

WSO2 API Manager

Documentation

Version 2.1.0

Table of Contents

1. Introduction	3
1.1 Overview	3
1.2 About this Release	4
2. Quick Start Guide	5

Introduction

The topics in this section introduce WSO2 API Manager Server, including the business cases it solves, its features, and architecture.

- [Overview](#)
- [About this Release](#)

Overview

As an organization implements SOA, it can benefit by exposing core processes, data and services as APIs to the public. External parties can mash up these APIs in innovative ways to build new solutions. A business can increase its growth potential and partnership advancements by facilitating developments that are powered by its APIs in a simple, decentralized manner.

However, leveraging APIs in a collaborative way introduces new challenges in exercising control, establishing trust, security and regulation. As a result, proper API management is crucial.

WSO2 API Manager overcomes these challenges with a set of features for API creation, publication, lifecycle management, versioning, monetization, governance, security etc. using proven WSO2 products such as [WSO2 Enterprise Service Bus](#), [WSO2 Identity Server](#), and [WSO2 Governance Registry](#). In addition, it is also powered by the [WSO2 Data Analytics Server](#) and is immediately ready for massively scalable deployments.

WSO2 API Manager is fully open source and is released under [Apache Software License Version 2.0](#), one of the most business-friendly licenses available today. It provides Web interfaces for development teams to deploy and monitor APIs, and for consumers to subscribe to, discover and consume APIs through a user-friendly storefront. The API Manager also provides complete API governance and shares the same metadata repository as WSO2 Governance Registry. If your setup requires to govern more than APIs, we recommend you to use WSO2 API manager for API governance and WSO2 Governance Registry for the other artefacts. That the default communication protocol of the Key Manager is Thrift.

The WSO2 API Manager is an on-going project with continuous improvements and enhancements introduced with each new release to address new business challenges and customer expectations. WSO2 invites users, developers and enthusiasts to [get involved](#) or get the assistance of our development teams at many different levels through online forums, mailing lists and support options.

About this Release

What is new in this release

The WSO2 API Manager version **2.1.0** is the successor of version **2.0.0**. It contains the following new features and enhancements:

- Ability to [manage APIs for web sockets](#)
- Ability to [generate client side SDKs](#) for subscribed APIs in the API Store
- Ability to [invoke workflows when the API lifecycle state changes](#)

Compatible WSO2 product versions

WSO2 APIM 2.1.0 is based on WSO2 Carbon 4.4.11 and is expected to be compatible with any of the WSO2 products that are based on any Carbon 4.4.x version. If you get any compatibility issues, please [contact team WSO2](#). For information on the third-party software required with APIM 2.1.0, see [Installation Prerequisites](#). For more information on the products in each Carbon platform release, see the [Release Matrix](#).

Fixed issues

See a list of [fixed issues](#) and [improvements](#) for WSO2 API Manager 2.1.0.

Known issues

For a list of known issues, see [WSO2 API Manager 2.1.0 - Known Issues](#).

Quick Start Guide

WSO2 API Manager is a complete solution for designing and publishing APIs, creating and managing a developer community, and for securing and routing API traffic in a scalable way. It leverages proven components from the WSO2 platform to secure, integrate and manage APIs. In addition, it integrates with the [WSO2 analytics platform](#) and provides out of the box reports and alerts, giving you instant insights into the APIs behavior.

Before you begin,

1. Install [Oracle Java SE Development Kit \(JDK\)](#) version 1.7.* or 1.8.* and set the `JAVA_HOME` environment variable.
2. [Download](#) WSO2 API Manager.
3. Start the API Manager by going to `<APIM_HOME>/bin` using the command-line and executing `wso2server.bat` (for Windows) or `wso2server.sh` (for Linux.)

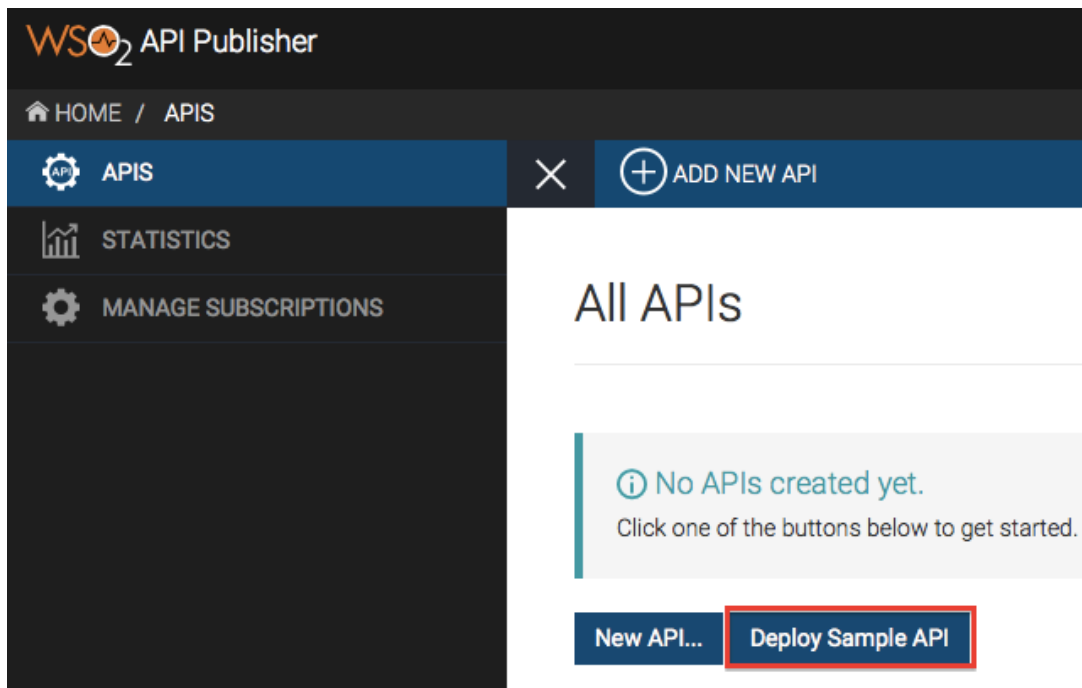
Let's go through the use cases of the API Manager:

- [Invoking your first API](#)
- [Understanding the API Manager concepts](#)
- [Deep diving into the API Manager](#)
 - [Creating users and roles](#)
 - [Creating an API from scratch](#)
 - [Adding API documentation](#)
 - [Adding interactive documentation](#)
 - [Versioning the API](#)
 - [Publishing the API](#)
 - [Subscribing to the API](#)
 - [Invoking the API](#)
 - [Monitoring APIs and viewing statistics](#)

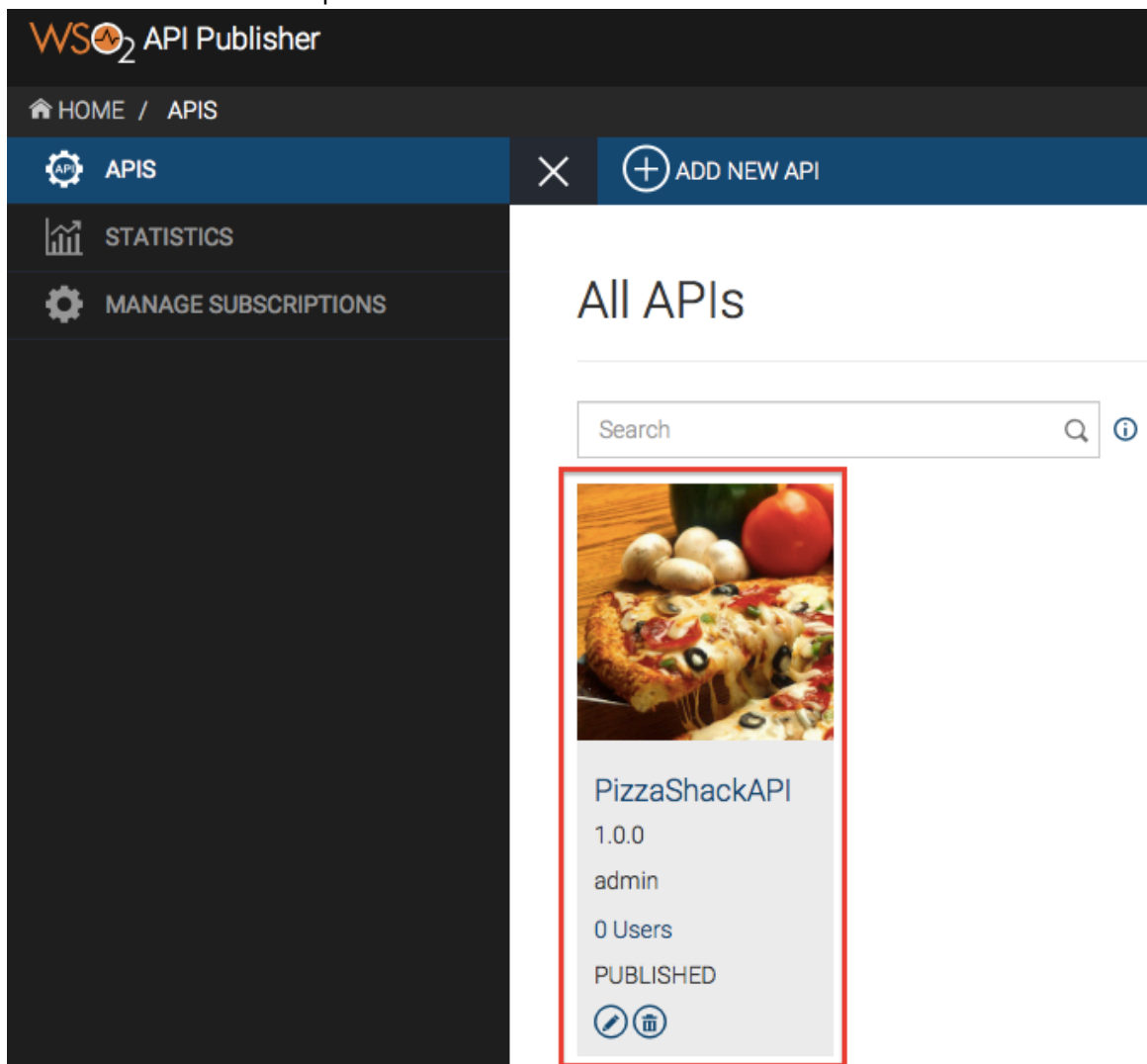
Invoking your first API

Follow the steps in this section to quickly deploy a sample API, publish it, subscribe to it, and invoke it.

1. Open the API Publisher (<https://<hostname>:9443/publisher>) and sign in with **admin/admin** credentials.
2. Click the **Deploy Sample API** button. It deploys a sample API called `PizzaShackAPI` into the API Manager.



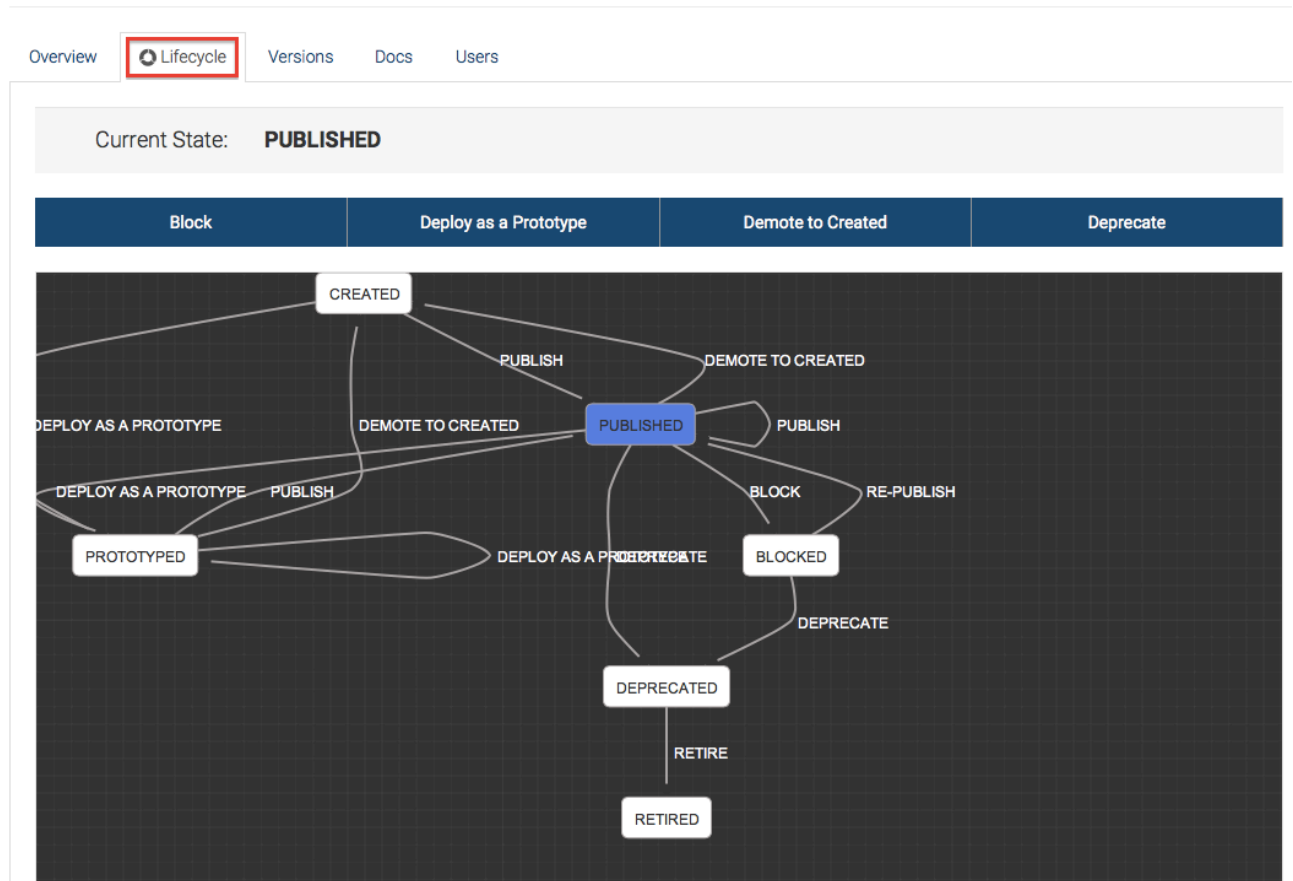
3. Click PizzaShackAPI to open it.



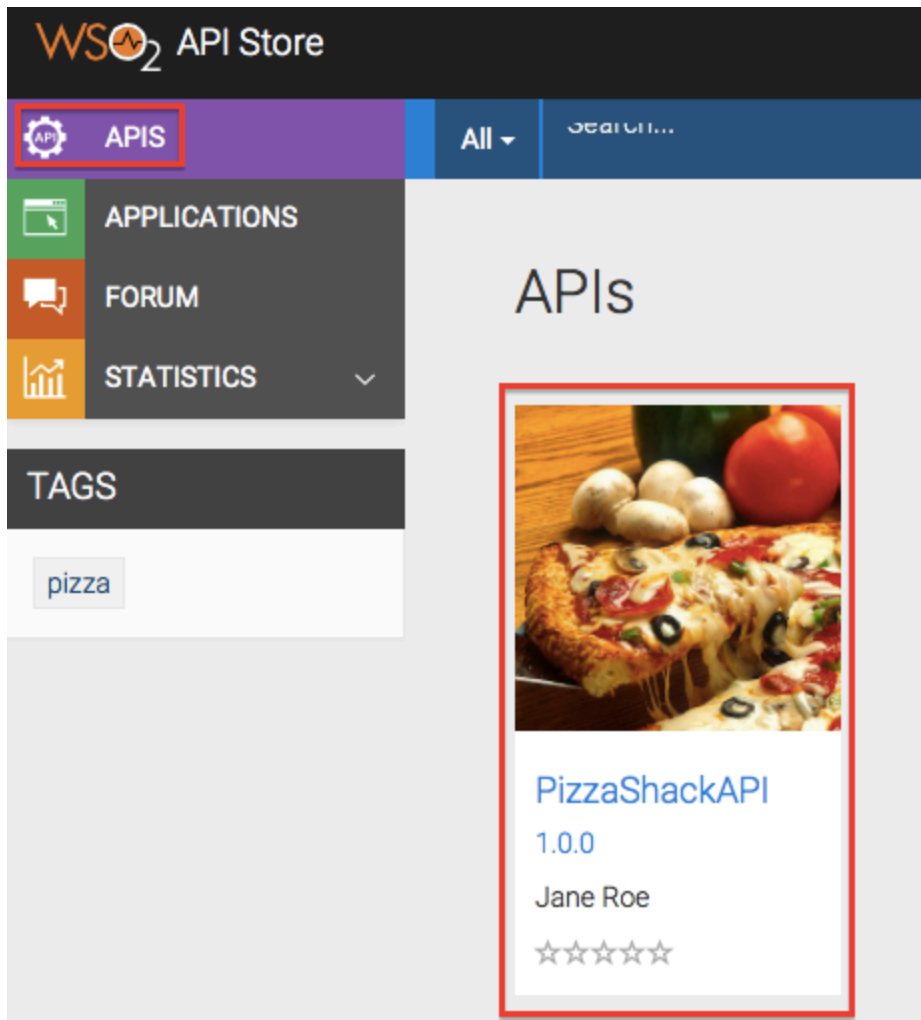
Let's publish this API.

- Go to the **Lifecycle** tab and note that the **State** is **PUBLISHED**. The API is already published to the API Store.

PizzaShackAPI - 1.0.0



- Sign in to the API Store (<https://<hostname>:9443/store>) with the **admin/admin** credentials and click on the **PizzaShackAPI** API.



6. Select the default application and an available tier, and click **Subscribe**.



7. When the subscription is successful, click **View Subscriptions** on the information message that appears. Click the **Production Keys** tab and click **Generate Keys** to generate an **access token** to invoke the API.

WSO2 API Store

APPLICATIONS APPLICATION LIST EDIT

APIS FORUM STATISTICS

DefaultApplication

Details **Production Keys** Sandbox Keys Subscriptions

No Keys Found
No keys are generated for this type in this application.

Grant Types
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.

☒ SAML2 ☒ IWA-NTLM ☐ Implicit ☒ Refresh Token
☒ Client Credential ☐ Code ☒ Password

Callback URL


Access token validity period
 Seconds.

Generate keys

You have now successfully subscribed to an API. Let's invoke the API using the integrated Swagger-based API Console.

- Click the **APIs** menu again and click the `PizzaShackAPI` to open it. When the API opens, click its **API Console** tab.

PizzaShackAPI - 1.0.0



Version: 1.0.0
By: Jane Roe
Updated: 21/Jul/2016 13:41:55 PM IST
Status: **PUBLISHED**
Rating: ☆☆☆☆☆

Applications
 Select Application...

Tiers
 Unlimited

Subscribe

Overview **API Console** Documentation Forum

Try

Using **Key**

Set Request Header

Expand the GET method (which retrieves the menu) and click **Try it out**.

GET /menu

Implementation Notes
Return a list of available menu items

Response Class (Status 200)
OK. List of APIs is returned.

Model | Example Value

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

Response Content Type application/json

Headers

Header	Description	Type	Other
--------	-------------	------	-------

Response Messages

HTTP Status Code	Reason	Response Model	Headers
304	Not Modified. Empty body because the client has already the latest version of the requested resource.		
406	Not Acceptable. The requested media type is not supported	Model Example Value	

```
{
  "message": "string",
  "error": {
    {
      "message": "string",
      "code": 0
    }
  },
  "description": "string",
  "code": 0,
  "moreInfo": "string"
}
```

[Try it out!](#)

Note the response for the API invocation. It returns the list of menu items.

Response Body

```
[
  {
    "name": "BBQ Chicken Bacon",
    "icon": "/images/6.png",
    "description": "Grilled white chicken, hickory-smoked bacon and fresh sliced onions in barbeque sauce",
    "price": "20.99"
  },
  {
    "name": "Chicken Parmesan",
    "icon": "/images/1.png",
    "description": "Grilled chicken, fresh tomatoes, feta and mozzarella cheese",
    "price": "20.99"
  },
  {
    "name": "Chilly Chicken Cordon Bleu",
    "icon": "/images/10.png",
    "description": "Spinash Alfredo sauce topped with grilled chicken, ham, onions and mozzarella",
    "price": "26.99"
  },
  {
    "name": "Double Bacon 6Cheese",

```

You have deployed a sample API, published it to the API Store, subscribed to it, and invoked the API using our integrated API Console.

Understanding the API Manager concepts

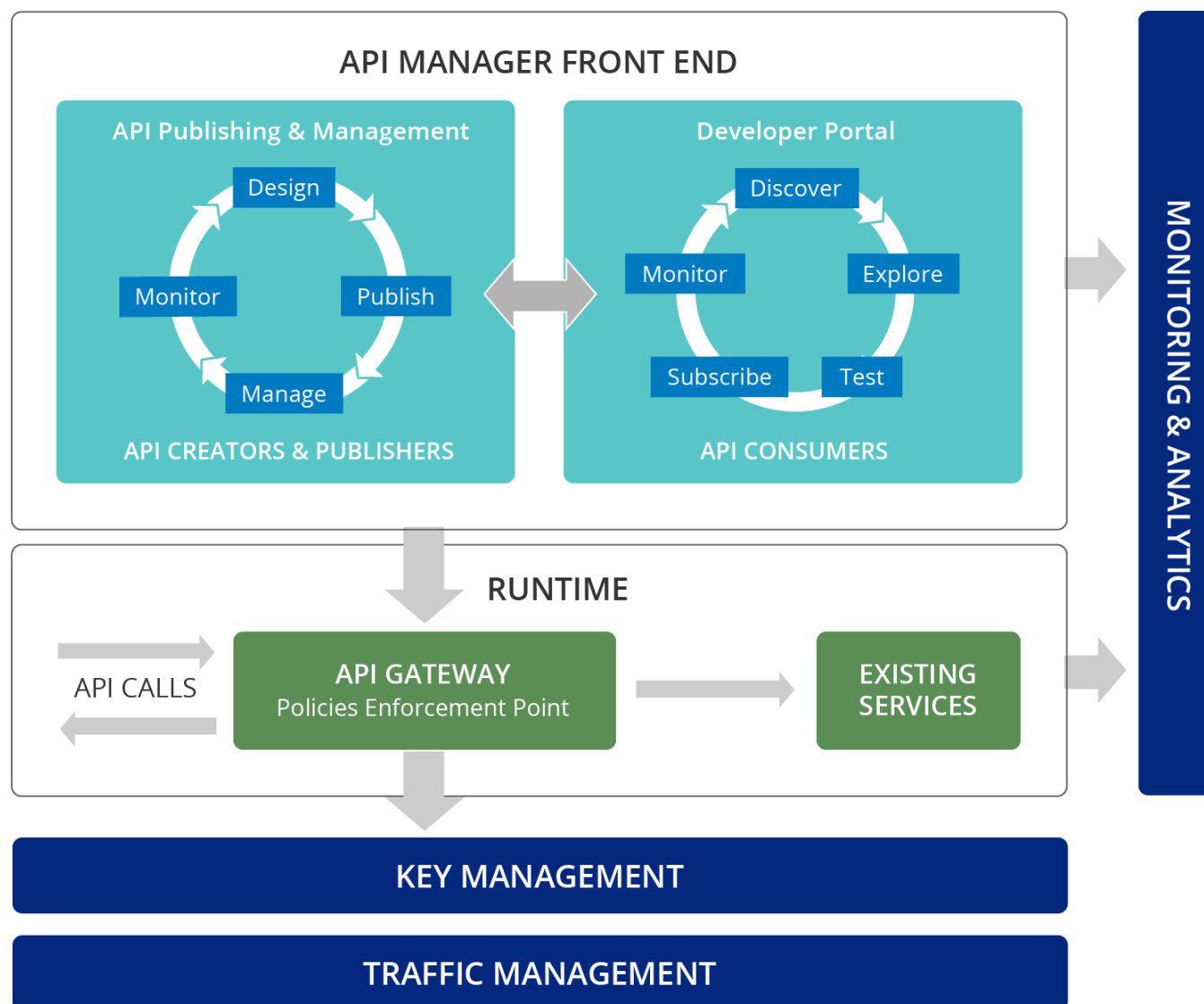
Before we look into the API management activities in detail, let's take a look at the basic API management concepts.

[[Components](#)] [[Users and roles](#)] [[API lifecycle](#)] [[Applications](#)] [[Throttling tiers](#)] [[API keys](#)] [[API resources](#)]

Components

The API Manager comprises of the following components:

- **API Publisher:** Enables API providers to publish APIs, share documentation, provision API keys and gather feedback on features, quality and usage. You access the Web interface via `https://<Server Host>:9443/publisher`.
- **API Store (Developer Portal):** Enables API consumers to self register, discover and subscribe to APIs, evaluate them and interact with API Publishers. You access the Web interface via `https://<Server Host>:9443/store`.
- **API Gateway:** Secures, protects, manages, and scales API calls. It is a simple API proxy that intercepts API requests and applies policies such as throttling and security checks. It is also instrumental in gathering API usage statistics. The Web interface can be accessed via `https://<Server Host>:9443/carbon`.
- **Key Manager:** Handles all security and key-related operations. The API Gateway connects with the Key Manager to check the validity of subscriptions, OAuth tokens, and API invocations. The Key Manager also provides a token API to generate OAuth tokens that can be accessed via the Gateway.
- **Traffic Manager:** Helps users to regulate API traffic, make APIs and applications available to consumers at different service levels and secures APIs against security attacks. The Traffic Manager features a dynamic throttling engine to process throttling policies in real-time.
- **WSO2 API Manager Analytics:** Provides a host of statistical graphs, an alerting mechanism on pre-determined events and a log analyzer.



Users and roles

The API manager offers three distinct community roles that are applicable to most enterprises:

- **Creator:** A creator is a person in a technical role who understands the technical aspects of the API (interfaces, documentation, versions, how it is exposed by the Gateway, etc.) and uses the API publisher to provision APIs into the API Store. The creator uses the API Store to consult ratings and feedback provided by API users. Creators can add APIs to the store but cannot manage their life cycle (e.g., make them visible to the outside world.)
- **Publisher:** A publisher manages a set of APIs across the enterprise or business unit and controls the API life cycle and monetization aspects.
- **Consumer:** A consumer uses the API Store to discover APIs, see the documentation and forums, and rate/comment on the APIs. Consumers subscribe to APIs to obtain API keys.

API lifecycle

An API is the published interface, while the service is the implementation running in the backend. APIs have their own lifecycles that are independent of the backend services they rely on. This lifecycle is exposed in the API Publisher and is managed by the publisher role.

The following stages are available in the default API life cycle:

- **CREATED:** API metadata is added to the API Store, but it is not visible to subscribers yet, nor deployed to the API Gateway.
- **PROTOTYPED:** The API is deployed and published in the API Store as a prototype. A prototyped API is usually a mock implementation made public in order to get feedback about its usability. Users can try out a prototyped API without subscribing to it.
- **PUBLISHED:** The API is visible in the API Store and available for subscription.
- **DEPRECATED:** The API is still deployed in the API Gateway (i.e., available at runtime to existing users) but not visible to subscribers. You can deprecate an API automatically when a new version of it is published.
- **RETIRED:** The API is unpublished from the API Gateway and deleted from the Store.
- **BLOCKED:** Access to the API is temporarily blocked. Runtime calls are blocked, and the API is not shown in the API Store anymore.

Applications

An application is primarily used to decouple the consumer from the APIs. It allows you to do the following:

- Generate and use a single key for multiple APIs.
- Subscribe multiple times to a single API with different SLA levels.

You create an application to subscribe to an API. The API Manager comes with a default application, and you can also create as many applications as you like.

Throttling tiers

Throttling tiers are associated with an API at subscription time and can be defined at an API-level, resource-level, subscription-level and application-level (per token). They define the throttling limits enforced by the API Gateway, e.g., 10 TPS (transactions per second). The final throttle limit granted to a given user on a given API is ultimately defined by the consolidated output of all throttling tiers together. The API Manager comes with three predefined tiers for each level and a special tier called *Unlimited*, which you can disable by editing the `<TierManagement>` element of the `<APIM_HOME>/repository/conf/api-manager.xml` file.

API keys

The API Manager supports two scenarios for authentication:

- An access token is used to identify and authenticate a whole application.

- An access token is used to identify the final user of an application (for example, the final user of a mobile application deployed on many different devices).

Application access token: Application access tokens are generated by the API consumer and must be passed in the incoming API requests. The API Manager uses the OAuth2 standard to provide key management. An API key is a simple string that you pass with an HTTP header (e.g., "Authorization: Bearer NtBQkXoKElu0H1a1fQ0DWfo6IX4a,") and it works equally well for SOAP and REST calls.

Application access tokens are generated at the application level and valid for all APIs that you associate to the application. These tokens have a fixed expiration time, which is set to 60 minutes by default. You can change this to a longer time, even for several weeks. Consumers can regenerate the access token directly from the API Store. To change the default expiration time, you open the `<APIM_HOME>/repository/conf/identity.xml` file and change the value of the element `<ApplicationAccessTokenDefaultValidityPeriod>`. If you set a negative value, the token never expires. **Changes to this value are applied only to the new applications that you create.**

Application user access token: You generate access tokens on demand using the Token API. In case a token expires, you use the Token API to refresh it.

Application user access tokens have a fixed expiration time, which is 60 minutes by default. You can update it to a longer time by editing the `<AccessTokenDefaultValidityPeriod>` element in the `<APIM_HOME>/repository/conf/identity/identity.xml` file.

The Token API takes the following parameters to generate the access token:

- Grant Type
- Username
- Password
- Scope

To generate a new access token, you issue a Token API call with the above parameters where **grant_type=password**. The Token API then returns two tokens: an access token and a refresh token. The access token is saved in a session on the client side (the application itself does not need to manage users and passwords). On the API Gateway side, the access token is validated for each API call. When the token expires, you refresh the token by issuing a token API call with the above parameters where **grant_type=refresh_token** and passing the refresh token as a parameter.

API resources

An API is made up of one or more resources. Each resource handles a particular type of request and is analogous to a method (function) in a larger API. API resources accept the following optional attributes:

- **verbs:** Specifies the HTTP verbs a particular resource accepts. Allowed values are GET, POST, PUT, PATCH, OPTIONS, DELETE. You can give multiple values at once.
- **uri-template:** A URI template as defined in <http://tools.ietf.org/html/rfc6570>. E.g., `/phoneverify/<phoneNumber>`.
- **url-mapping:** A URL mapping defined as per the servlet specification (extension mappings, path mappings, and exact mappings).
- **Throttling tiers:** Limits the number of hits to a resource during a given period of time.
- **Auth-Type:** Specifies the Resource level authentication along the HTTP verbs. Auth-type can be None, Application, or Application User.
 - None: Can access the particular API resource without any access tokens.
 - Application: An application access token is required to access the API resource.
 - Application User: A user access token is required to access the API resource.

Deep diving into the API Manager

Let's take a look at the typical API management activities in detail:

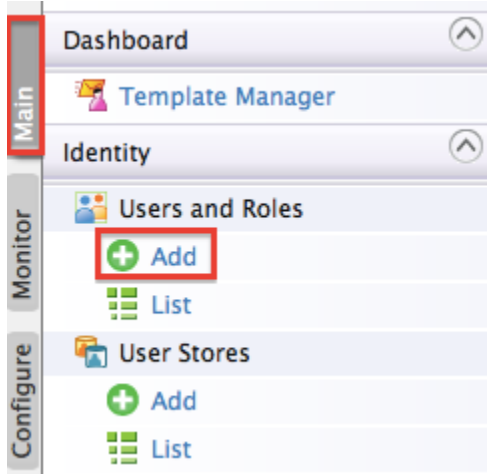
- [Creating users and roles](#)
- [Creating an API from scratch](#)
- [Adding API documentation](#)

- Adding interactive documentation
- Versioning the API
- Publishing the API
- Subscribing to the API
- Invoking the API
- Monitoring APIs and viewing statistics

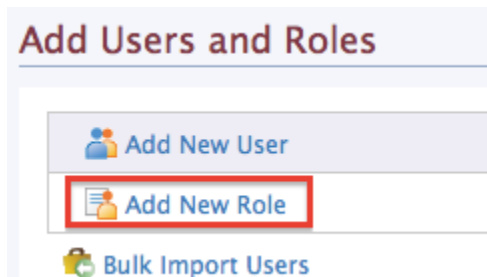
Creating users and roles

In [users and roles](#), we introduced a set of users who are commonly found in many enterprises. Let's see how you can sign in to the Management Console as an admin and create these roles.

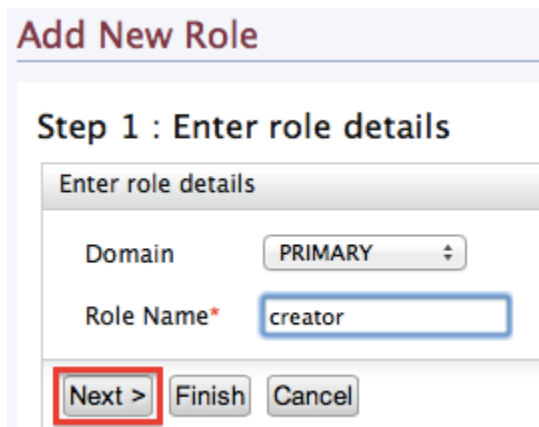
1. Sign in to the Management Console (<https://<hostname>:9443/carbon>) of the API Manager using **admin/admin** credentials.
2. Click **Add** in the **Users and Roles** section under the **Main** menu.



3. Click **Add New Role**.

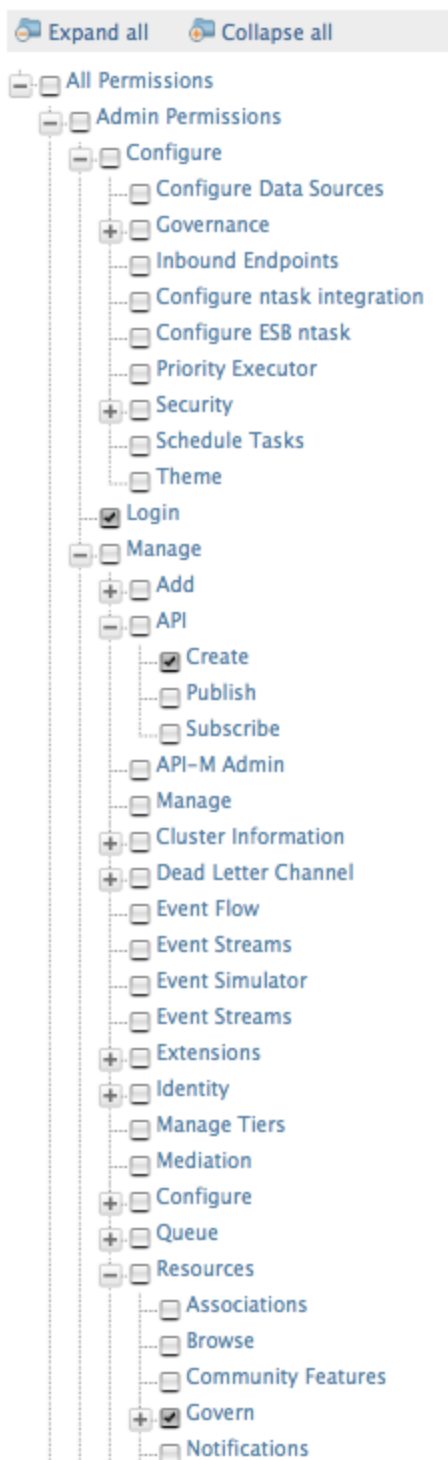


4. Give the role name as `creator` and click **Next**.



5. A list of permissions opens. Select the following and click **Finish**.
 - All Permissions > Admin Permissions > Configure > Governance and all underlying permissions

- All Permissions > Admin Permissions > Login
- All Permissions > Admin Permissions > Manage > API > Create
- All Permissions > Admin Permissions > Manage > Resources > Govern and all underlying permissions



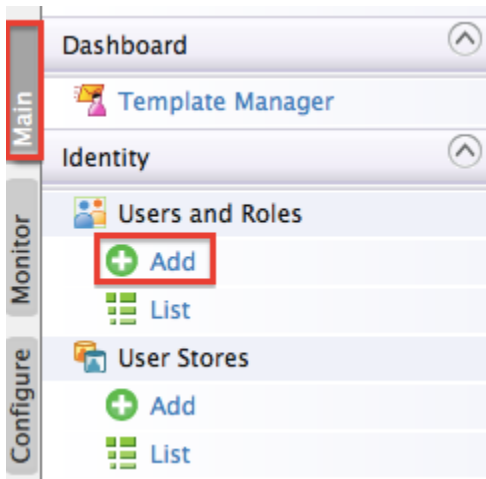
6. Similarly, create the `publisher` role with the following permissions.
 - All Permissions > Admin Permissions > Login
 - All Permissions > Admin Permissions > Manage > API > Publish
7. Note that the API Manager comes with the `subscriber` role available by default. It has the following permissions:
 - All Permissions > Admin Permissions > Login
 - All Permissions > Admin Permissions > Manage > API > Subscribe

8. The roles you added (creator and publisher) are now displayed under **Roles**.

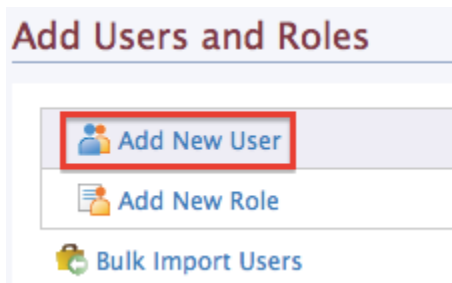
Roles	
Search Roles Select Domain ALL-USER-STORE-DOMAINS ▾ Enter role name pattern (* for all) <input type="text"/> Search Roles	
Name	Actions
admin	Assign Users View Users
Application/admin_DefaultApplication_PRODUCTION	Rename Permissions Assign Users View Users Delete
creator	Rename Permissions Assign Users View Users Delete
Internal/everyone	Permissions
Internal/subscriber	Rename Permissions Assign Users View Users Delete
publisher	Rename Permissions Assign Users View Users Delete

Let's create users for each of the roles.

9. Click **Add** in the **Users and Roles** section under the **Main** menu.



10. Click **Add New User**.



11. Give the username/password and click **Next**. For example, let's create a new user by the name `apipublisher`.

Add New User

Step 1 : Enter user name

Enter user name

Domain PRIMARY ▾

User Name*

Password*

Password Repeat*

Next > Finish Cancel

12. Select the role you want to assign to the user (e.g., publisher) and click **Finish**.

Add User

Step 2 : Select roles of the user

Enter role name pattern (* for all) Search Users

Users of Role

Select all on this page | Unselect all on this page

☐ admin

☐ creator

☒ publisher

☒ Internal/everyone

☐ Internal/subscriber

☐ Application/admin_DefaultApplication_PRODUCTION

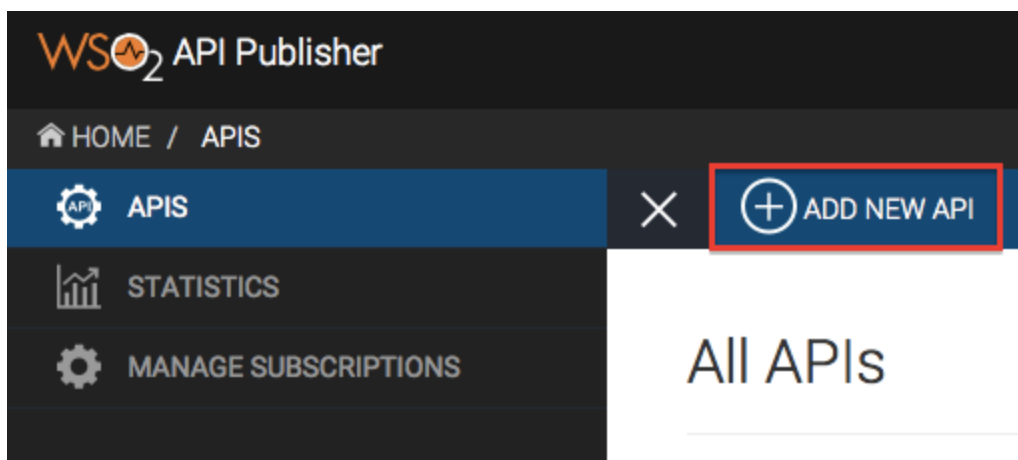
Finish Cancel

13. Similarly, create a new user by the name `apicreator` and assign the creator role.

Creating an API from scratch

Let's create an API from scratch.

1. Sign in to the API Publisher (<https://<hostname>:9443/publisher>) as **apicreator**.
2. In the **APIS** menu, click **Add New API**.



3. Select the option to design a new API and click **Start Creating**.

Let's get started!

☐ **I have an Existing API**
Use an existing API's endpoint or the API definition to create an API.

☐ **I have a SOAP Endpoint**
Use an existing SOAP endpoint to create a managed API. Import WSDL of the SOAP service.

☒ **Design New API**
Design and prototype a new API.

Start Creating

4. Give the information in the table below.

Field	Sample value
Name	PhoneVerification
Context	/phoneverify
Version	1.0.0
Visibility	Public
API Definition	<ul style="list-style-type: none"> • URL pattern: CheckPhoneNumber • Request types: GET, POST

Click **Add** and then click **Next: Implement >** to move on to the next page.

Design API

General Details

Name: * PhoneVerification

Context: * /phoneverify

Version: * 1.0.0

Visibility: * Public

Description:
Maximum 20000 characters.

Tags: * Add tags
Type a Tag and Enter

Thumbnail Image

Select image
Dimensions (max): 100 x 100 pixels

API Definition

URL Pattern: /phoneverify/1.0.0 CheckPhoneNumber

GET POST PUT DELETE PATCH HEAD more

Add

Save Next: Implement >

5. Select the **Managed API** option.

PhoneVerification: /phoneverify/1.0.0

1 Design **2 Implement** **3 Manage**

Managed API
Provide the production and sandbox endpoints of the API to be managed.

Prototyped API
Use the inbuilt JavaScript engine to prototype the API or provide an endpoint to a prototype API. The inbuilt JavaScript engine does not have suppo...

6. Give the following information and click **Next: Manage** > once you are done.

Field	Sample value
Endpoint type	HTTP/REST Endpoint
Production endpoint	In this guide, we work with a service exposed by the Cdyne services provider. We use their phone validation service, which has SOAP and REST interfaces. Endpoint is http://ws.cdyne.com/phoneverify/phoneverify.asmx . To verify the URL, click the Test button next to it. This sample service has two operations: CheckPhoneNumber and CheckPhoneNumbers. Let's use CheckPhoneNumber here.
Sandbox endpoint	Endpoint is http://ws.cdyne.com/phoneverify/phoneverify.asmx . To verify the URL, click the Test button next to it.

1 Design
2 Implement
3 Manage

Managed API
Provide the production and sandbox endpoints of the API to be managed.

Endpoint Type : * ? HTTP/REST Endpoint
☐ Load Balanced ☐ Failover

Production Endpoint : * ? http://ws.cdyne.com/phoneverify/phoneverify.asmx ⚙️ Test
✓ Valid

Sandbox Endpoint : * ? http://ws.cdyne.com/phoneverify/phoneverify.asmx ⚙️ Test
✓ Valid

Show More Options

Message Mediation Policies

Enable Message Mediation ☐ Check to select a message mediation policy to be executed in the message flow

CORS configuration

Enable API based CORS Configuration ☐

Save Next : Manage >

7. Provide the following information in the **Manage** tab. Leave default values for the rest of the parameters in the UI.

Field	Value	Description
Subscription Tiers	<Select all available tiers>	The API can be available for subscription at different levels of service. They allow you to limit the number of successful hits to an API during a given period of time.

The screenshot shows the 'Implement' tab of the WSO2 API Manager interface. It is divided into two main sections: 'Configurations' and 'Throttling Settings'.

Configurations:

- Make this the Default Version:** A checkbox that is currently unchecked. Below it, text reads: 'No default version defined for the current API'.
- Transports:** Two checkboxes are checked: 'HTTPS' and 'HTTP'.
- Response Caching:** A dropdown menu is set to 'Disabled'.

Throttling Settings:

- Maximum Backend Throughput:** Two radio buttons are present: 'Unlimited' (selected) and 'Specify'.
- Subscription Tiers:** Four checkboxes are checked, each with a description:
 - Unlimited:** Allows unlimited requests
 - Gold:** Allows 5000 requests per minute
 - Silver:** Allows 2000 requests per minute
 - Bronze:** Allows 1000 requests per minute
- Advanced Throttling Policies:** Two radio buttons are present: 'Apply to API' and 'Apply per Resource' (selected). Below them is a button labeled 'Select Policy per Resource'.

At the bottom of the 'Throttling Settings' section, there is a link: 'Refer documentation for more information about each throttling setting.'

8. Once you are done, click **Save**.

Adding API documentation

1. In the **APIS** menu, click the thumbnail of the API to open it.
2. Click on the API's **Docs** tab and click **Add New Document**.

PhoneVerification - 1.0.0

The screenshot shows the 'Docs' tab for the 'PhoneVerification - 1.0.0' API. At the top, there are four tabs: 'Overview', 'Versions', 'Docs' (which is selected and highlighted with a red box), and 'Users'. Below the tabs, there is a button labeled 'Add New Document' (also highlighted with a red box). Below the button, there is a message box with an information icon and the text: 'No documentation associated with the API'. Below this message, it says: 'There is no documentation created for this API. You can add new documentation to this API by clicking the "Add New Document" button.'

3. The document options appear. Note that you can create documentation inline, via a URL, or as a file. For inline documentation, you can edit the content directly from the API publisher interface. You get several documents types:
 - How To
 - Samples and SDK
 - Public forum / Support forum (external link only)
 - API message formats
 - Other
4. Create a 'How To' named PhoneVerification, specifying in-line content as the source and optionally

entering a summary. When you have finished, click **Add Document**.

PhoneVerification - 1.0.0

Overview
Versions
Docs
Users

Add New Document

Name*

Summary*

Type
☒ How To
☐ Samples & SDK
☐ Public Forum
☐ Support Forum
☐ Other (specify)

Source
☒ Inline
☐ URL
☐ File

Add Document
Cancel

i No documentation associated with the API
There is no documentation created for this API. You can add new documentation to this API by clicking the "Add New Document" button.

5. Once the document is added, click **Edit Content** to open an embedded editor.

PhoneVerification - 1.0.0

Overview
Versions
Docs
Users

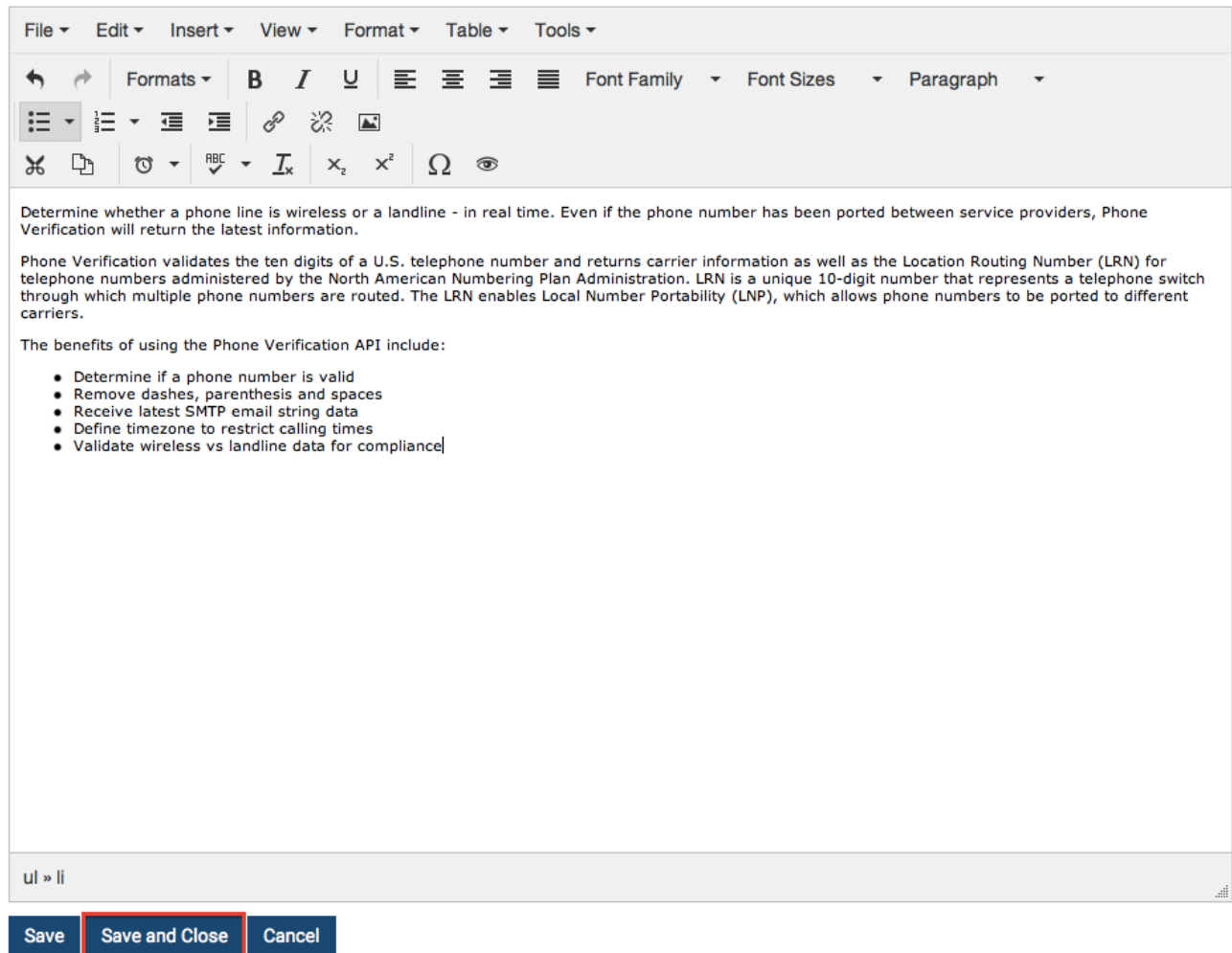
Add New Document

Name	Type	Modified On	Actions
PhoneVerification	How To	July 22, 2016 8:48:53 PM GMT+05:30	<input checked="" type="button" value="Edit Content"/> <input type="button" value="Update"/> <input type="button" value="Delete"/>

Show 10 entries Showing 1 to 1 of 1 entries

6. Enter your API's documentation and click **Save and Close**.

PhoneVerification



Adding interactive documentation

WSO2 API Manager has an integrated [Swagger UI](#) , which is part of the Swagger project.


Swagger is a 100% open source, standard, language-agnostic specification and a complete framework for describing, producing, consuming, and visualizing RESTful APIs, without the need of a proxy or third-party services . Swagger allows consumers to understand the capabilities of a remote service without accessing its source code and interact with the service with a minimal amount of implementation logic. Swagger helps describe a services in the same way that interfaces describe lower-level programming code.

The [Swagger UI](#) is a dependency-free collection of HTML, JavaScript, and CSS that dynamically generates documentation from a Swagger-compliant API. Swagger-compliant APIs give you interactive documentation and more discoverability. The Swagger UI has JSON code, and its UI facilitates easier code indentation, provides keyword highlighting, and shows syntax errors on the fly. You can add resource parameters, summaries and descriptions to your APIs using the Swagger UI.

Also, see the [Swagger 2.0 specification](#).

1. Open the API Publisher (<https://<hostname>:9443/publisher>) and sign in as **apicreator**.
2. Click the **Edit** icon for the `PhoneVerification` API. This opens the API in its edit mode.

All APIs




PizzaShackAPI

1.0.0

admin

1 User

PUBLISHED



PhoneVerifica...

1.0.0

apicreator

0 Users

CREATED

- Click the **Edit Source** button under the **API Definition** section.

API Definition

URL Pattern Url Pattern E.g.: path/to/resource

☐ GET
 ☐ POST
 ☐ PUT
 ☐ DELETE
 ☐ PATCH
 ☐ HEAD
 [more](#)

GET	/CheckPhoneNumber	+ Summary	<input type="button" value="Trash"/>
POST	/CheckPhoneNumber	+ Summary	<input type="button" value="Trash"/>

- The JSON code of the API opens in a separate page. Expand its GET method, add the following parameters and click **Apply Changes**.

```
parameters:
  - in: query
    name: PhoneNumber
    description: Give the phone number to be validated
    type: string
    required: true
  - in: query
    name: LicenseKey
    description: Give the license key as 0 for testing purpose
    type: string
    required: true
```

```

1  swagger: '2.0'
2  paths:
3    /CheckPhoneNumber:
4      get:
5        responses:
6          '200':
7            description: ''
8            x-auth-type: Application & Application User
9            x-throttling-tier: Unlimited
10       parameters:
11         - in: query
12           name: PhoneNumber
13           description: Give the phone number to be validated
14           type: string
15           required: true
16         - in: query
17           name: LicenseKey
18           description: Give the license key as 0 for testing purpose
19           type: string
20           required: true
21       post:
22         responses:
23           '200':
24             description: ''
25         parameters:
26           - name: Payload
27             description: Request Body
28             required: false
29             in: body
30             schema:
31               type: object
32               properties:
33                 payload:
34                   type: string
35             x-auth-type: Application & Application User
36             x-throttling-tier: Unlimited
37       info:
38         title: PhoneVerification
39         version: 1.0.0

```

5. Back in the API Publisher, note that the changes you did appear in the API Console's UI. You can add more parameters and edit the summary/descriptions using the API Publisher UI as well. Once done, click **Save**.

API Definition

URL Pattern: Url Pattern E.g.: path/to/resource Import Edit Source

☐ GET
 ☐ POST
 ☐ PUT
 ☐ DELETE
 ☐ PATCH
 ☐ HEAD
 [more](#)

+ Add

GET /CheckPhoneNumber + Summary ⓘ

Description + Add Implementation Notes

Produces : *Empty* **Consumes** : *Empty*

Parameters :

Parameter Name	Description	Parameter Type	Data Type	Required	Delete
PhoneNumber	Give the phone number to be validated	query	string	True	ⓘ
LicenseKey	Give the license key as 0 for testing purpose	query	string	True	ⓘ

+ Add Parameter

POST /CheckPhoneNumber + Summary ⓘ

Save Next: Implement >

Versioning the API


Let's create a new version of this API.

1. Sign in to the API Publisher as **apicreator** if you are not logged in already.
2. Click the PhoneVerification API to open it and then click **Create New Version**.



PhoneVerification - 1.0.0

Overview
Versions
Docs
Users



0 Users

CREATED

Docs

[View in Store](#)

Visibility	Public
Context	/phoneverify/1.0.0
Production URL	http://ws.cdyne.com/phoneverify/phoneverify.asmx
Sandbox URL	http://ws.cdyne.com/phoneverify/phoneverify.asmx
Date Last Updated	July 22, 2016 9:20:02 PM GMT+05:30
Tier Availability	Bronze,Unlimited,Gold,Silver
Default API Version	None
Published Environments	Production and Sandbox

3. Give a new version number (e.g., 2.0.0) and click **Done**.

PhoneVerification - 1.0.0

Create New Version

New Version: * ⓘ

Make this the Default Version ⓘ
☐

Info!

No default version defined for the current API

Done


Cancel

4. Note that the new version of the API is created in the API Publisher.


Publishing the API

1. Sign in to the API Publisher as the **apipublisher** user that you created earlier in this guide, and click the **PhoneVerification** API's version 2.0.0.


All APIs



PizzaShackAPI
 1.0.0
 admin
 1 User
 PUBLISHED



PhoneVerifica...
 1.0.0
 apicreator
 0 Users
 CREATED



PhoneVerifica...
 2.0.0
 apicreator
 0 Users
 CREATED

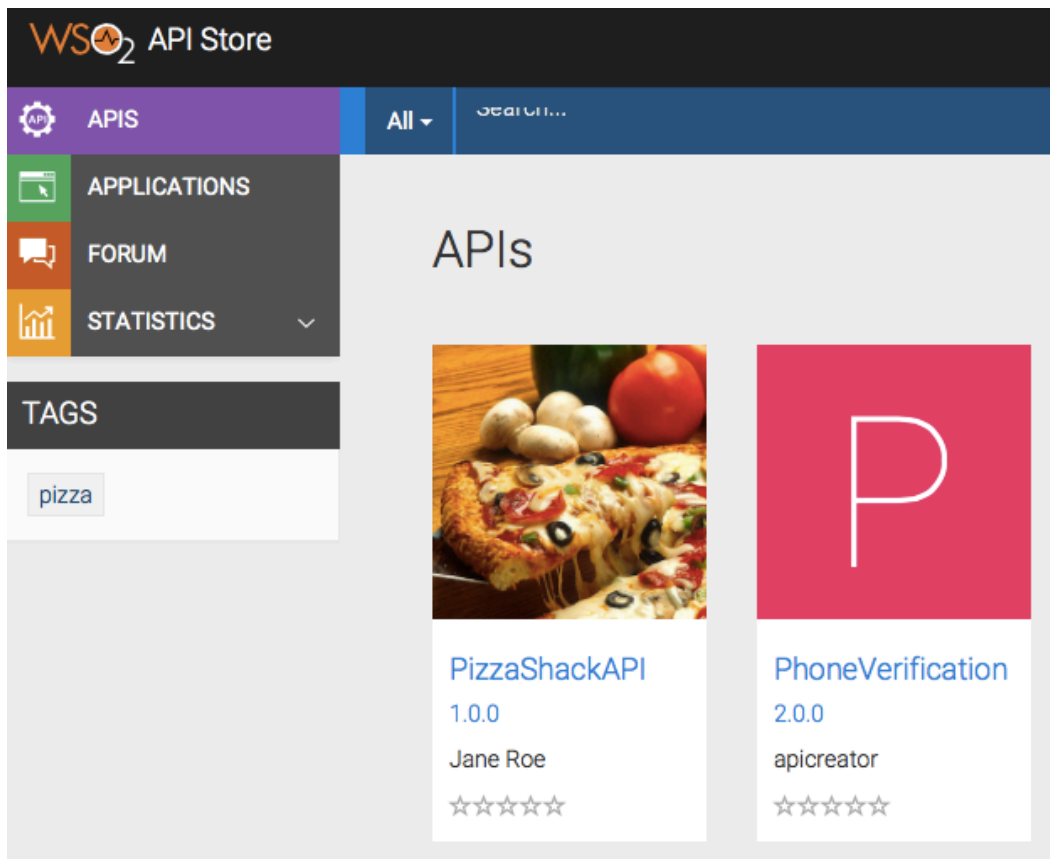
- The API opens. Go to its **Lifecycle** tab and click **Publish**.
PhoneVerification - 2.0.0

Current State: **CREATED**

☐ Require re-subscription when publish the API
☐ Deprecate old versions after publish the API

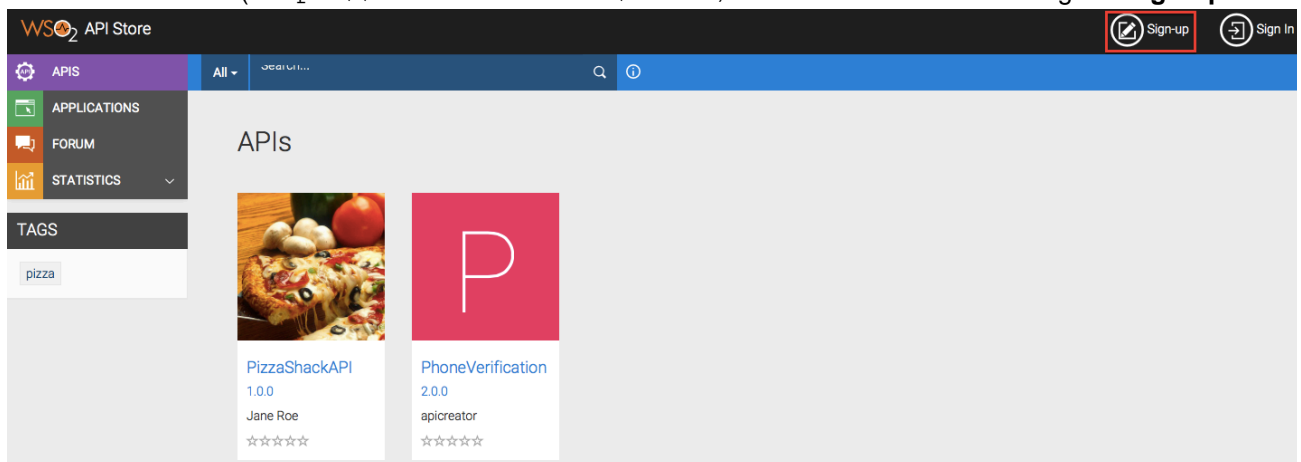
The check boxes mean the following:

- Require re-subscription when publish the API:** Invalidates current user subscriptions, forcing users to subscribe again.
 - Deprecate old versions after publish the API:** If selected, any prior versions of the API that are published will be set to the DEPRECATED state automatically.
- Go to the API Store (<https://<hostname>:9443/store>) using your browser and note that the PhoneVerification 2.0.0 API is visible under the **APIs** menu.



Subscribing to the API

1. Go to the API Store (<https://<hostname>:9443/store>) and create an account using the **Sign-up** link.



2. After signing up, sign in to the API Store and click the PhoneVerification 2.0.0 API that you published earlier.
3. Note that you can now see the subscription options. Select the default application and the Bronze tier. Click **Subscribe**.

PhoneVerification - 2.0.0



Version: 2.0.0
 By: apicreator
 Updated: 22/Jul/2016 21:27:16 PM IST
 Status: **PUBLISHED**
 Rating: ★★★★★

Applications

DefaultApplication

Tiers

Bronze

4. Once the subscription is successful, click **View Subscriptions** in the information message that appears to review your subscriptions.

Subscription Successful



You have successfully subscribed to the API.

[View Subscriptions](#)
[Stay on this page](#)

5. Click the **Production Keys** tab of the application and then click **Generate Keys** to generate an access token that you use later to invoke the API. If you have already generated keys before, click **Re-generate**.

WSO2 API Store

APPLICATIONS APPLICATION LIST EDIT

APIS FORUM STATISTICS

DefaultApplication

Details **Production Keys** Sandbox Keys Subscriptions

No Keys Found
 No keys are generated for this type in this application.

Grant Types
 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.

☒ SAML2
 ☒ IWA-NTLM
 ☐ Implicit
 ☒ Refresh Token

☒ Client Credential
 ☐ Code
 ☒ Password

Callback URL

Access token validity period

3600 Seconds

Generate keys

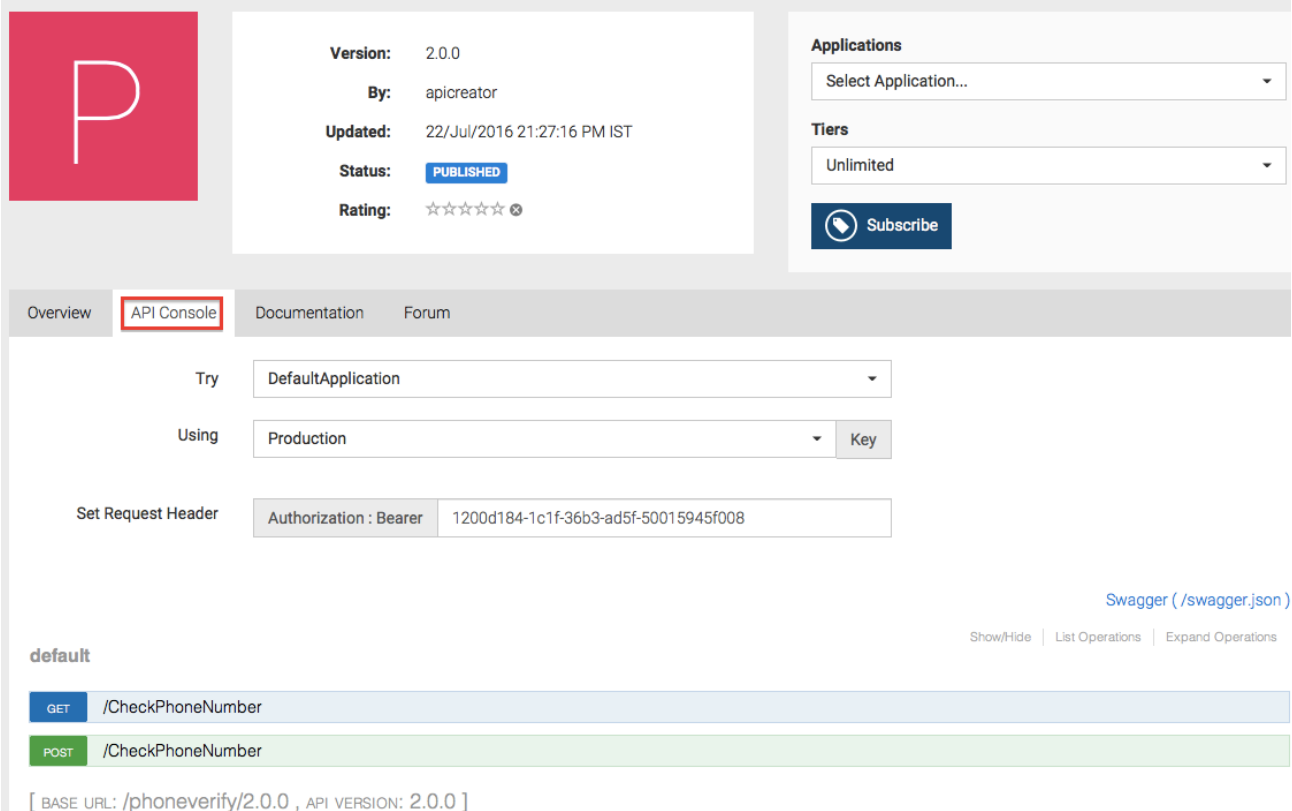
Tip : You can set a token validity period in the given text box. By default, it is set to one hour. If you set a minus value (e.g., -1), the token will never expire.

You are now successfully subscribed to an API. Let's invoke it.

Invoking the API

1. Click the **APIs** menu in the API Store and then click on the API that you want to invoke. When the API opens, go to its **API Console** tab.

PhoneVerification - 2.0.0



The screenshot shows the API console for 'PhoneVerification - 2.0.0'. The 'API Console' tab is selected and highlighted with a red box. The console displays the following information:

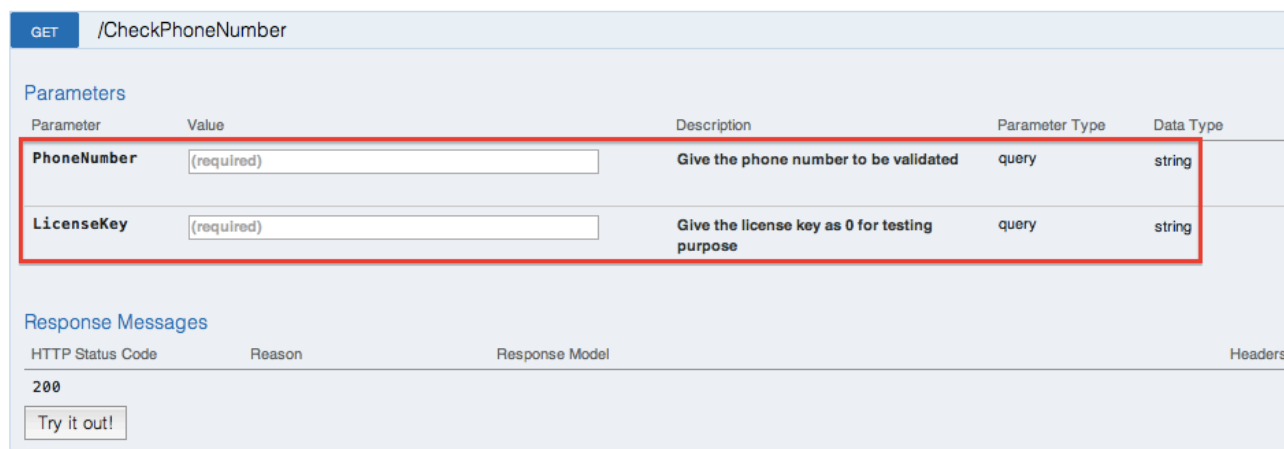
- Version:** 2.0.0
- By:** apicreator
- Updated:** 22/Jul/2016 21:27:16 PM IST
- Status:** PUBLISHED
- Rating:** ☆☆☆☆☆

On the right, there are dropdowns for 'Applications' (Select Application...) and 'Tiers' (Unlimited), along with a 'Subscribe' button.

Below the tabs, there are input fields for 'Try' (DefaultApplication), 'Using' (Production), and 'Set Request Header' (Authorization : Bearer 1200d184-1c1f-36b3-ad5f-50015945f008).

The 'default' section shows two methods: GET /CheckPhoneNumber and POST /CheckPhoneNumber. The GET method is highlighted with a blue box. Below the methods, the base URL is shown as [BASE URL: /phoneverify/2.0.0 , API VERSION: 2.0.0].

2. Expand the GET method of the resource CheckPhoneNumber. Note the parameters that you added when creating the interactive documentation now appear with their descriptions so that as a subscriber, you know how to invoke this API.



The screenshot shows the details for the GET /CheckPhoneNumber method. The 'Parameters' section is expanded, showing two parameters:

Parameter	Value	Description	Parameter Type	Data Type
PhoneNumber	(required)	Give the phone number to be validated	query	string
LicenseKey	(required)	Give the license key as 0 for testing purpose	query	string

The 'Response Messages' section shows a single response with status code 200. Below the response messages, there is a 'Try it out!' button.

3. Give sample values for the PhoneNumber and LicenseKey and click **Try it out** to invoke the API.

GET /CheckPhoneNumber

Parameters

Parameter	Value	Description	Parameter Type	Data Type
PhoneNumber	18006785432	Give the phone number to be validated	query	string
LicenseKey	0	Give the license key as 0 for testing purpose	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

4. Note the response for the API invocation. Since we used a valid phone number in this example, the response is valid.

Response Body

```
<?xml version="1.0" encoding="utf-8"?>
<PhoneReturn xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://ws.cdyne.com/PhoneVerify/query">
  <Company>Toll Free</Company>
  <Valid>true</Valid>
  <Use>Assigned to a code holder for normal use.</Use>
  <State>TF</State>
  <RC />
  <OCN />
  <OriginalNumber>18006785432</OriginalNumber>
  <CleanNumber>8006785432</CleanNumber>
  <SwitchName />
  <SwitchType />
  <Country>United States</Country>
  <CLLI />
</PhoneReturn>
```

Response Code

200

Response Headers

```
{
  "pragma": "no-cache",
  "content-type": "text/xml; charset=utf-8",
  "cache-control": "no-cache",
  "expires": "-1"
}
```

You have invoked an API using the API Console.

Monitoring APIs and viewing statistics

Both the API publisher and store provide several statistical dashboards.













A P I

P u b l i s h e r



s t a t i s t i c s

Statistics




APIs

-  Published APIs Over Time
-  API Usage
-  API Response Times
-  API Last Access Times
-  Usage by Resource Path
-  Usage by Destination
-  API Usage Comparison
-  API Throttled Requests
-  Faulty Invocations
-  API Latency Time
-  API Usage Across Geo Locations
-  API Usage Across User Agent

Applications

-  App Throttled Requests
-  Applications Created Over Time

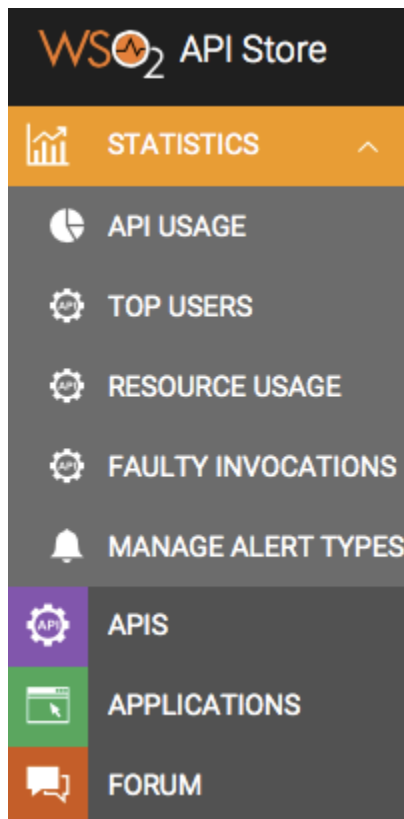
Subscriptions

-  API Subscriptions
-  Developer Signups Over Time
-  Subscriptions Created Over Time

A P I

S t o r e

s t a t i s t i c s



The steps below explain how to configure WSO2 API Manager Analytics with the API Manager. The statistics in these dashboards are based on data from WSO2 Data Analytics Server (DAS).

Let's do the configurations first.

Before you begin,

1. Download the WSO2 APIM Analytics distribution by clicking **ANALYTICS** in the [WSO2 API Management page](#). It is best to download and extract it to the same directory to which you downloaded WSO2 API Manager.
2. If you have the API Manager server running, stop the server.
3. If you are running on Windows, download the `snappy-java_1.1.1.7.jar` from [here](#) and copy the JAR file to the `<ANALYTICS_HOME>\repository\components\lib` directory.

1. To enable Analytics, open the `<APIM_HOME>/repository/conf/api-manager.xml` file and set the `Enabled` property under `Analytics` to `true` as shown below. Save this change.

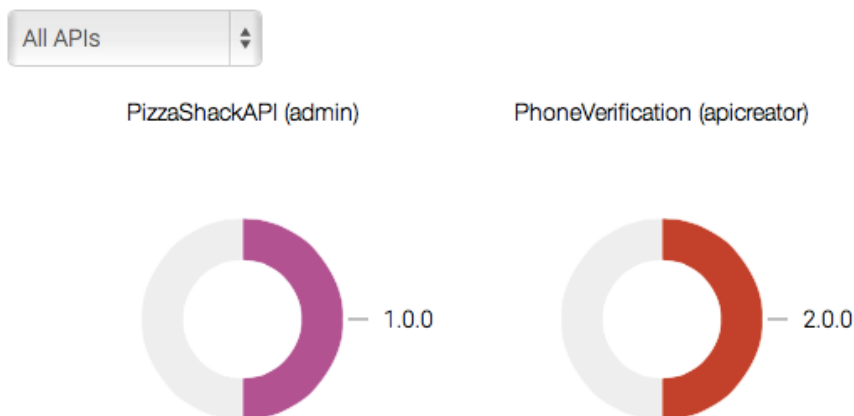
```
<Enabled>true</Enabled>
```

2. Open the `<APIM_HOME>/repository/conf/log4j.properties` file. Add `DAS_AGENT` to the end of the `log4j.rootLogger` property as shown in the example below.

```
log4j.rootLogger=ERROR, CARBON_CONSOLE, CARBON_LOGFILE, CARBON_MEMORY,
CARBON_SYS_LOG, ERROR_LOGFILE, DAS_AGENT
```

3. Start the WSO2 APIM Analytics server, and then start the API Manager server. To start a WSO2 product server, navigate to the `<PRODUCT_HOME>/bin` directory in your console and run one of the following scripts as relevant.
 - a. On Windows: `wso2server.bat --run`
 - b. On Linux/Mac OS: `sh wso2server.sh`
4. Invoke several APIs to generate some statistical data and wait a few seconds.
5. Connect to the API Publisher as a creator and click one of the statistical dashboards available in the **Statistics** menu. For example,

Overall API Subscriptions (Across All Versions)



The **Statistics** menu is available for API creators and shows statistics of all APIs. Additionally, API creators can also see the following:

- Statistics of the APIs created by them by selecting the **My APIs** option in the drop down menu above each table or graph.
- The subscriptions of each API by clicking **Manage Subscriptions**.
- The alerts that can be configured for their APIs by clicking **Manage Alert Types**.

This concludes the API Manager quick start. You have set up the API Manager and gone through the basic use cases of the product. For more advanced use cases, see the [Tutorials](#), [Deep Dive](#) and [Admin Guide](#) of the API Manager documentation.