



WSO2 API Manager 3.2.0

Fundamentals

Getting Started with Developer Portal

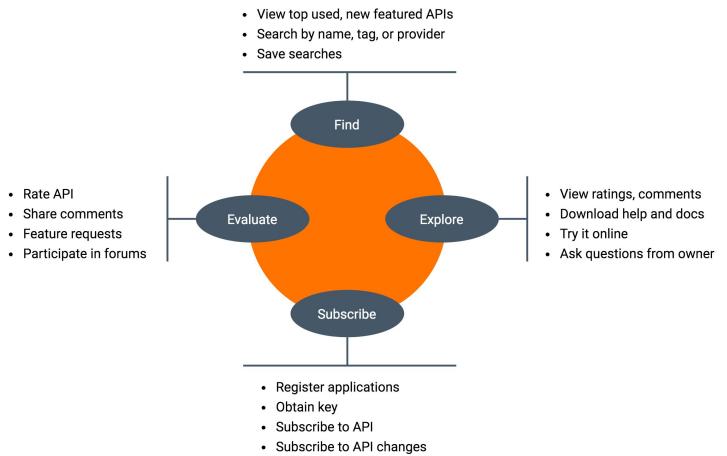


WSO2 Training



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, SubContext, Documentation, Tags, Custom Property and Microgateway Label.



WSO2 API Manager has [Apache Solr](#) based indexing for API documentation content. It provides both the API Publisher and Developer Portal a full-text search facility to search through the API documentation, and find the documents and related APIs

The API search facility

Search for an API by,

- API name
- API version
- API provider
- Context
- API status
- Description
- Subcontext
- API documentation
- Tag
- Microgateway label
- Custom property

Link - [API Search](#)



Discover APIs

The screenshot shows the WSO2 Developer Portal's 'Discover APIs' section. On the left, there's a sidebar with 'API Categories' (selected 'Pizza'), a 'Tag Cloud' (with 'pizza' highlighted), and a search bar. The main area displays two API cards:

- Starwars**: By admin, Version 1.0.0, Context /starward. It has a 5-star rating and a 'GRAPHQL' button.
- PizzaShackAPI**: By admin, Version 2.0.0, Context /pizzashack. It has a 5-star rating and a 'PROTOTYPED' button.

A dropdown menu titled 'Search Options' lists 10 search criteria, such as 'By API Name [Default]', 'By API Provider [Syntax - provider:xxxx]', etc.

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

APIs and Applications



API Listing

The screenshot shows the VOD Developer Portal's API listing interface. The top navigation bar includes links for 'APIS' and 'Applications', a search bar, and user settings. On the left, there's a sidebar with 'API Categories' (selected), 'Tag Cloud' (with a 'pizza' tag highlighted), and a 'Tag Cloud' section for 'pizza'. The main content area displays two API entries:

- Starwars**: By: admin, Version: 1.0.0, Context: /starward. It has a **GRAPHQL** icon and a **★★★★★** rating.
- PizzaShackAPI**: By: admin, Version: 2.0.0, Context: /pizzashack. It has a **PROTOTYPED** status indicator and a **★★★★★** rating.

A 'Search Options' dropdown menu is open, listing 10 search criteria from 'By API Name [Default]' to 'By API Properties [Syntax - property_name.property_value]'. The interface uses a light blue and white color scheme with some dark grey accents.



API Overview

The screenshot shows the API Overview page for the 'PizzaShackAPI' application. The top navigation bar includes links for DEVELOPER, APIs, Applications, AI, and a search bar. On the right, there are links to GO TO PUBLIC DEV PORTAL and ADMIN.

The main content area displays the following details:

- Overview:** Describes the API as a simple API for Pizza Shack online pizza delivery store.
- Subscriptions:** Version 1.0.0, Context (pizzaShack)*1.0.0, Provider Jane Rose (marketing@pizzashack.com), Technical Owner John Doe (architecture@pizzashack.com), Key Managers All Applicable, Rating 4 stars.
- Gateway Environments:** Production and Sandbox.
- Tags:** pizza.
- Resources:** Order (1), Item (1), OrderItem (OrderItem) (3).
- TEST >>** A button to test the API.
- SDK Generation:** Icons for Android, Java, and JS.

On the left sidebar, there are links for Overview, Subscriptions (0), Try Out, Comments (0), Documentation, and SDKs (0). The bottom of the sidebar indicates 'No Documents Available'.

Clicking on an API Thumbnail will go 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 a physical 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

The screenshot shows the WSO2 Developer Portal's Applications section. At the top, there is a navigation bar with links for 'APIs' and 'Applications'. A search bar is present, along with a 'GO TO PUBLIC DEV PORTAL' button and an 'ADMIN' link. Below the navigation, a heading 'Applications' is followed by a sub-heading: '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.' A large search input field labeled 'Search application by name' is centered above a table. The table has columns: Name (sorted), Owner, Policy, Workflow Status, Subscriptions, and Actions. One row is visible, showing 'DefaultApplication' as the name, 'admin' as the owner, 'Unlimited' as the policy, 'ACTIVE' as the workflow status, and '0' as the number of subscriptions. The 'Actions' column contains edit and delete icons. On the far left, there is a small circular icon with a play button symbol. On the far right, there is a yellow circle containing the number '10'.

Name ↑	Owner	Policy	Workflow Status	Subscriptions	Actions
DefaultApplication	admin	Unlimited	ACTIVE	0	

Application Creation

- Create Application Form

Create an application
Create an application providing name, quota and token type parameters. Description is optional

Application Name *

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

Per Token Quota. *

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

Application Description

(490) characters remaining

SAVE **CANCEL**



Application Overview

● Application Overview

The screenshot shows the VSO Developer Portal Applications page. The application listed is "PizzaShackApp". The details shown are:

Setting	Value
Description	Pizzashack Application
Throttling Tier	10PerMin (Allows 10 request per minute)
Token Type	Self-contained (JWT)
Workflow Status	APPROVED
Application Owner	admin

On the left sidebar, there are navigation links: Overview, Production Keys, OAuth2 Tokens, Api Key, Sandbox Keys, OAuth2 Tokens, Api Key, and Subscriptions. On the right, there are "EDIT" and "DELETE" buttons.

Application Listing

● Application Listing

 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.

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

13

Generate Keys

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

The screenshot shows the Veeva Developer Portal interface. At the top, there's a navigation bar with links for 'APIS', 'APPLICATIONS', a search bar, and user account information ('GO TO PUBLIC DEV PORTAL' and 'ADMIN'). Below the navigation is a header for the application 'PizzaShackApp' with a status of '0 Subscriptions'. On the left, a sidebar menu lists options: Overview, Production Keys (selected), OAuth2 Tokens, Api Key, and Subscriptions. The main content area is titled 'Production OAuth2 Keys' and contains two sections: 'Key and Secret' and 'Key Configuration'. In the 'Key and Secret' section, it says 'Production Key and Secret is not generated for this application'. In the 'Key Configuration' section, there are fields for 'Token Endpoint' (set to 'https://localhost:8243/token') and 'Revoke Endpoint' (set to 'https://localhost:8243/revoke'). Below these are 'Grant Types' checkboxes: Refresh Token, SAML2, Password, Client Credentials, IWA-NTLM, Device Code, Code, and JWT. A note below the checkboxes states: 'The application can use the following grant types to generate Access Tokens. Based on the application requirement, you can enable or disable grant types for this application.' At the bottom of the configuration section is a 'Callback URL' input field with the placeholder 'Callback URL is a redirection URI 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.' A red box highlights the 'GENERATE KEYS' button at the bottom-left of the configuration section. A small yellow badge with the number '14' is visible in the top right corner of the main content area.

Generate Keys

The screenshot shows the ADO Developer portal interface. At the top, there's a navigation bar with 'ADO DEVELOPER' logo, 'APIs', 'Applications', and a search bar. Below the navigation is a main content area for the application 'PizzaShackApp'. On the left, there's a sidebar with 'Overview', 'Production Keys', 'OAuth2 Tokens', 'Sandbox Keys', 'OAuth2 Tokens', 'Api Key', and 'Subscriptions'. Under 'Production Keys', it shows 'Key and Secret' with a consumer key: 'R0jgDN5kWvQ5e7TG83Zka'. Below this is a modal window titled 'Generate Access Token' with the instruction 'Please Copy the Access Token'. It contains a warning: '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 URL or in the page's refresh.)'. There are two buttons: 'GENERATE ACCESS TOKEN' and 'CURL TO GENERATE ACCESS TOKEN'. The main content area also displays 'Key Configurations' with fields for 'Token Endpoint' (https://localhost:8243/token), 'Revoke Endpoint' (https://localhost:8243/revoke), and 'Grant Types' (Refresh Token, SAML2, Password, Client Credentials, NTLM, Device Code, Code, JWT). A 'Callback URL' field is present with the placeholder 'Callback URL: https://localhost:8243/callback' and a note: 'Callback URL is a reference URL in the client application which is used by the authorization server to send the client user-agent (usually web browser) back after granting access.' At the bottom right of the main content area, there are 'EDIT' and 'DELETE' buttons.

Consumer key and consumer secret - represents the credentials of the registered application

Supported Grant Types

- Supported Grant types:
 - Password Grant
 - Client Credentials Grant
 - Authorization Code Grant
 - Implicit 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).

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.

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

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

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



```
{ "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.

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

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

grant_type=client_credentials &client_id=xxxxxxxxxxxx &client_secret=xxxxxxxxxxxx → { "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.

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

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

grant_type=authorization_code
&code=xxxxxxxxxxxxx
&redirect_uri=https://example-ap
p.com/redirect
&client_id=xxxxxxxxxxxx
&client_secret=xxxxxxxxxxxx

{
  "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

Implicit (legacy): A simplified flow that can be used by public clients, where the access token is returned immediately without an extra authorization code exchange step.

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

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

response_type=token/code
&client_id=xxxxxxxxxxxx
&redirect_uri=https://example-ap
p.com/redirect
&scope=photo
&state=XJdTynWz11IRCCrF
→ {
    "access_token": "MTQ0NjJkZmQ5OTM2NDE1ZTZjNGZmZjI3",
    "token_type": "bearer",
    "expires_in": 3600,
    "refresh_token": "IwOGYzYTlmM2YxOTQ5MGE3YmNmMDFkNTVk",
    "scope": "create"
}
```

Demo: <https://www.oauth.com/playground/implicit.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.

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

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

grant_type=refresh_token
&refresh_token=xxxxxxxxxxxx
&client_id=xxxxxxxxxx
&client_secret=xxxxxxxxxx

{
    "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.

Ex:- Application access token for a mobile application and User access token for the end user who uses the mobile app

The screenshot shows the VSO2 Developer Portal interface. On the left, there's a sidebar with options like Overview, Production Keys, OAuth2 Tokens, and **Api Key**, which is currently selected. The main area displays the 'Production API Key' section for the 'PizzaShackApp'. It includes a 'Key Restrictions' dropdown set to 'None', a 'GENERATE KEY' button (which is highlighted with a red box), and a note about generating a self-contained JWT token. A modal window titled 'Generate API Key' is open, containing a warning icon, the generated API key text area, and a 'CLOSE' button.

An API key is the simplest form of application-based security that you can configure for an API.

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

In Default doesn't expire

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. Add subscription to an API in the Application overview.
 - a. Select an API, which a subscription needs to be added.



Subscription and Key Generation Wizard.

- In the API Overview, click on Subscriptions from left menu
- Click on the Subscription and Key Generation Wizard.
- This will guide you with creating a new application, subscribing to the API and generating tokens.

Subscription & Key Generation Wizard

The screenshot shows the first step of the wizard, titled "Create application". It includes fields for "Application Name" (set to "NewApplication"), "Per Token Quota" (set to "10PerMin"), and "Application Description" (containing "This is a new application"). A note at the bottom states "(487) characters remaining". At the bottom are "CANCEL" and "NEXT" buttons. Above the form, there are five navigation icons: "Create application", "Subscribe to new application", "Generate Keys", "Generate Access Token", and "Copy Access Token".

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

Per Token Quota *
10PerMin
Assign API request quota per access token. Allocated quota will be shared among all the subscribed APIs of the application.

Application Description
This is a new application
(487) characters remaining

CANCEL NEXT

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

Subscription & Key Generation Wizard

Subscription & Key Generation Wizard

Create application **Subscribe to new application** **Generate Keys**

New Application
Select an Application to subscribe

Throttling Policy
Unlimited
Available Policies - Unlimited

Key Type
 PRODUCTION SANDBOX

Grant Types
 Refresh Token SAML2 Implicit Password Client Credentials IWA-NTLM Device Code Code JWT

The application can use the following grant types to:

Callback URL

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
This is Resident Key Manager

Environment
Sandbox

Token Endpoint
https://localhost:9443/oauth2/token

Subscription & Key Generation Wizard

Create application **Subscribe to new application** **Generate Keys** **Generate Access Token** **Copy Access Token**

Access Token
eyJhbGciOiJSUzIiLCJzdWIiOiIxMjM0NTY3ODkwIiwidHlwZSI6IlJTMjU2In0..

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.)

TEST **RESET** **FINISH**

Above token has a validity period of 1000 seconds and the token has been generated on behalf of user.

Subscribing to 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 of an API Management interface. On the left, there's a sidebar with links: Overview, Subscriptions (which is highlighted with a red box), Try Out, Comments, Documentation, and SDKs. The main content area has a header 'Subscriptions' and a 'SUBSCRIPTION & KEY GENERATION WIZARD'. Below this, a note says: 'An application is primarily used to decouple the consumer from the APIs. It allows you to generate and use a single key for multiple APIs and subscribe multiple times to a single API with different SLA levels.' The 'Subscribe' section contains a form with fields: 'Application' dropdown set to 'PizzaShackApp' (also highlighted with a red box), 'Throttling Policy' dropdown set to 'Unlimited' (also highlighted with a red box), and a 'SUBSCRIBE' button at the bottom. A small note 'Select an Application to subscribe' is visible between the dropdowns.



Add Subscription to an API in the Application

The screenshot shows the WSO2 Developer Portal interface. On the left, there's a sidebar with various application management options: Overview, Production Keys, OAuth2 Tokens, Sandbox Keys, OAuth2 Tokens, and Api Key. The 'Subscriptions' option is highlighted with a red box. The main area shows an application named 'PizzaShackApp' with a status of '0 Subscriptions'. A 'Subscription Management' section contains a 'SUBSCRIBE APIs' button, also highlighted with a red box. Below this, a message says 'No Subscriptions Available' with a note that no subscriptions are available for this application. A modal window titled 'Subscribe APIs' is open, listing two APIs: 'customer-leasing' and 'PizzaShackAPI'. For 'customer-leasing', the 'Policy' dropdown is set to 'Bronze' and the 'SUBSCRIBE' button is highlighted with a red box. For 'PizzaShackAPI', the 'Policy' dropdown is set to 'Unlimited' and the 'SUBSCRIBE' button is also highlighted with a red box. The modal includes a search bar, a 'Displaying all APIs' message, and pagination controls.

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.

The screenshot shows the Oracle Developer Cloud interface for the 'PizzaShackAPI'. The left sidebar has items like Overview, Subscriptions, Try Out (which is selected), Comments, Documentation, and SDKs. The main area is titled 'Try Out' for the 'PizzaShackAPI' application, which is listed under 'Subscriptions' and has a status of 'Not Ready'. The 'Security' section shows 'OAuth' selected as the security type. Below it, there's a 'Gateway' section with 'Production and Sandbox' selected. The 'Schemes' dropdown is set to 'HTTPS'. At the bottom, there's a table of API endpoints under the 'default' environment:

Method	Path	Status
POST	/order	Locked
GET	/menu	Locked
GET	/order/{orderId}	Locked
PUT	/order/{orderId}	Locked

To Invoke the API,

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

Invoke an API

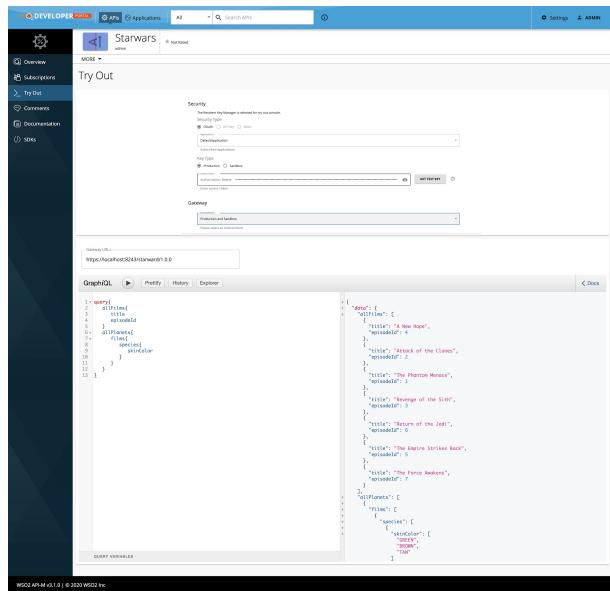
The screenshot shows a REST API invoke interface. At the top, there's a dropdown for 'Schemes' set to 'HTTPS' and a 'Authorize' button. Below that, under the 'default' configuration, there are two endpoints listed: 'POST /order' and 'GET /menu'. The 'GET /menu' endpoint is selected. A tooltip for this endpoint says 'Return a list of available menu items'. Under 'Parameters', it says 'No parameters'. There's a large blue 'Execute' button at the bottom of the endpoint details, which is highlighted with a red box. Below the endpoint details, there's a 'Responses' section with three entries: 200 (OK), 304 (Not Modified), and 406 (Not Acceptable). The 200 response example value is shown as a JSON object:

```
[{"name": "string", "price": "string", "description": "string", "category": "string", "image": "string"}]
```

- 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 3.2.0 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.

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

Create New Account

Fill in the form below to complete registration

First Name *	Last Name *
User1	User
Password *	Confirm password *
.....
Email *	
user1@gmail.com	
Organization	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](#)*

[Sign In](#) [Register](#)



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, quota and token type parameters. Description is optional

Application Name *
SharedApp

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

Per Token Quota: *
10PerMin

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

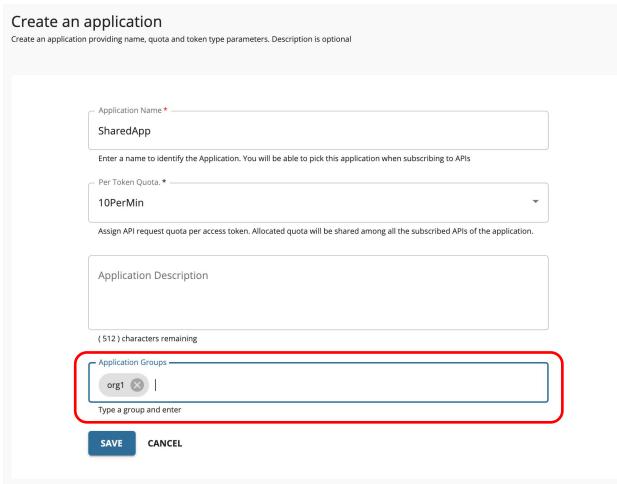
Application Description

(512) 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 VSO Developer Portal interface. At the top, there's a navigation bar with 'VSO DEVELOPER PORTAL' on the left, followed by 'APIs' and 'Applications' (which is the active tab). There are dropdowns for 'All' and 'Search APIs', and icons for 'Settings' and 'USER1'. Below the navigation is a section titled 'Applications' with a sub-section 'ADD NEW APPLICATION'. A descriptive text explains that an application is a logical collection of APIs. The main area displays a table of applications:

Name ↑	Owner	Policy	Workflow Status	Subscriptions	Actions
DefaultApplication	user1	Unlimited	ACTIVE	0	
SharedAPP	admin	10PerMin	ACTIVE	0	

Client SDKs

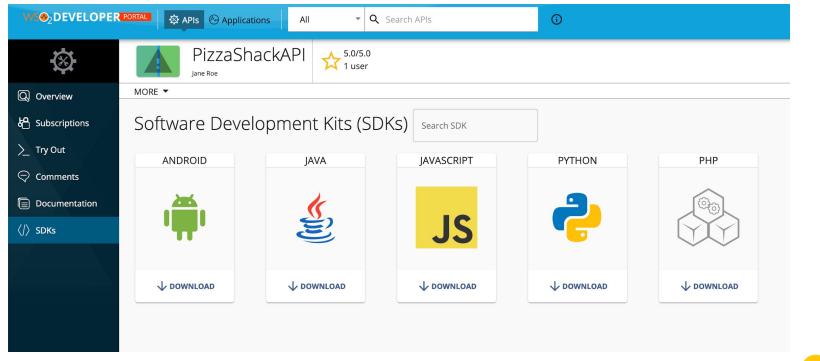


Client SDKs

- 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",
"javascript", "python",
"php"]
```

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

The image displays two side-by-side screenshots of the VSO Developer Portal interface, specifically for the PizzaShackAPI.

Left Screenshot (Overview):

- API Details:** PizzaShackAPI, Version 1.0.0, Context /pizzashack/v1.0.0, Provider Jane Doe (marketing@pizzashack.com), Technical Owner John Doe (architecture@pizzashack.com), Key Managers All Applicable.
- Rating:** 5 stars (yellow).
- Environment:** Production and Sandbox.
- Tags:** pizza.

Right Screenshot (Comments):

- Comments Section:** Shows a sample comment from admin: "This is a sample comment".
- Buttons:** ADD COMMENT and CANCEL.

Link - [Rate and Comment](#)

Let's Try it Out

Subscribing to APIs
Invoking the API

