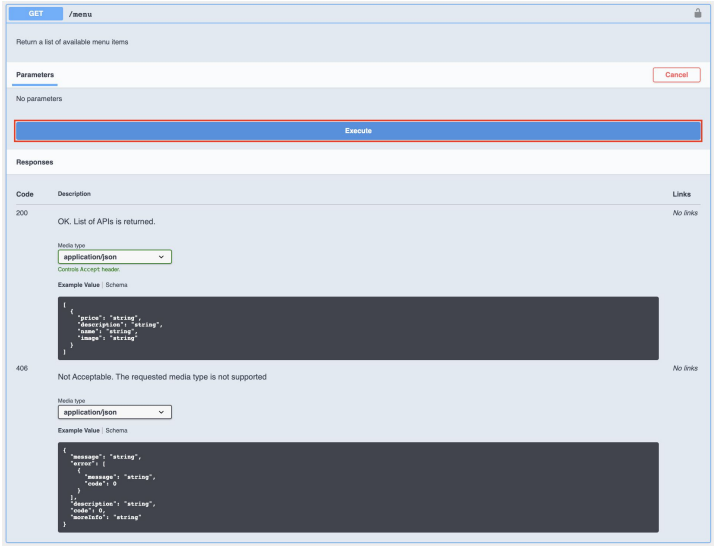
- API should be subscribed to an Application.
- Access token needs to be generated to access the API resources.



To Invoke the API,

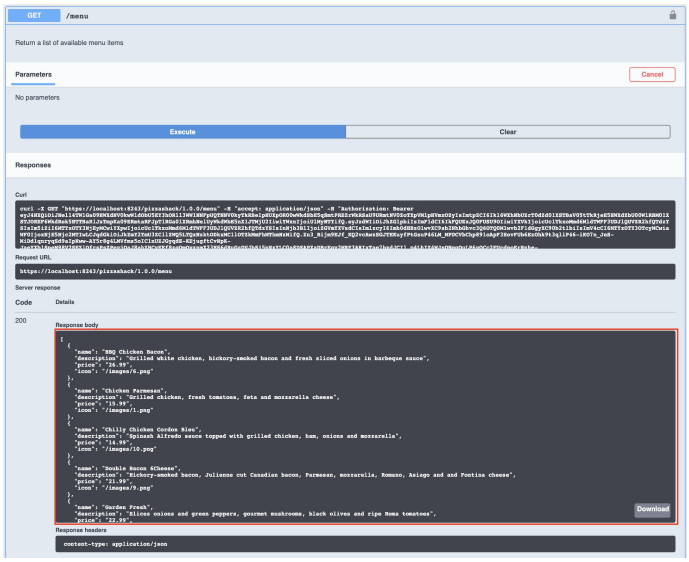
- Click on Try Out menu item in the left menu.
- Paste the access token generated in the key generation step or click on get test key.

Invoke an API

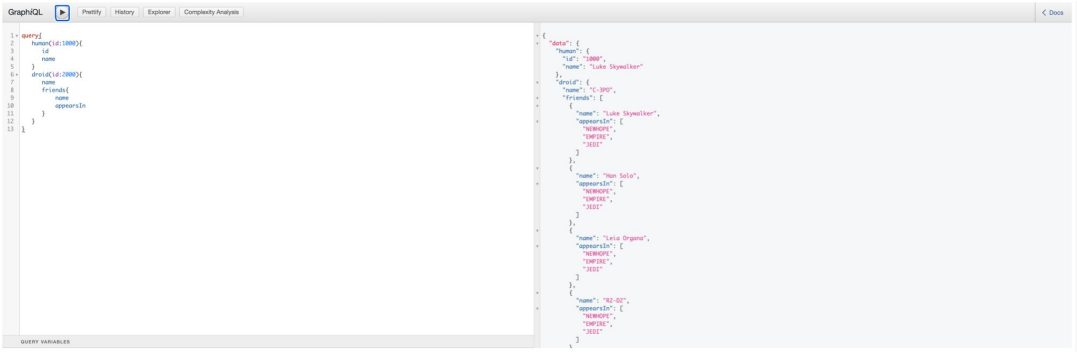


- Click on the API resource which needs to be invoked.
- Provide any parameters required (path, query or body, etc)
- Click Execute to send the request.

Invoke an API



Tryout GraphQL APIs



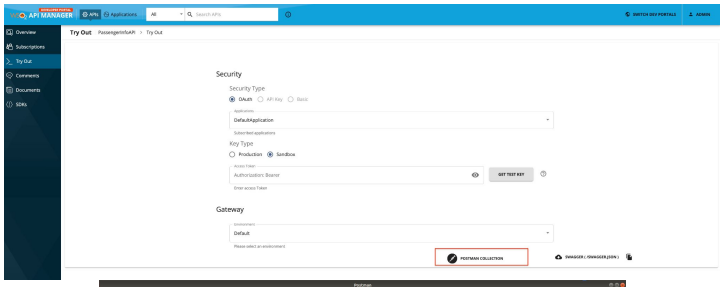
API Manager 4.0.0 Developer Portal is featured with an integrated GraphiQL try out console where you can provide the query, invoke and test the API.

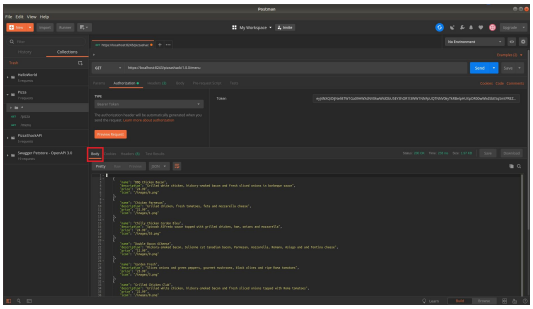
As in previous section, Create an application, subscribe and generate an access token for the GraphQL API.

Select the API and go to the Tryout page.

Using the integrated GraphiQL 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

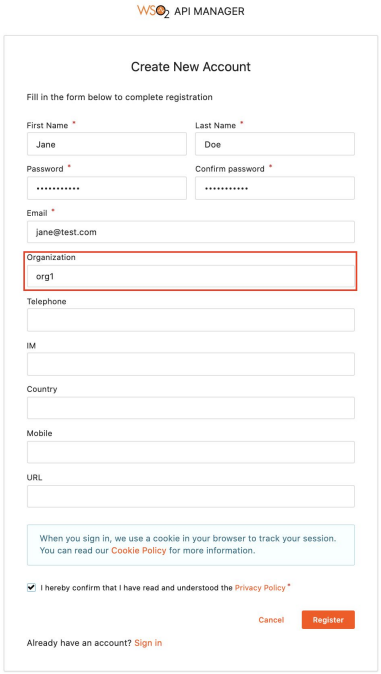
Application Sharing

Application Sharing

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

```
[apim.devportal]
enable_application_sharing = true
```

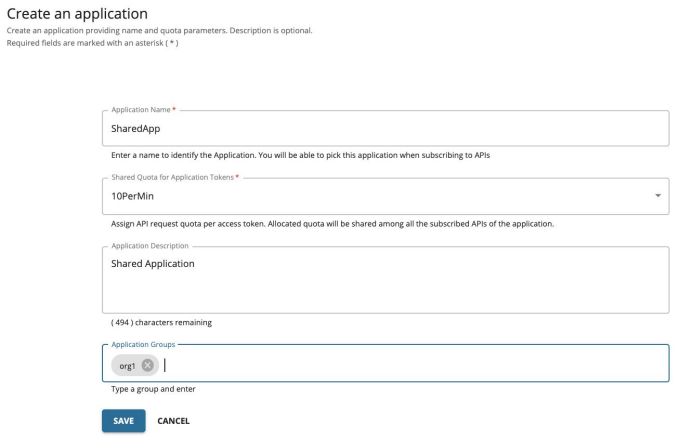
- Register users with Organizations



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

Application Sharing

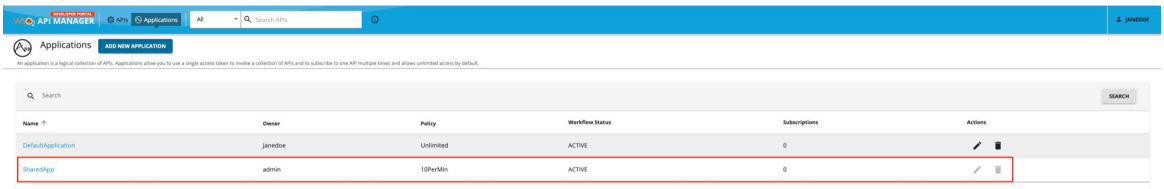
- Create an Application with Application Groups defined.



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.

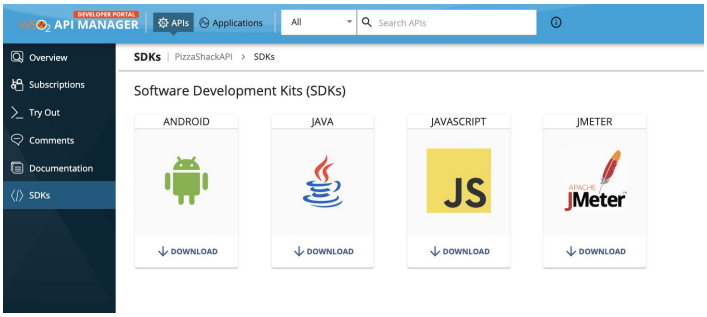


Client SDKs

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

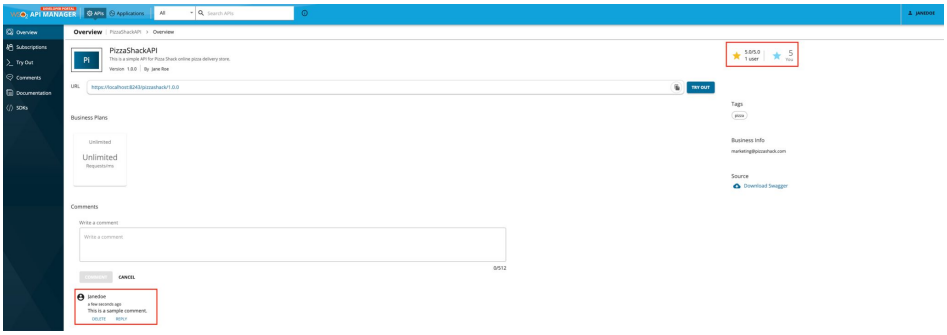
The Developer Portal provides several useful features to build and nurture an active community of users as well as a developer community, for your APIs.

- 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