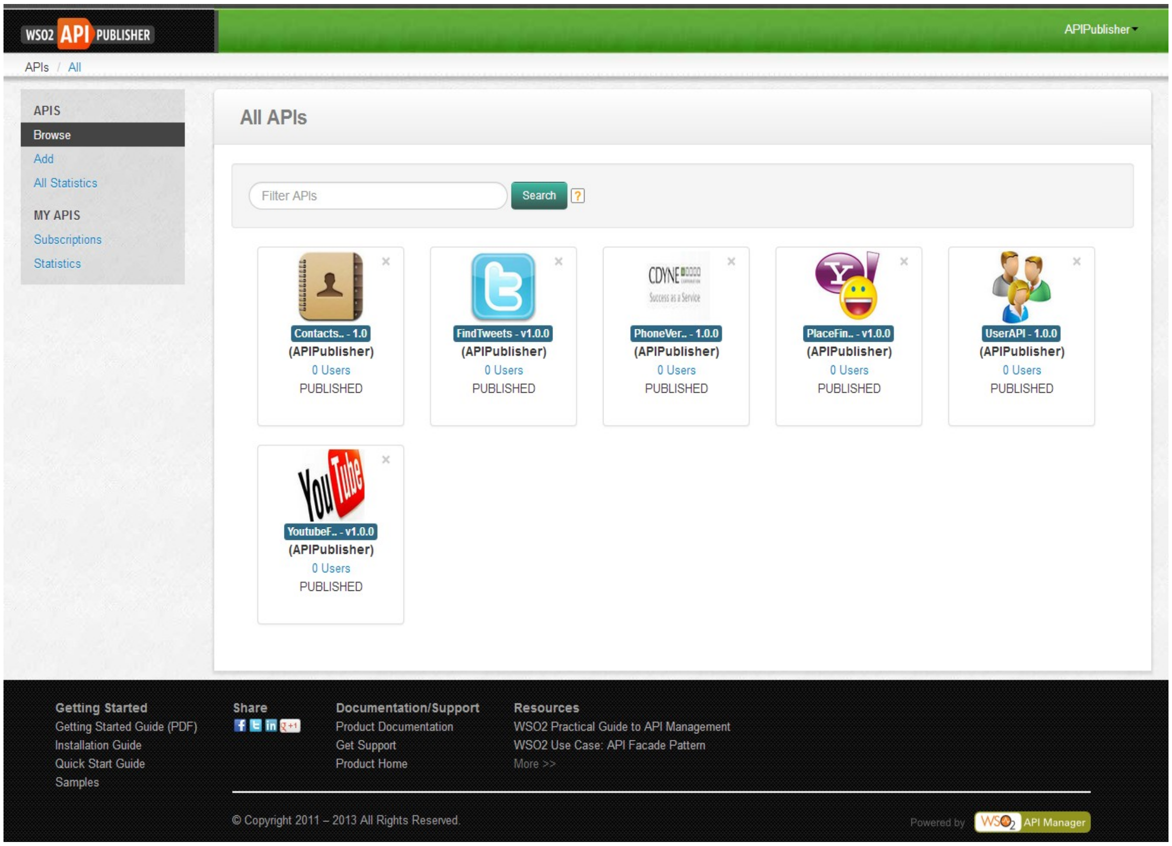
WSO2 API MANAGER 1.4.0

GETTING STARTED GUIDE



Introduction and Concepts

WSO2 API Manager is a complete solution for publishing APIs, creating and managing a developer community and for routing API traffic in a scalable manner. It leverages proven, production-ready, integration, security and governance components from the WSO2 Enterprise Service Bus, WSO2 Identity Server, and WSO2 Governance Registry products.

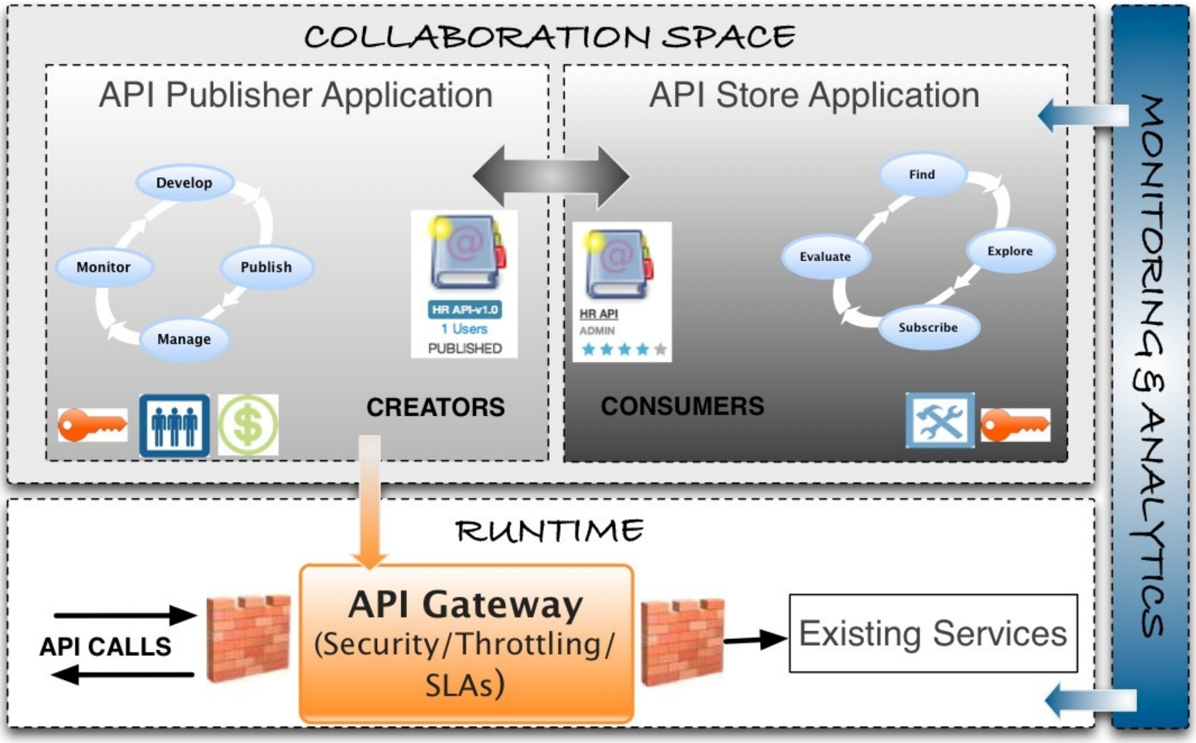
In addition, as it is powered by the WSO2 Business Activity Monitor, the WSO2 API Manager is ready for massively scalable deployment immediately.

As with all WSO2 products, the WSO2 API Manager is 100% open source.

Components

The API manager product comprises the following components:

- API Gateway: to secure, protect, manage, and scale API calls. The API gateway is a simple API proxy which intercepts API requests and applies policies such as throttling and security checks. It is also instrumental in gathering API usage statistics.
- API Publisher: enables API providers to easily publish their APIs, share documentation, provision API keys, and gather feedback on APIs features, quality and usage.
- API Store: provides a space for consumers to self-register, discover APIs functionality, subscribe to APIs, evaluate them and interact with API publishers.



Additionally, statistics are provided by the monitoring component, which integrates with our Business Activity Monitoring (BAM) solution. The BAM solution is deployed separately and analyzes events generated by the API manager (no specific configuration is required at the API manager level to enable this functionality).

Users and Roles

The API manager offers three distinct community roles:

- Creator: a creator will typically be a person in a technical role who understands the technical aspects of the API (interfaces, documentation, versions, how it will be exposed by API gateway) and uses the API publisher web application to provision APIs into the API store. The creator will use the API store to consult ratings and feedback provided by API users. Creator can add APIs to the store but cannot manage their lifecycle (i.e. make them visible to the outside world).

- Publisher: the publisher typically manages a set of APIs across the enterprise or business unit and controls the API lifecycle and monetization aspects. The publisher is also interested in usage patterns for APIs and as such has access to all API statistics.

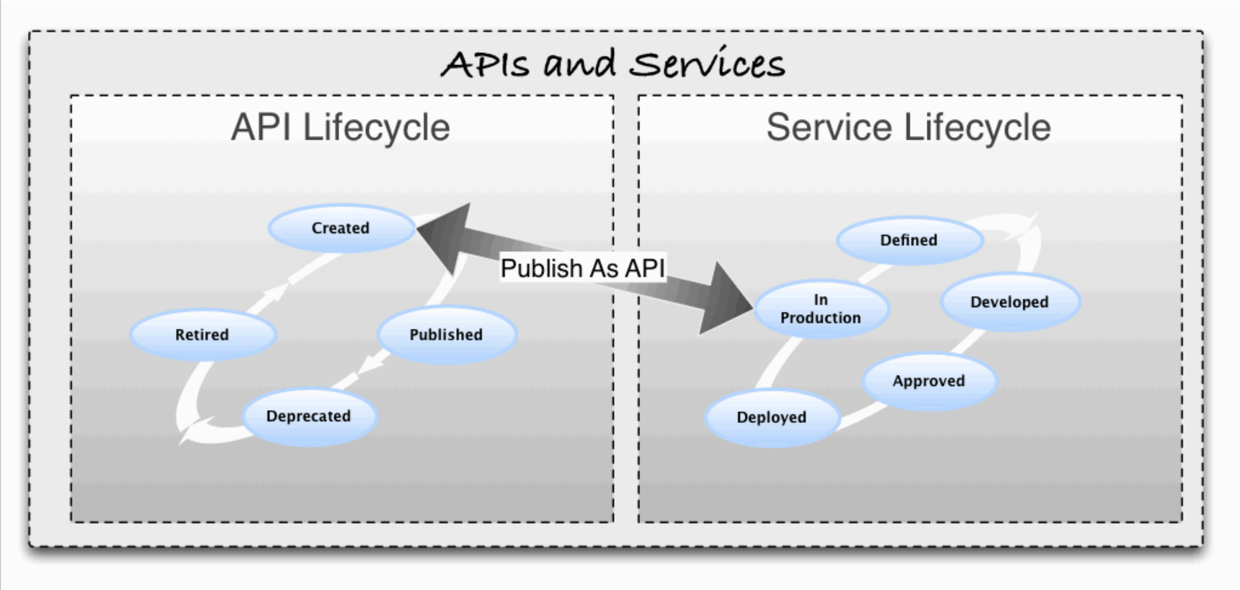
- Consumer: the consumer uses the API store to discover APIs, consult the documentation and forums as well as rate/comments on the API. He/she subscribes to APIs to obtain an API key.

API Lifecycle

An API is the published interface, while the service is the implementation running in the back-end. APIs have their own lifecycle, independently from the back-end service they rely on. This lifecycle is exposed in the API publisher web application and managed by the API publisher role.

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

- CREATED: API metadata has been added to API store, but it is not visible to subscribers yet, nor deployed to the API gateway
- PUBLISHED: API is visible in API store, and eventually (if the “Propagate Changes to API Gateway” option is selected at publishing time)
- DEPRECATED: API is still deployed into API gateway (i.e. available at runtime to existing users) but not visible to subscribers. An API can automatically be deprecated when a new version is published.
- RETIRED: API is unpublished from the API gateway and deleted from the store.
- BLOCKED: Access is temporarily blocked. Runtime calls are blocked and the API is not shown in the API store anymore.



The API and Service life cycles can be managed in the same governance registry/repository and linked automatically. This feature is available in version 4.5 of the WSO2 Governance Registry.

Applications

The application concept is used to decouple the consumer from the APIs and allows to :

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

You must create an application to subscribe to an API. The product comes out of the box with a Default Application and you can create as many applications as you need.

Throttling Tiers

Throttling Tiers are associated to an API at subscription time and define the throttling limits enforced by the API gateway (for example, 10 tx/sec). At the publisher level, the user defines the list of tiers which are available for a given API.

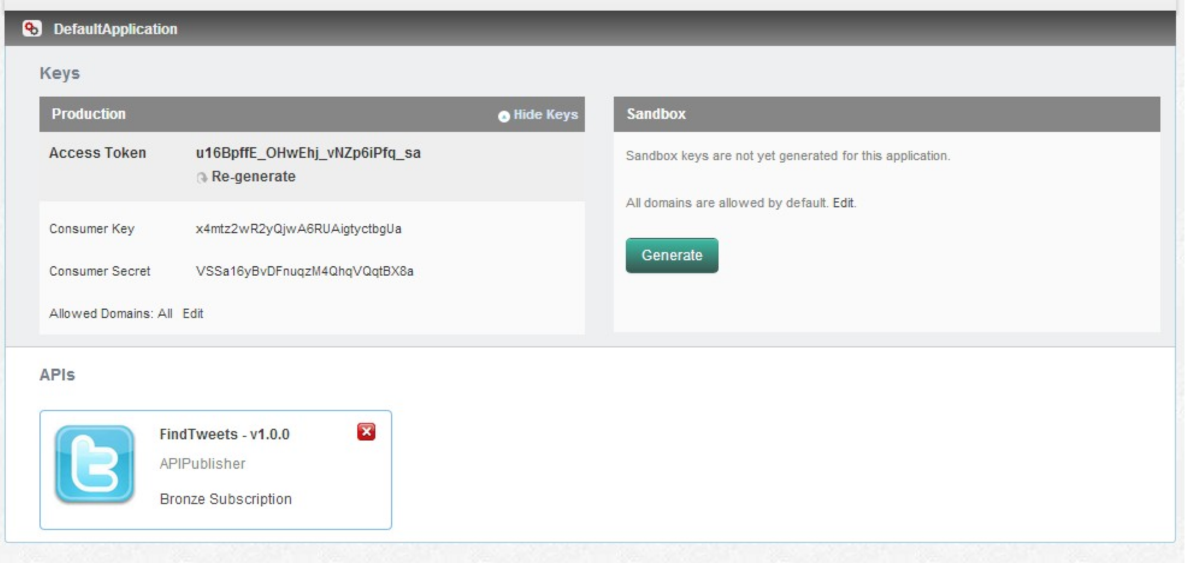
The product comes with three defined tiers (Gold/Silver/Bronze) and a special tiers called Unlimited tiers, which can be disabled by editing the <TierManagement> node of the api-manager.xml file.

To edit the existing tiers or create your own tiers, check the following blog entry: http://sumedha.blogspot.fr/2012/06/how-to-add-new-throttling-tier-to-wso2.html

API Keys

We support two scenarios for authentication :

1. We use an access token to identify and authenticate a whole application
2. We use an access token to identify the final user of an application (for example, the final user of a mobile application deployed on many different devices).



DefaultApplication

Keys

Production

Access Token u16BpffE_OHwEhj_vNZp6iPfq_sa

Consumer Key x4mtz2wR2yQjwA6RUAigtyctbgUa

Consumer Secret VSSa16yBvDFnuqzM4QhqVQqtBX8a

Allowed Domains: All Edit

Sandbox

Sandbox keys are not yet generated for this application.

All domains are allowed by default. Edit.

APIs

FindTweets - v1.0.0

APIPublisher

Bronze Subscription

Application Access Token

Application access tokens are generated by the API consumer and must be passed in the incoming API requests. We leveraged the OAuth2 standard to provide a simple, easy to use key management mechanism. The API key is a simple string, which must be passed as an HTTP header (like this: "Authorization: Bearer NtBQkXoKElu0H1a1fQ0DWfo6IX4a") and works equally well for SOAP and REST calls.

Application access tokens are generated at the application level and valid for all APIs which are associated to this application. Application access tokens have a fixed expiration time, which is by default set to 60 minutes. You can update this expiration time to a much longer time, such as several weeks (FYI, 4 weeks are 2419200 seconds!) .Consumers have the ability to re-generate the access token directly from the API store interface.

The default expiration time can be changed by editing the <apimgr_root_install>/repository/conf/identity.xml file and changing the value for <ApplicationAccessTokenDefaultValidityPeriod>.Importantly you can set the application access token expiration time as Never Expired value. For that you need to set the <ApplicationAccessTokenDefaultValidityPeriod> configuration value as a minus value.

Application User Access Token

We also support the generation of access tokens on demand through a specific API called token API [In previous API manager releases this API is referred as login API; the login API is dep -recated from 1.3]. In case a token expires, access tokens can be refreshed through this token

API.

Application user access tokens have a fixed expiration time, which is by default set to 60 minutes.

You can update this expiration time to a much longer time, such as several weeks (FYI, 4 weeks are 2419200 seconds!) .The default expiration time can be changed by editing the <apimgr_root_install>/repository/conf/identity.xml file and changing the value for <AccessTokenDefaultValidityPeriod>.

For generation of access tokens, token API takes 4 parameters:

1. Grant Type
2. Username
3. Password
4. Scope

To generate a new access token, a token API call must be issued with above parameters where grant_type=password. The Token API then returns two tokens: an access token and a refresh token. The access token can then be stored in 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.

In case the token expires, the application will have to refresh the token by issuing the token API call with above parameters where grant_type=refresh_token , passing the refresh token as parameter.

To refresh access tokens, login API takes 2 parameters:

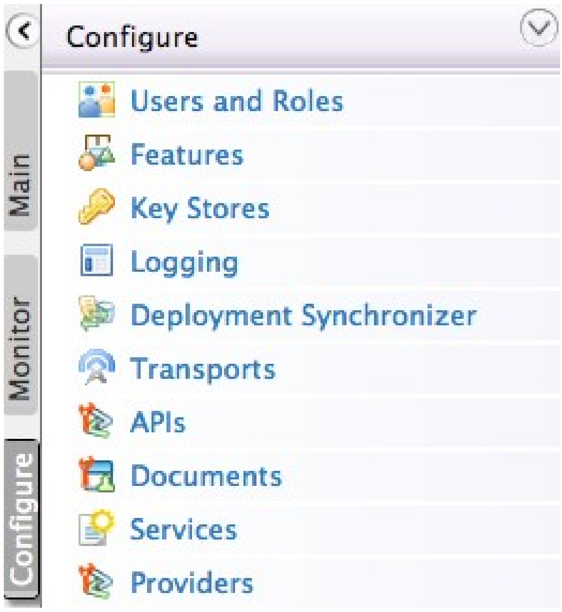
1. Grant Type
2. Refresh Token

Defining Users and Roles

You can use the API manager as an administration user (admin/admin), which can play the creator, publisher and subscriber roles. In this section, we explain how to setup custom roles and users.

Defining roles

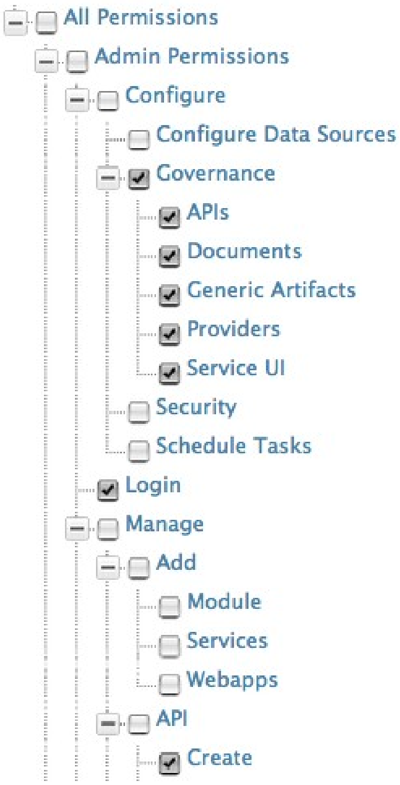
To define the creator role, you need to:



- Log into the API Manager admin console , which is available by default at : https://hostname:9443/carbon. You can login to the console using the default admin/admin credentials.
- Select the configure tab on the left side
- Select Users and Roles
- Select Roles
- Click Add New Role
- Provide creator as the role name
- Click Next - You will be presented with a list of permissions. For the creator role, you need to select the following permissions:

1. Configure > Governance and all underlying permissions
2. Login
3. Manage > API > Create
4. Manage > Resources > Govern and all underlying permissions.

- Click Finish (at the bottom of the page)



- □ All Permissions
  - □ Admin Permissions
    - □ Configure
      - □ Configure Data Sources
      - ☑ Governance
        - ☑ APIs
        - ☑ Documents
        - ☑ Generic Artifacts
        - ☑ Providers
        - ☑ Service UI
      - □ Security
      - □ Schedule Tasks
    - ☑ Login
    - □ Manage
      - □ Add
        - □ Module
        - □ Services
        - □ Webapps
      - □ API
        - ☑ Create

- Repeat steps 7 and 8 to create the publisher role, with the following permissions:

    5. Login
    6. Manage > API > Publish

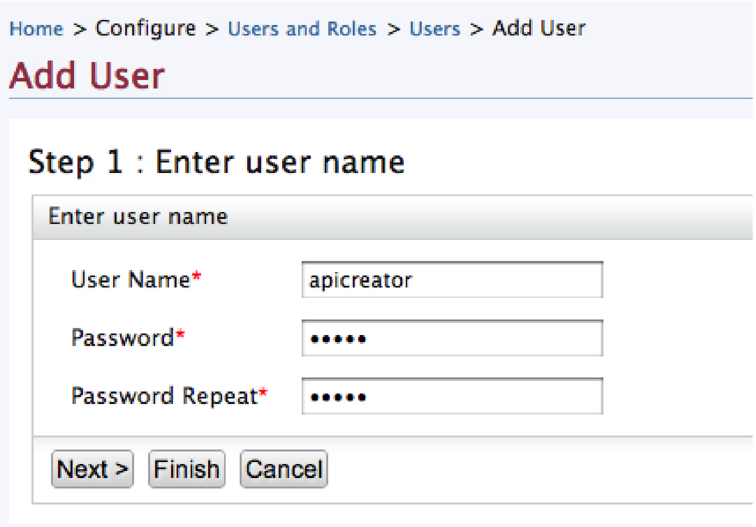
The subscriber role is already defined out of the box, as it's used in the self-registration process. If you wish to create a different subscriber role, you must:

1. Create the subscriber role by repeating steps 7 and 8 above with the following permissions
    1. Login
    2. Manage > API > Subscribe
2. Edit accordingly the <SelfSignUp> node in the <install_root>/repository/conf/api-manager.xml file, so that users created via the self-sign up mechanism are automatically assigned this role.

Defining users via the admin console

You can now create a user in each of those roles. To do so:

9. Go to Configure > Users and Roles
10. Click Users
11. Click Add New User
12. Provide user name and password
13. Click Next



Home > Configure > Users and Roles > Users > Add User

Add User

Step 1 : Enter user name

Enter user name

User Name* apicreator

Password* •••••

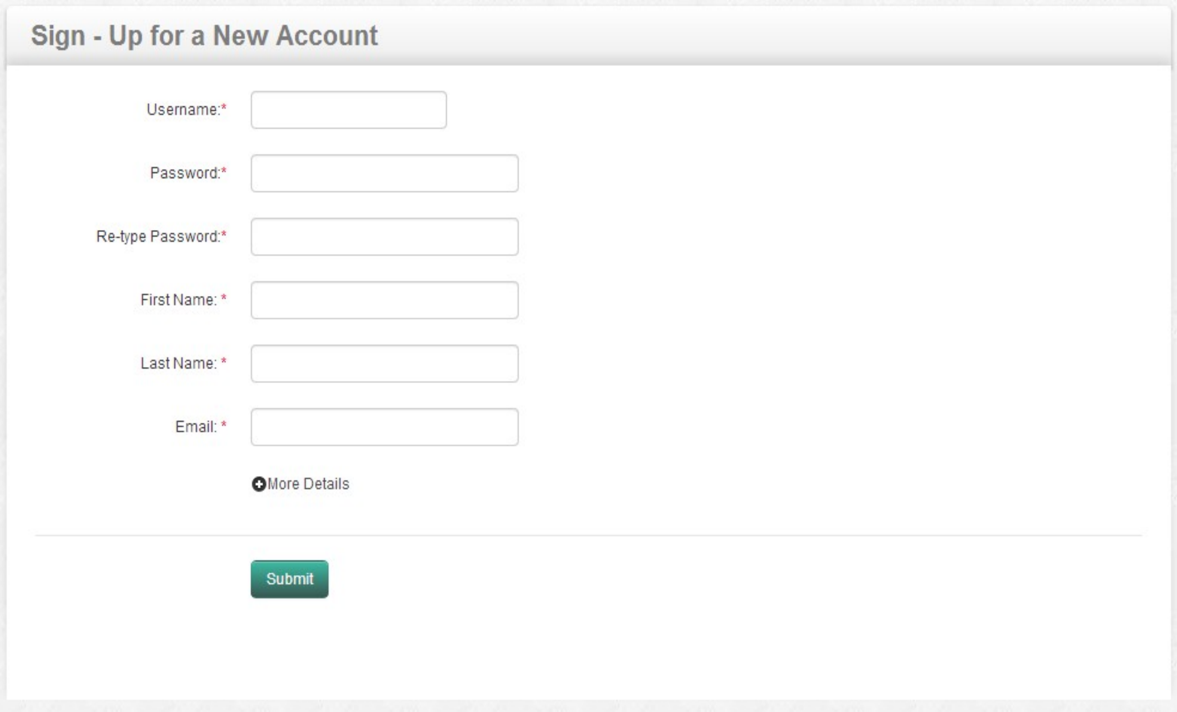
Password Repeat* •••••

14. Select the role you want to assign to the user (creator, publisher or subscriber)
15. Click Finish.
16. Repeat those steps to create a user in the publisher and subscriber roles.

Defining users via self-registration

When a user connects to the API store for the first time, they can self-register. To do so, they need to:

1. Open the API Store web application from https://<YourHostName>:9443/store
2. Click Sign-Up at the top right of the window
3. Enter a user-id and password. The password must follow certain validity rules, as per the image below:



Publishing APIs

The API provider web application can be used by API creators to provision APIs into the API store. In this section, we explain how to define and attach documentation to an API.

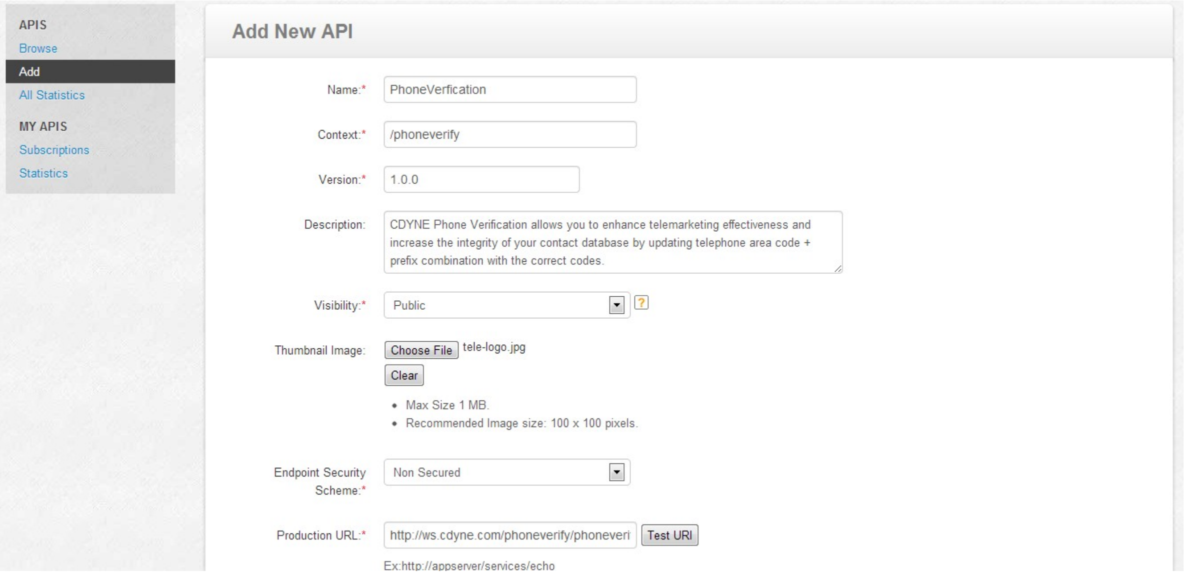
The Phone Number validation API

Along this guide, we work with a service exposed by the Cdyne services provider (www.wdyne.com). We use their phone validation service, which has SOAP and REST interfaces and is documented using a WSDL file. This service is documented at : http://wiki.cdyne.com/index.php/Phone_Verification.

Adding an API to the Store

To add the API to the store, follow those steps:

1. Open the API Publisher web application from https://<YourHostName>:9443/publisher
2. Login using the user in creator role you defined previously (in our case, apicreator)
3. Click Add
4. Provide information on the API as per the table below.



Add New API

Name:* PhoneVerfication

Context:* /phoneverify

Version:* 1.0.0

Description: CDYNE Phone Verification allows you to enhance telemarketing effectiveness and increase the integrity of your contact database by updating telephone area code + prefix combination with the correct codes.

Visibility:* Public

Thumbnail Image: tele-logo.jpg

- Max Size 1 MB.
- Recommended Image size: 100 x 100 pixels.

Endpoint Security Scheme:* Non Secured

Production URL:* http://ws.cdyne.com/phoneverify/phoneveri

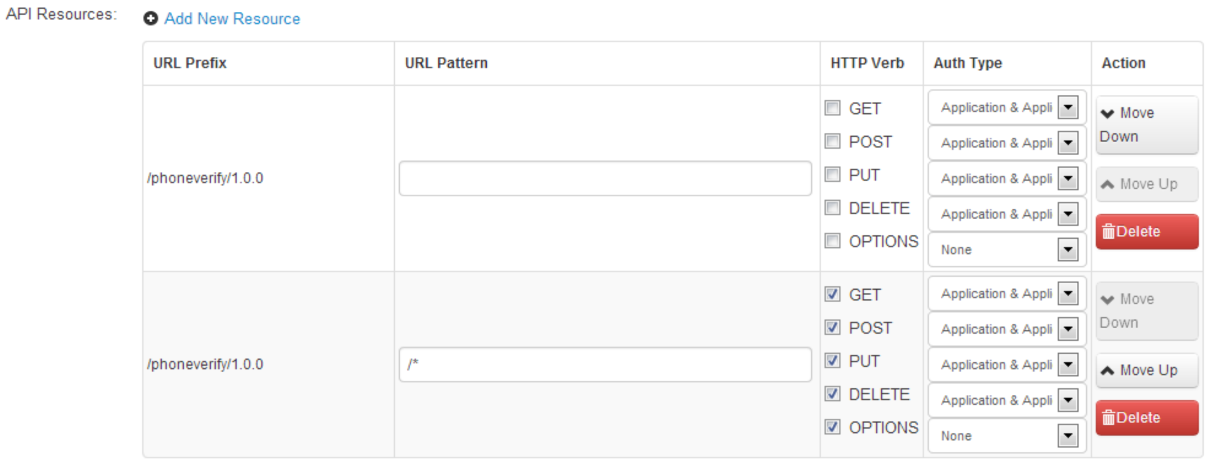
Ex:http://appserver/services/echo

FIELD	VALUE	DESCRIPTION
Name	PhoneVerification	Name of API as you want it to appear in the API store
Context	/phoneverify	URI context path that is used by to API consumers
Version	1.0.0	API version (in the form of version.major.minor)
Description	Text	High level description of API functionality
Visibility	Public/Restricted	If the API can be accessed by everyone,then select the visibility as 'Public'. Else select it as 'Restricted' and enter the user roles to which the API can be accessible from both API publisher and store.
Thumbnail Image	Image file	Icon to be displayed in API store (can be jpeg, tiff, png format)
Endpoint Security Scheme	Non-Secured/Secured	If the back-end service is a secured service, select 'Secured' and enter the credentials for secured service in appearing text boxes. Else keep as Non-Secured.
Production URL	URL	Endpoint of the back-end service URL, here: http://ws.cdyne.com/phoneverify/phoneverify.asmx
Sandbox URL	URL	Endpoint of sandbox (testing) back end service. A sandbox URL is meant to be used for online testing of an API with easy access to an API key.
WSDL	URL	URL of WSDL file (describing API interface) http://ws.cdyne.com/phoneverify/phoneverify.asmx?wsdl
WADL	URL	URL to WADL file (describing API interface)

FIELD	VALUE	DESCRIPTION
Tags	String	One of more tags separated by comma. Tags are used to group/search for API
Tier Availability	Bronze/Gold/Silver/Unlimited	The API can be available at different level of service; you can select multiple entries from the list. At subscription time, the consumer chooses which tier they are interested in.
Business Owner and Email	String	Information about the person responsible for this API at the business level
Technical Owner and Email	String	Information about the person responsible for this API at the technical level

API Resources

An API is made up of one or more resources. Each resource handles a particular type of requests. A resource is analogous to a method (function) in a larger API.



API Resources: Add New Resource

URL Prefix	URL Pattern	HTTP Verb	Auth Type	Action
/phoneverify/1.0.0		□ GET □ POST □ PUT □ DELETE □ OPTIONS	Application & Appli Application & Appli Application & Appli Application & Appli None	Move Down Move Up Delete
/phoneverify/1.0.0	/*	☑ GET ☑ POST ☑ PUT ☑ DELETE ☑ OPTIONS	Application & Appli Application & Appli Application & Appli Application & Appli None	Move Down Move Up Delete

API resources can accept following optional attributes:

1. verbs: Specifies the HTTP verbs a particular resource would accept. Allowed values are GET, POST, PUT, DELETE. Multiple values can be specified.

2. uri-template: A URI template as defined in http://tools.ietf.org/html/rfc6570 (eg: /phoneverify/{phoneNumber})

3. url-mapping: A URL mapping as defined as per the servlet specification (extension mappings, path mappings and exact mappings)

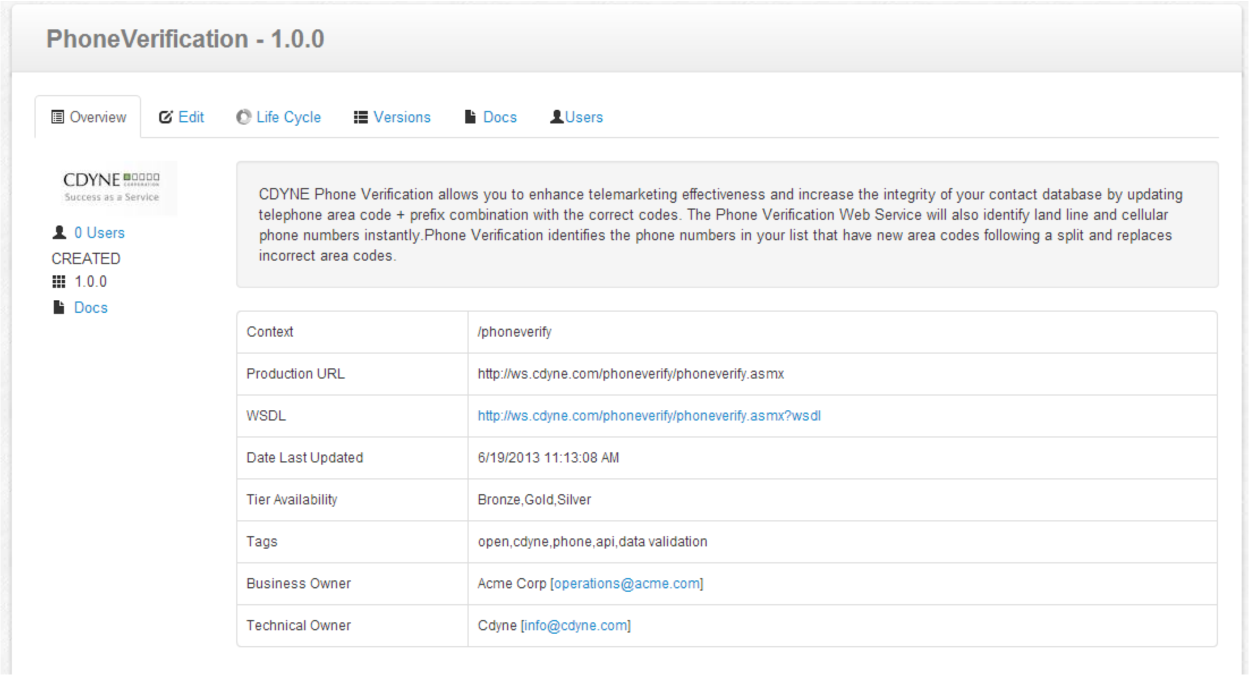
4. Auth-Type: Specifies the Resource level authentication along HTTP verbs. Auth-type can be None,Application or Application User.

   1) None : Can access the particular API resource without any access tokens
   2) Application: Application access token required to access the API resource
   3) Application User: User access token required to access the API resource

Once a request has been accepted by a resource, it will be mediated through an in-sequence. Any response from the back-end is handled through the out-sequence. Fault sequence is used to me diate any errors that might occur in either sequence. Default in-sequence, out-sequence and fault sequence are generated when the API is published.

Adding Documentation

Once the API has been created, you can click on the icon and open its details. You see something similar to the image below:



PhoneVerification - 1.0.0

CDYNE Phone Verification allows you to enhance telemarketing effectiveness and increase the integrity of your contact database by updating telephone area code + prefix combination with the correct codes. The Phone Verification Web Service will also identify land line and cellular phone numbers instantly.Phone Verification identifies the phone numbers in your list that have new area codes following a split and replaces incorrect area codes.

Context	/phoneverify
Production URL	http://ws.cdyne.com/phoneverify/phoneverify.asmx
WSDL	http://ws.cdyne.com/phoneverify/phoneverify.asmx?wsdl
Date Last Updated	6/19/2013 11:13:08 AM
Tier Availability	Bronze,Gold,Silver
Tags	open,cdyne,phone,api,data validation
Business Owner	Acme Corp [operations@acme.com]
Technical Owner	Cdyne [info@cdyne.com]

You can now switch to the Docs tab and add documentation to the API. Documentation can be provided inline,via a URL or as a file. For inline documentation, you can edit the contents directly from the API publisher interface.

Several documents types are available:

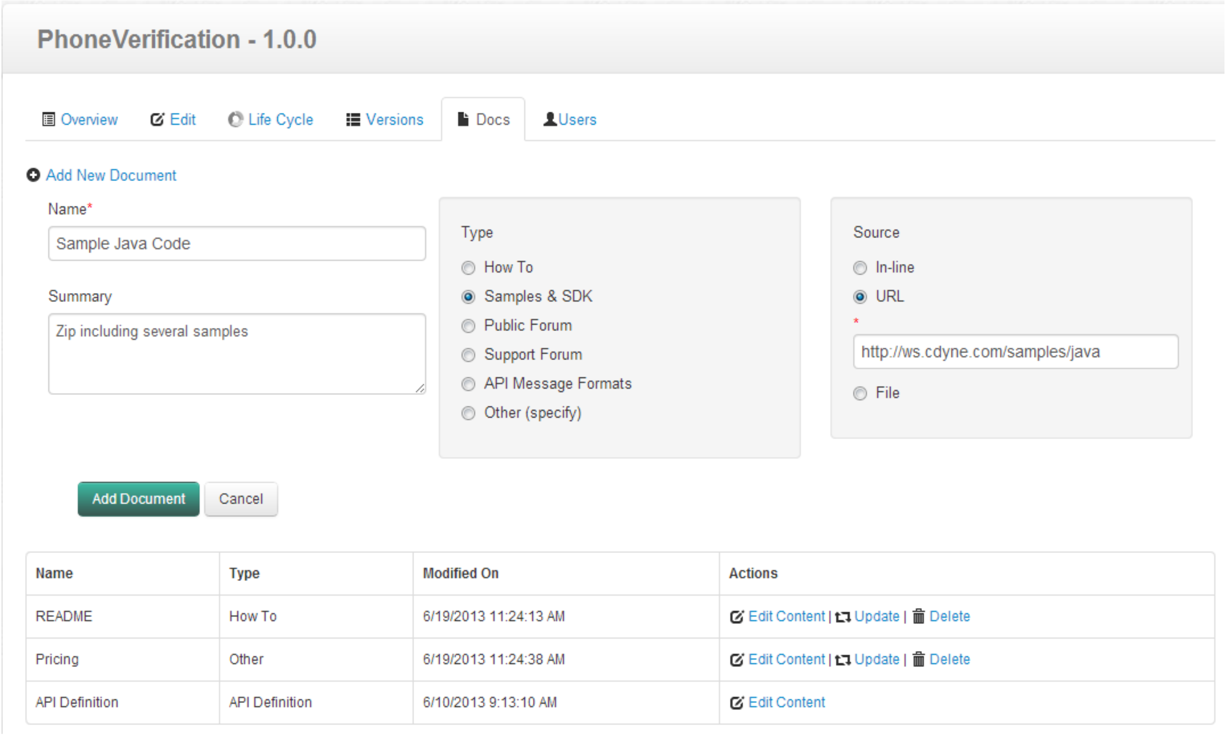
5. How To
6. Samples and SDK
7. Public forum / Support forum (external link only)
8. API message formats
9. Other

To create a How-To document:

- Select the How To type
- Provide a name for the document
- Provide a short description of the document (this will appear in the API store)
- Select whether the document is stored inline or provided via a URL

- Click Add New Document

Once the document has been added, you can edit the contents by clicking on the Edit Content link. An embedded editor allows you to edit the document contents.



PhoneVerification - 1.0.0

Name*: Sample Java Code

Summary: Zip including several samples

Type
○ How To
● Samples & SDK
○ Public Forum
○ Support Forum
○ API Message Formats
○ Other (specify)

Source
○ In-line
● URL
*
http://ws.cdyne.com/samples/java
○ File

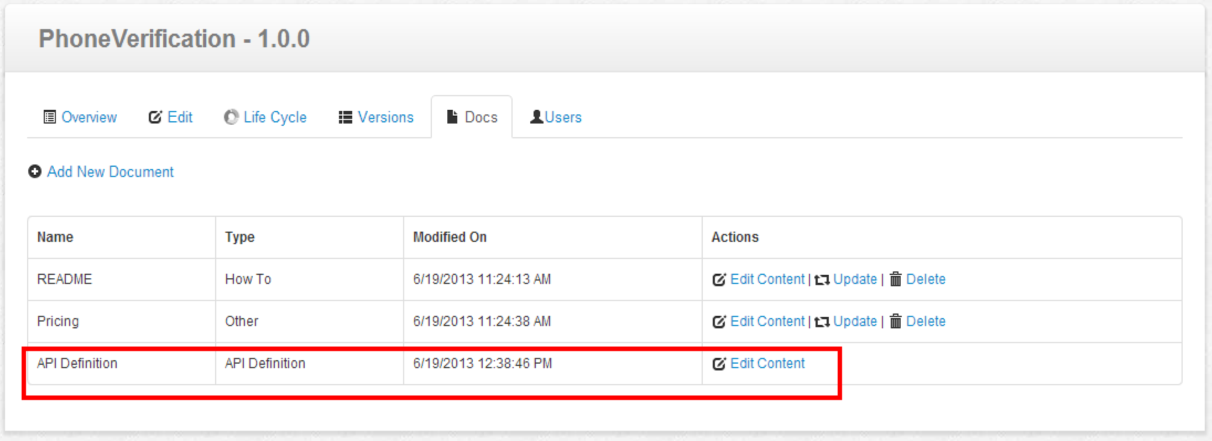
Name	Type	Modified On	Actions
README	How To	6/19/2013 11:24:13 AM	Edit Content | Update | Delete
Pricing	Other	6/19/2013 11:24:38 AM	Edit Content | Update | Delete
API Definition	API Definition	6/10/2013 9:13:10 AM	Edit Content

Adding Interactive Documentation Support with Swagger

WSO2 API Manager 1.4.0 provides interactive documentation support through the integration with Swagger. Swagger is a specification and complete framework implementation for describing, producing, consuming, and visualizing RESTful web services. In Swagger, when APIs are described in simple static JSON representation, they can be loaded through Swagger UI which in turn provide the interactive documentation.

When an API is created in API Publisher, the JSON representation of that API will be automatically generated and saved into the registry as "API Definition". This "API Definition" is describing the API with the information provided at the API creation level.

This "API Definition" of a particular API will be listed in “Doc” tab as below and can be modified when required.



PhoneVerification - 1.0.0

Name	Type	Modified On	Actions
README	How To	6/19/2013 11:24:13 AM	Edit Content | Update | Delete
Pricing	Other	6/19/2013 11:24:38 AM	Edit Content | Update | Delete
API Definition	API Definition	6/19/2013 12:38:46 PM	Edit Content

Users can update/customize the automatically generated API definition for each API. For that go to "Doc" tab of 'PhoneVerification' API. There it can be found the API Definition which containing the JSON representation of the API. If you want to do any modifications to Paths, Parameters, descriptions etc user can do it by editing the JSON representation of API Definition.

For example,in the 'PhoneVerification' API,we have changed the path for all the HTTP methods of API definition from '/phoneverify/1.0.1/' as '/phoneverify/1.0.1/CheckPhoneNumber'.

API Definition

```
{
   "apiVersion": "1.0.0",
  "swaggerVersion": "1.1",
  "basePath": "http://localhost:8280",
  "resourcePath": "PhoneVerification",
  "apis": [
     {
        "path": "/PhoneVerification/1.0.0/CheckPhoneNumber",
        "description": "no-info",
        "operations": [
           {
             "httpMethod": "GET",
             "summary": "PhoneVerify",
             "nickname": "getDetails",
             "parameters": [
                {
                   "name": "Authorization",
                   "description": "Access Token",
                   "paramType": "header",
                   "required": true,
                   "allowMultiple": false ,
     "dataType": "String"
                 },
                 {
                   "name": "Payload",
                   "description": "Request Payload",
                   "paramType": "path",
                   "required": true ,
                   "allowMultiple": false ,
                   "dataType": "String"
                }
             ]
           },
           {
             "httpMethod": "PUT",
             "summary": "no-info",
             "nickname": "no-info",
             "parameters": [
                {
                   "name": "Authorization",
                   "description": "Access Token",
                   "paramType": "header",
                   "required": true ,
                   "allowMultiple": false ,
                   "dataType": "String"
                },
                {
                   "name": "Payload",
                   "description": "Request Payload",
                   "paramType": "body",
                   "required": true ,
                   "allowMultiple": false ,
                   "dataType": "String"
                }
             ]
           },
           {
             "httpMethod": "POST",
             "summary": "no-info",
             "nickname": "no-info",
             "parameters": [
                {
                   "name": "Authorization",
                   "description": "Access Token",
                   "paramType": "header",
                   "required": true ,
                   "allowMultiple": false ,
                   "dataType": "String"
                },
                {
                   "name": "Payload",
                                     "description": "Request Payload",
                "paramType": "body",
                "required": true ,
                "allowMultiple": false ,
                "dataType": "String"
             }
           ]
         },
         {
           "httpMethod": "DELETE",
           "summary": "no-info",
           "nickname": "no-info",
           "parameters": [
             {
                "name": "Authorization",
                "description": "Access Token",
                "paramType": "header",
                "required": true ,
                "allowMultiple": false ,
                "dataType": "String"
             },
             {
                "name": "Payload",
                "description": "Request Payload",
                "paramType": "body",
                "required": true ,
                "allowMultiple": false ,
                "dataType": "String"
             }
           ]
         },
         {
           "httpMethod": "OPTIONS",
           "summary": "no-info",
           "nickname": "no-info",
           "parameters": [
             {
                "name": "Payload",
                "description": "Request Payload",
                "paramType": "body",
                "required": true ,
                "allowMultiple": false ,
                "dataType": "String"
             }
           ]
         }
       ]
     }
     ]}
```

The API is now ready to be published. This has to be done by a user in the publisher role.

Publishing the API

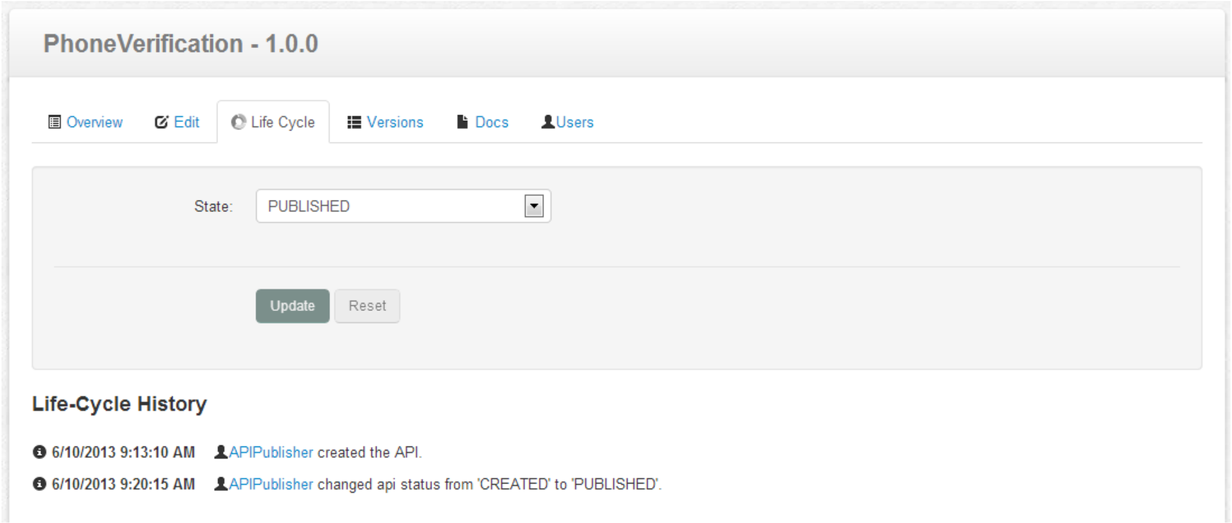
To publish the API:

1. Logout as apicreator and login as apipublisher.
2. Click on the PhoneVerification API - You can see that an additional tab is now available, allowing us to manage the API lifecycle
3. To publish the API, select the PUBLISHED state from the list. The following options are available:

   - Propagate changes to API Gateway: define an API proxy in the API gateway runtime component (API will be exposed to the consumers via the API gateway). Note if you do not select this option, you are only changing the API metadata and the API gateway will have to be configured manually according to the information published in the API store.
   - Deprecate Old Versions (only appears when a new version is published): automatically deprecates (moves to deprecated state) prior versions of this API.
   - Require re-subscription (only appears when a new version is published): forces users to subscribe to the new version. If this option is not selected, a user will be automatically subscribed to the new version, provided he/she was subscribed to the previous version.

4. Click Update.

The API is now published and visible to consumers in the API store. The API life cycle history is visible at the bottom of the page.



State: PUBLISHED

Life-Cycle History

6/10/2013 9:13:10 AM APIPublisher created the API.

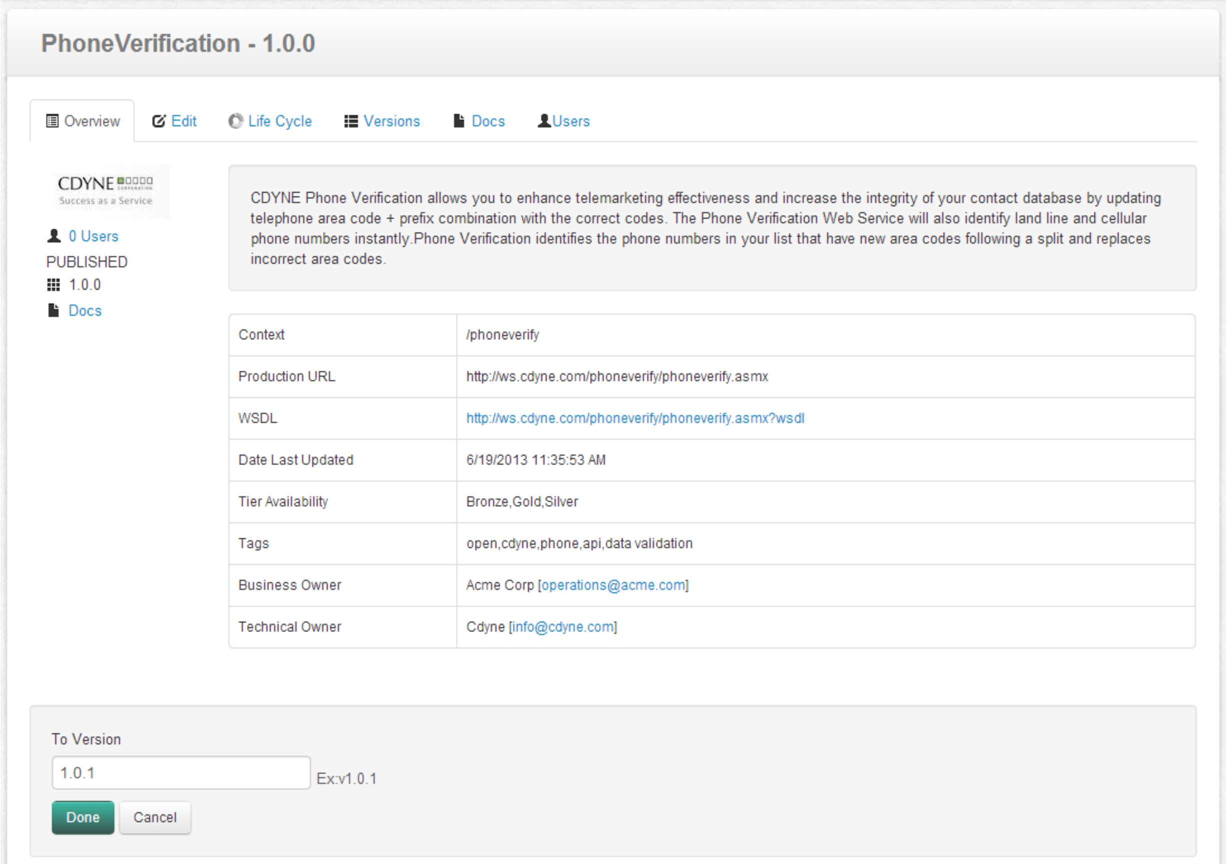
6/10/2013 9:20:15 AM APIPublisher changed api status from 'CREATED' to 'PUBLISHED'.

API Versioning

You can create a copy of an API from the Overview tab, if you are in the creator role.

To create a new version:

1. Click Copy
2. Specify a new version number (of the version.major.minor format)
3. Click Done.



CDYNE Phone Verification allows you to enhance telemarketing effectiveness and increase the integrity of your contact database by updating telephone area code + prefix combination with the correct codes. The Phone Verification Web Service will also identify land line and cellular phone numbers instantly.Phone Verification identifies the phone numbers in your list that have new area codes following a split and replaces incorrect area codes.

Context	/phoneverify
Production URL	http://ws.cdyne.com/phoneverify/phoneverify.asmx
WSDL	http://ws.cdyne.com/phoneverify/phoneverify.asmx?wsdl
Date Last Updated	6/19/2013 11:35:53 AM
Tier Availability	Bronze,Gold,Silver
Tags	open,cdyne,phone,api,data validation
Business Owner	Acme Corp [operations@acme.com]
Technical Owner	Cdyne [info@cdyne.com]

To Version: 1.0.1 Ex:v1.0.1

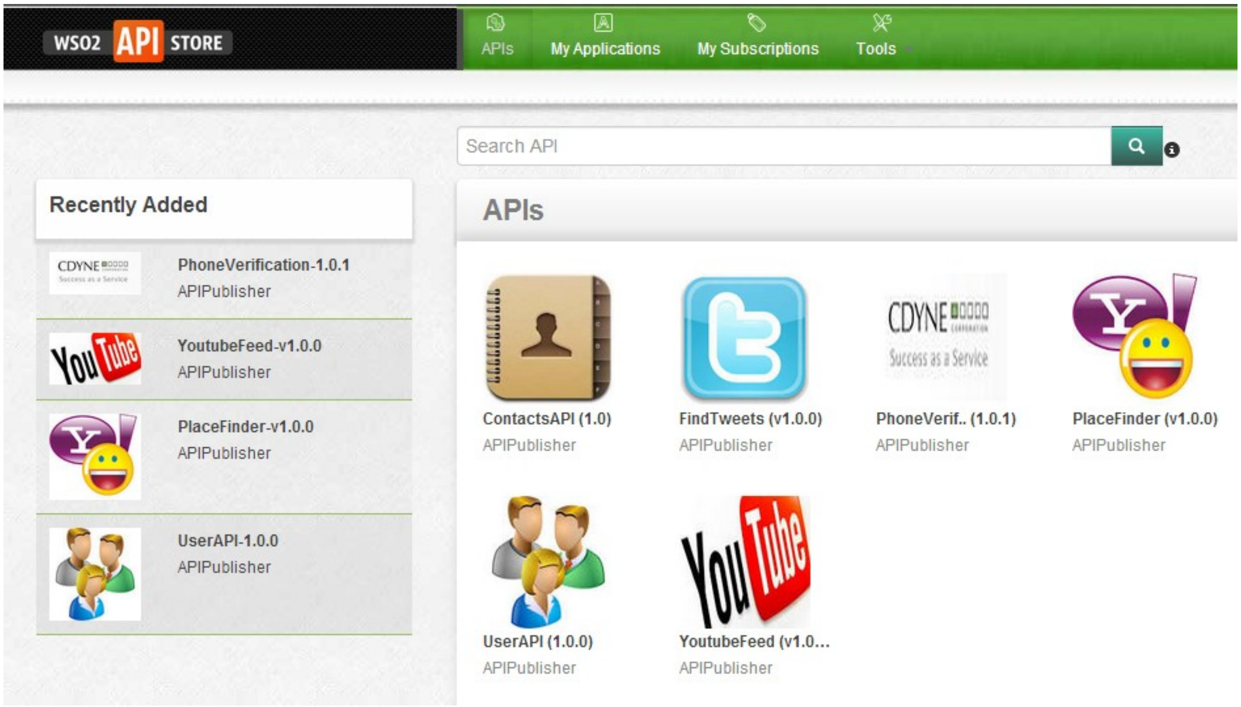
This will duplicate the entire contents of the API information, including the documentation. Once the new version has been created, you can publish it as described in “Publishing the API” and choose the Deprecate Versions options to automatically deprecate version 1.0.0.

Using the API store

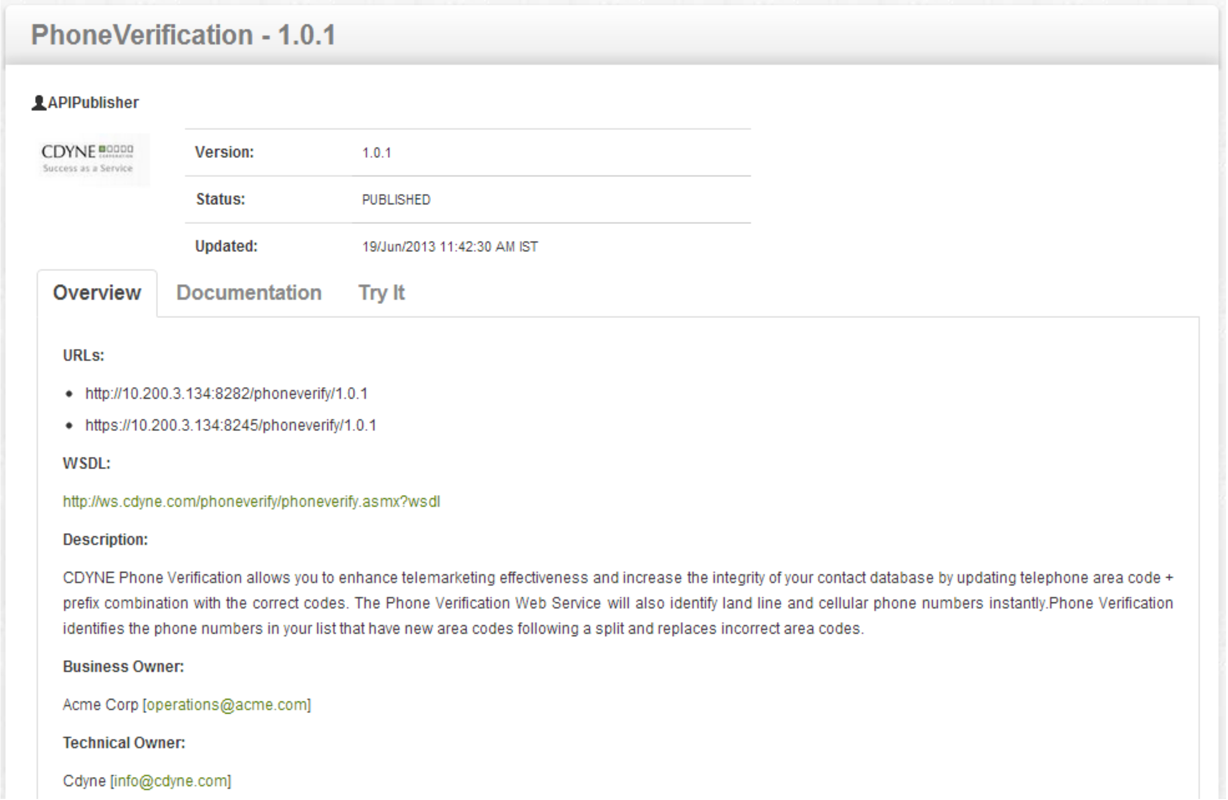
Now that we have successfully published the API, we can open the API store and check its contents.

Browsing the store

To view the API store contents, open the following URL : https://<YourHostName>:9443/store.



In the above API store, you can see the PhoneVerification API, at version 1.0.1. If you click on the icon, you can see the details entered by the API creator:



Version:	1.0.1
Status:	PUBLISHED
Updated:	19/Jun/2013 11:42:30 AM IST

URLs:

- http://10.200.3.134:8282/phoneverify/1.0.1
- https://10.200.3.134:8245/phoneverify/1.0.1

WSDL:

http://ws.cdyne.com/phoneverify/phoneverify.asmx?wsdl

Description:

CDYNE Phone Verification allows you to enhance telemarketing effectiveness and increase the integrity of your contact database by updating telephone area code + prefix combination with the correct codes. The Phone Verification Web Service will also identify land line and cellular phone numbers instantly.Phone Verification identifies the phone numbers in your list that have new area codes following a split and replaces incorrect area codes.

Business Owner:

Acme Corp [operations@acme.com]

Technical Owner:

Cdyne [info@cdyne.com]

You can browse the API store, check the documentation without the necessity to provide credentials. You can search API by their name and as advanced options you can search API by its version,context and provider .Additionally you can search APIs by clicking on the tags to the right.

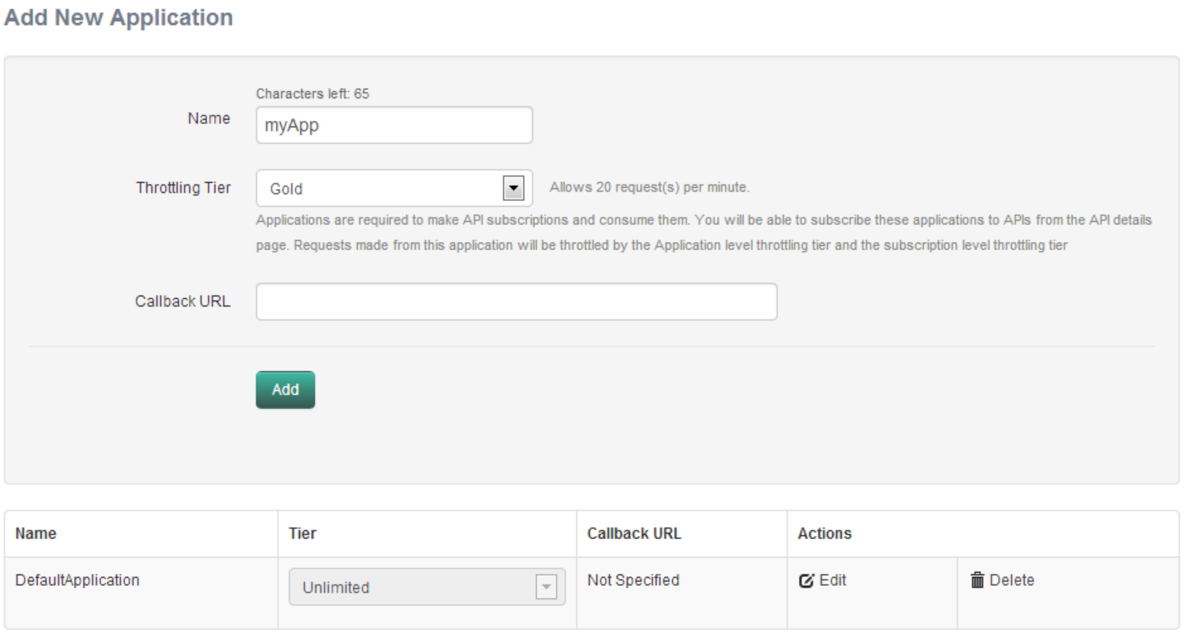
You can consult the documentation from the Documentation tab.

Add Applications

You can add a new application and allocate a throttling level to the application by following below steps:

1. Login in the store using a user in the subscriber role. If you don't have any, you can self-sign from the same page.

2. Click My Applications menu which lists down existing applications including default application and a form which allows to create a new application.
3. Enter a name for the application and select the throttling level.
4. Callback URL is optional and it is used for Authorization Code grant type of Oauth 2.0 with the exposed /authorize endpoint from APIManager.
5. Click Add



Add New Application

Characters left: 65

Name: myApp

Throttling Tier: Gold ▾ Allows 20 request(s) per minute.

Applications are required to make API subscriptions and consume them. You will be able to subscribe these applications to APIs from the API details page. Requests made from this application will be throttled by the Application level throttling tier and the subscription level throttling tier

Callback URL:

Add

Name	Tier	Callback URL	Actions	
DefaultApplication	Unlimited ▾	Not Specified	Edit	Delete

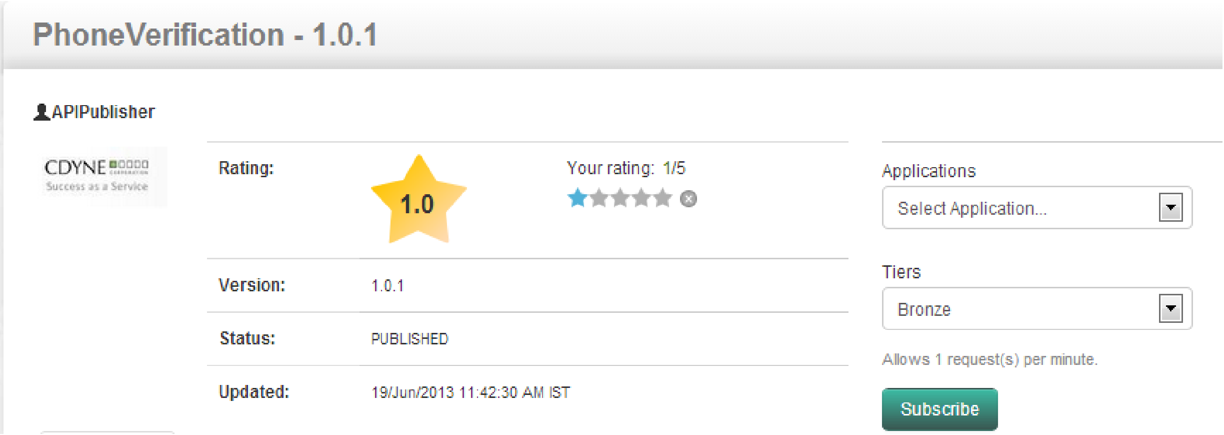
Applications are required to make API subscriptions and consume them. You will be able to subscribe these applications to APIs from the API details page. Requests made from this application will be throttled by the Application level throttling tier and the subscription level throttling tier.

Subscribing to an API

As a consumer, you can subscribe to an API by following those steps:

1. Login in the store using a user in the subscriber role. If you don’t have any, you can self-sign from the same page.

2. You now see additional information for the API and can add ratings and provide comments.



PhoneVerification - 1.0.1

APIPublisher

Rating: 1.0

Your rating: 1/5

Version: 1.0.1

Status: PUBLISHED

Updated: 19/Jun/2013 11:42:30 AM IST

Applications: Select Application...

Tiers: Bronze

Allows 1 request(s) per minute.

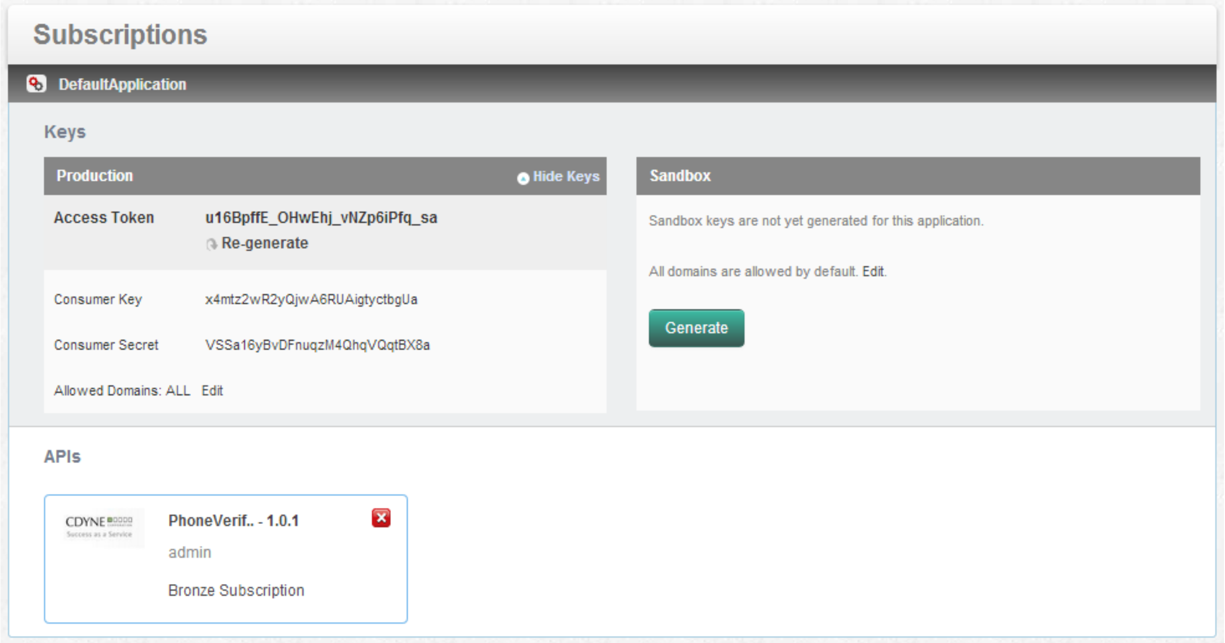
3. Choose an application from the drop-down list : you can use the default one (DefaultApplication) or create a new one right from the drop-down choice.

4. Select the tiers (Service Level) you need - The description of the service level is shown below the Tiers field.

5. Click Subscribe.

6. Once the subscription is successful , you can switch to My Subscriptions .

7. From the MySubscriptions page, you can manage the API keys (at application level) : click Generate to generate the OAuth access token, consumer key and consumer secret, then Show keys to view the generated string. You have now successfully subscribed to the API and can start using it.



Subscriptions

DefaultApplication

Keys

Production

Access Token u16BpffE_OHwEhj_vNZp6iPfq_sa

Consumer Key x4mtz2wR2yQjwA6RUAigtyctbgUa

Consumer Secret VSSa16yBvDFnuqzM4QhqVQqtBX8a

Allowed Domains: ALL Edit

Sandbox

Sandbox keys are not yet generated for this application.

All domains are allowed by default. Edit.

APIs

PhoneVerif.. - 1.0.1

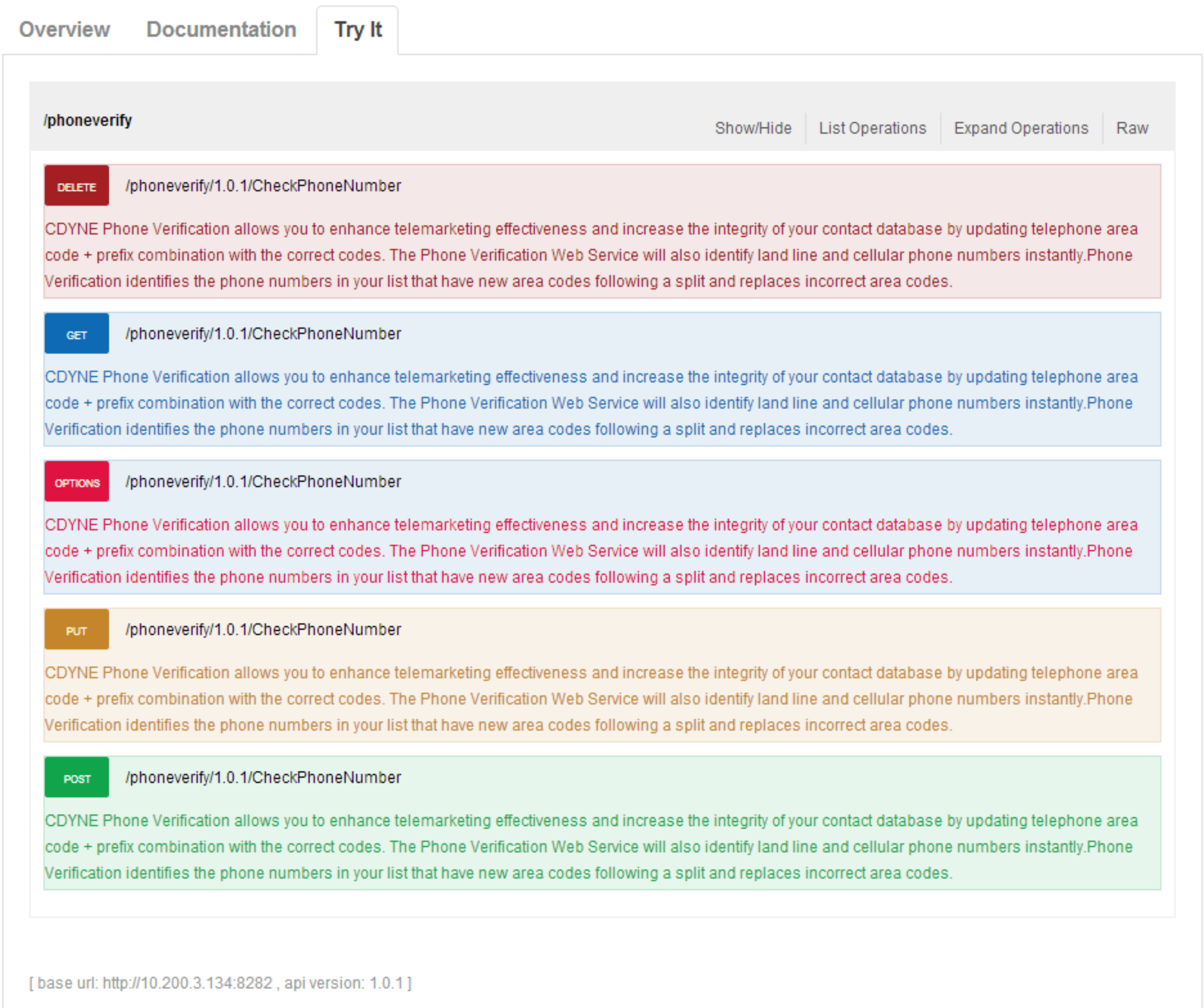
admin

Bronze Subscription

Calling an API

To test the API, you can use the integrated Swagger interactive documentation support (or any other simple REST client application or curl).

Click on the PhoneVerification API from My Subscriptions and click on the 'Try It' tab. You can find the interactive documentation for PhoneVerification API.



/phoneverify

DELETE /phoneverify/1.0.1/CheckPhoneNumber

CDYNE Phone Verification allows you to enhance telemarketing effectiveness and increase the integrity of your contact database by updating telephone area code + prefix combination with the correct codes. The Phone Verification Web Service will also identify land line and cellular phone numbers instantly.Phone Verification identifies the phone numbers in your list that have new area codes following a split and replaces incorrect area codes.

GET /phoneverify/1.0.1/CheckPhoneNumber

CDYNE Phone Verification allows you to enhance telemarketing effectiveness and increase the integrity of your contact database by updating telephone area code + prefix combination with the correct codes. The Phone Verification Web Service will also identify land line and cellular phone numbers instantly.Phone Verification identifies the phone numbers in your list that have new area codes following a split and replaces incorrect area codes.

OPTIONS /phoneverify/1.0.1/CheckPhoneNumber

CDYNE Phone Verification allows you to enhance telemarketing effectiveness and increase the integrity of your contact database by updating telephone area code + prefix combination with the correct codes. The Phone Verification Web Service will also identify land line and cellular phone numbers instantly.Phone Verification identifies the phone numbers in your list that have new area codes following a split and replaces incorrect area codes.

PUT /phoneverify/1.0.1/CheckPhoneNumber

CDYNE Phone Verification allows you to enhance telemarketing effectiveness and increase the integrity of your contact database by updating telephone area code + prefix combination with the correct codes. The Phone Verification Web Service will also identify land line and cellular phone numbers instantly.Phone Verification identifies the phone numbers in your list that have new area codes following a split and replaces incorrect area codes.

POST /phoneverify/1.0.1/CheckPhoneNumber

CDYNE Phone Verification allows you to enhance telemarketing effectiveness and increase the integrity of your contact database by updating telephone area code + prefix combination with the correct codes. The Phone Verification Web Service will also identify land line and cellular phone numbers instantly.Phone Verification identifies the phone numbers in your list that have new area codes following a split and replaces incorrect area codes.

[base url: http://10.200.3.134:8282 , api version: 1.0.1]

The PhoneVerification API takes two parameters: the phone number and a license key, which is set to 0 for testing purposes.

The API URL is the following:

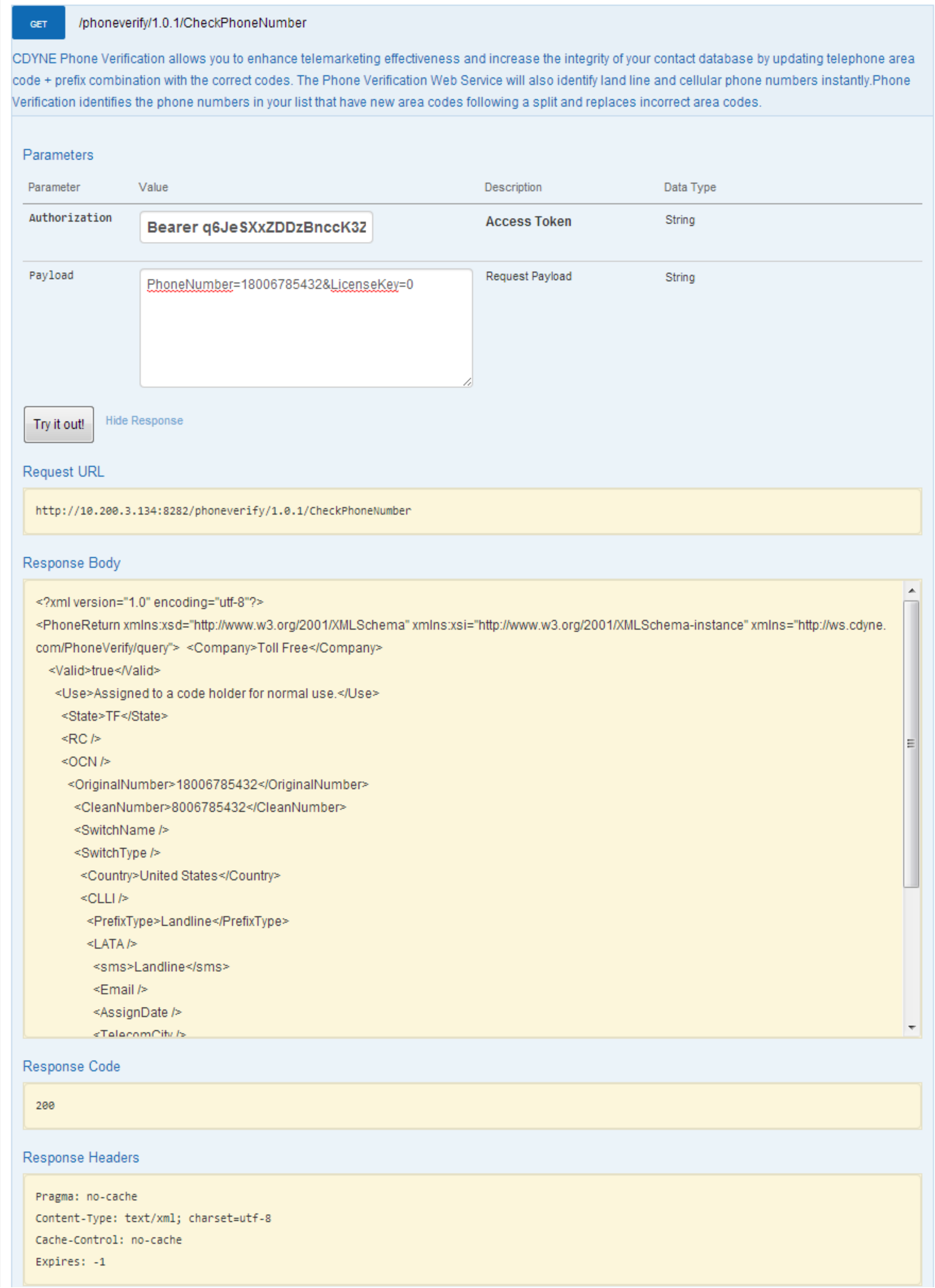
http://host:8280/phoneverify/1.0.1/CheckPhoneNumber

The API Payload is;

PhoneNumber=18006785432&LicenseKey=0

where /phoneverify is the context and 1.0.1 the version. The rest of the URL is driven by the back end service requirements.

We also need to pass the API key : this is done use the Authorization header with a value of : Bearer <access token> , so in the case above Bearer **q6JeSXxZDDzBnccK3ZZGf5_AZTka**



/phoneverify/1.0.1/CheckPhoneNumber

CDYNE Phone Verification allows you to enhance telemarketing effectiveness and increase the integrity of your contact database by updating telephone area code + prefix combination with the correct codes. The Phone Verification Web Service will also identify land line and cellular phone numbers instantly.Phone Verification identifies the phone numbers in your list that have new area codes following a split and replaces incorrect area codes.

Parameters

Parameter	Value	Description	Data Type
Authorization	Bearer q6JeSXxZDDzBnccK3Z	Access Token	String
Payload	PhoneNumber=18006785432&LicenseKey=0	Request Payload	String

Request URL

```
http://10.200.3.134:8282/phoneverify/1.0.1/CheckPhoneNumber
```

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 />
        <PrefixType>Landline</PrefixType>
        <LATA />
         <sms>Landline</sms>
         <Email />
         <AssignDate />
         <TelecomCity />
```

Response Code

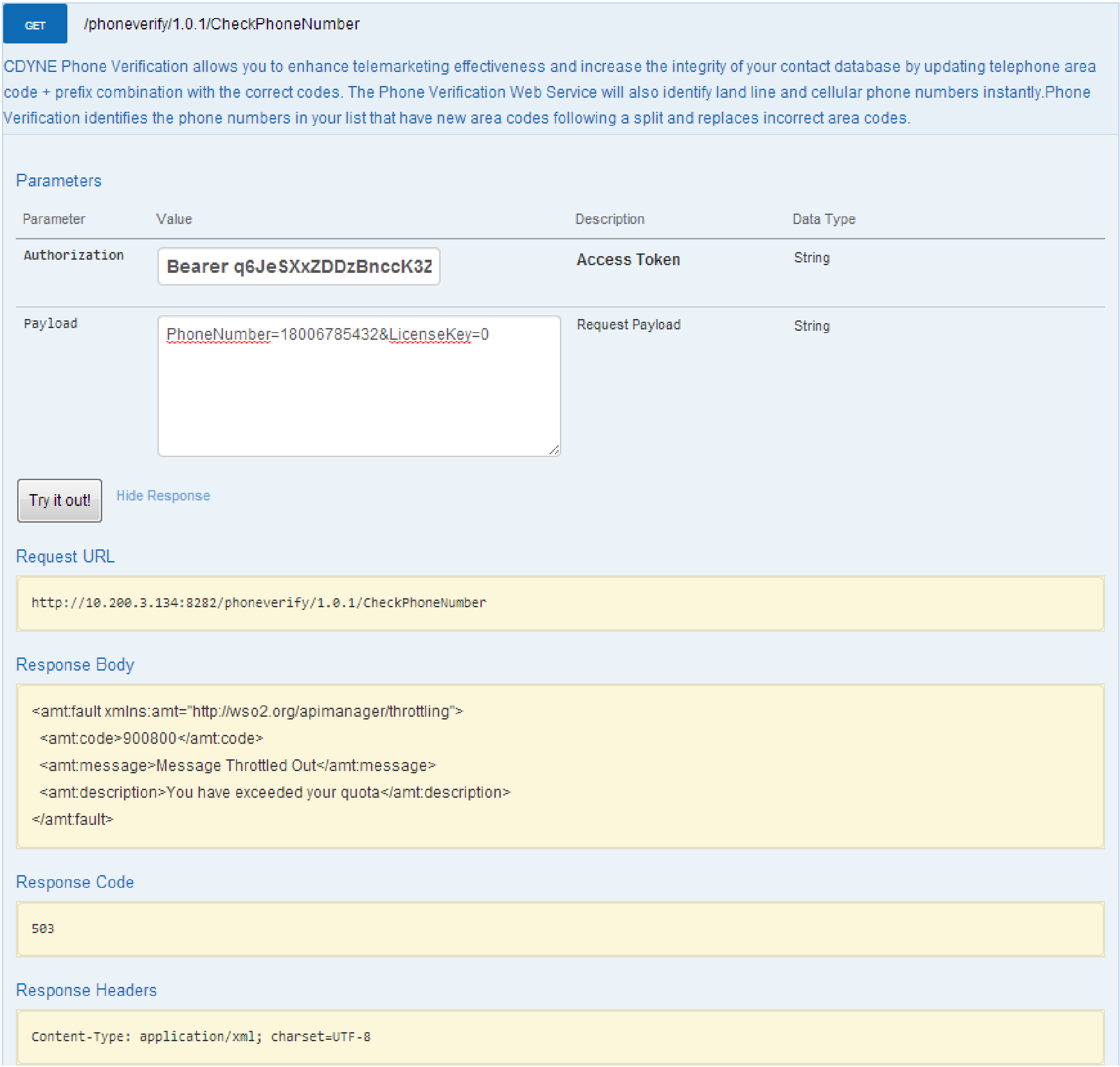
```
200
```

Response Headers

```
Pragma: no-cache
Content-Type: text/xml; charset=utf-8
Cache-Control: no-cache
Expires: -1
```

h

Since we have applied a Bronze tiers, which limits call to 1 per minute, another attempt to call the API results in a throttling error.



/phoneverify/1.0.1/CheckPhoneNumber

CDYNE Phone Verification allows you to enhance telemarketing effectiveness and increase the integrity of your contact database by updating telephone area code + prefix combination with the correct codes. The Phone Verification Web Service will also identify land line and cellular phone numbers instantly.Phone Verification identifies the phone numbers in your list that have new area codes following a split and replaces incorrect area codes.

Parameters

Parameter	Value	Description	Data Type
Authorization	Bearer q6JeSXxZDDzBnccK32	Access Token	String
Payload	PhoneNumber=18006785432&LicenseKey=0	Request Payload	String

Request URL

```
http://10.200.3.134:8282/phoneverify/1.0.1/CheckPhoneNumber
```

Response Body

```
<amt:fault xmlns:amt="http://wso2.org/apimanager/throttling">
  <amt:code>900800</amt:code>
  <amt:message>Message Throttled Out</amt:message>
  <amt:description>You have exceeded your quota</amt:description>
</amt:fault>
```

Response Code

```
503
```

Response Headers

```
Content-Type: application/xml; charset=UTF-8
```

Deleting an API

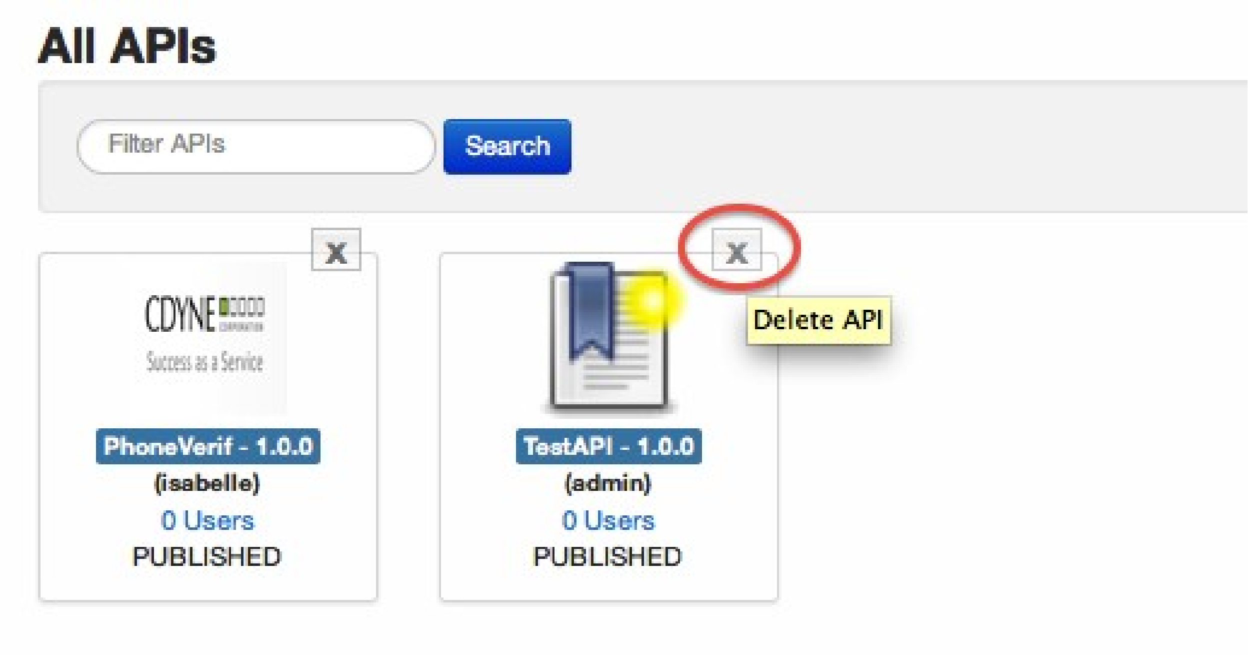
To explicitly delete an API from the store:

1. Open the API publisher web application at https://<YourHostName>:9443/publisher

2. Login using the user in creator role you defined previously (in our case, apicreator)

3. Click the Delete button at the top right of the API icon.

4. Confirm the deletion.



Monitoring and Statistics

The API publisher web application provides several statistical dashboards:

1. Usage of APIs per creator
2. Usage of all APIs
3. Average response times
4. Usage of an API per subscriber
5. Usage of an API per subscriber per version
6. Number of subscriptions per API

Aside from the number of subscriptions per API, all other dashboards require to use WSO2 Business Activity Monitoring 2.3.0 for analytics - You need to use a WSO2 BAM 2.3.0, available for download on our web site. Simply unzip BAM 2.3.0 to install it.

Enabling and configuring statistics

In order to use BAM 2.3.0 with the API manager, you need to:

1. Enable the API tracking option in <apimgr_install_root>/repository/conf/api-manager.xml.

To do so:

Edit this file and locate the APIUsageTracking element and configure below properties.

a) Change <Enabled> node of APIUsageTracking element as true.

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

b) Change <DataSourceName> node of APIUsageTracking element as the JNDI name of the data source to be used for getting BAM statistics.This data source should be defined in the master-datasources.xml file in conf/datasources directory.

```
<DataSourceName>jdbc/WSO2AM_STATS_DB</DataSourceName>
```

2. Configure the database used to store analytical data (the BAM tooling will analyze the data and write it to this database). The <DataSourceName>jdbc/WSO2AM_STATS_DB</DataSourceName> XML node defines the datasource used to fetch analytical data.

This must correspond to the datasource definition available in the <apimgr_install_root>/repository/conf/datasources/master-datasources.xml file. Note that the master-datasources.xml file needs to be edited, so that it points to the BAM_install_root/repository/databases/APIMGT_STATS_DB database (this is where the default analytics scripts write by default).

3. Prepare BAM to create statistics from the API manager. To achieve this:

a) Copy the Statistics/API_Manager_Analytics.tbox file directory into BAM_HOME/repository/deployment/server/bam-toolbox (create this directory if not already there) - The toolbox file describes the information collected, how to analyze the data, as well as the location of the database where the analyzed data is stored.

b) As mentioned in the above step 2, add the same data-source configuration defined in API Manager [which is named as jdbc/WSO2AM_STATS_DB] to the master-datasources file of BAM as well. The location of master-datasources.xml file is

<BAM_install_root>/conf/datasources/master-datasources.xml

c) Change the cassandra port given in JDBC connection url from the below data-source con figuration which can be found in master-datasources.xml file of BAM according to the port offset defined in BAM server. If the port offset is 1, then cassandra port should changed to '9161'.

```
   <datasource>
    <name>WSO2BAM_CASSANDRA_DATASOURCE</name>
    <description>The datasource used for Cassandra data</description>
    <definition type="RDBMS">
        <configuration>
            <url>jdbc:cassandra://localhost:9161/EVENT_KS</url>
            <username>admin</username>
            <password>admin</password>
        </configuration>
    </definition>
 </datasource>
```

c) Change port offset for the BAM product to 1 by editing the repository/conf/carbon.xml file (search for the offset node) - This increments all ports used by the servers by 1, which means the BAM server will now run with port 9444, port that the API manager uses by default.

d) Start WSO2 BAM server from <BAM_HOME>/bin/wso2server.[sh/bat]

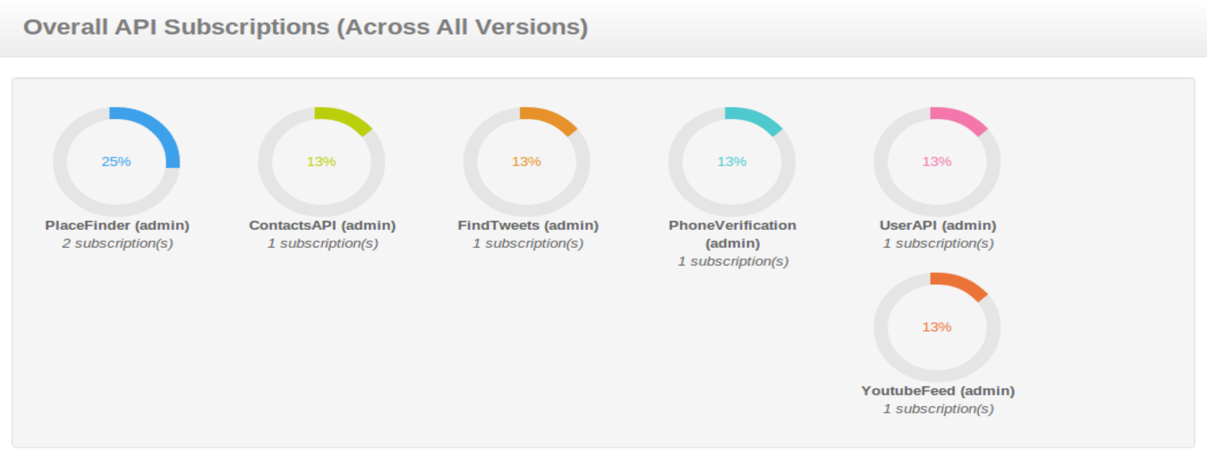
Viewing Statistics

To see statistics, you need to first generate some traffic via the API gateway (calling the Cdyne API above for example) and wait a few instants.

To view statistics:

A) Connect to the API publisher web application as a creator or publisher. In publisher role, you are able to see all stats , as creator you also see specific stats for the APIs you created.

B) Go to APIs > All Statistics



Overall API Subscriptions (Across All Versions)

PlaceFinder (admin)
2 subscription(s)

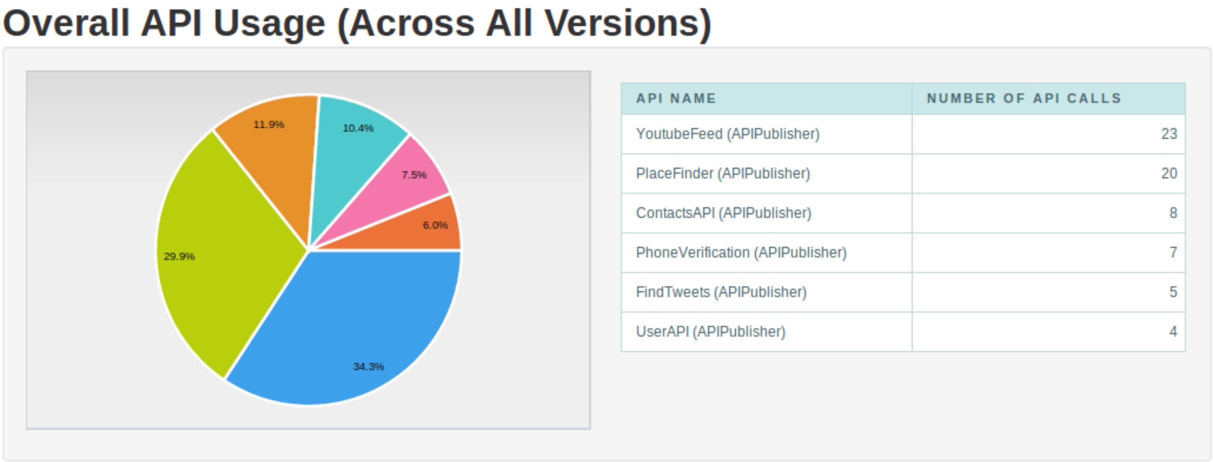
ContactsAPI (admin)
1 subscription(s)

FindTweets (admin)
1 subscription(s)

PhoneVerification (admin)
1 subscription(s)

UserAPI (admin)
1 subscription(s)

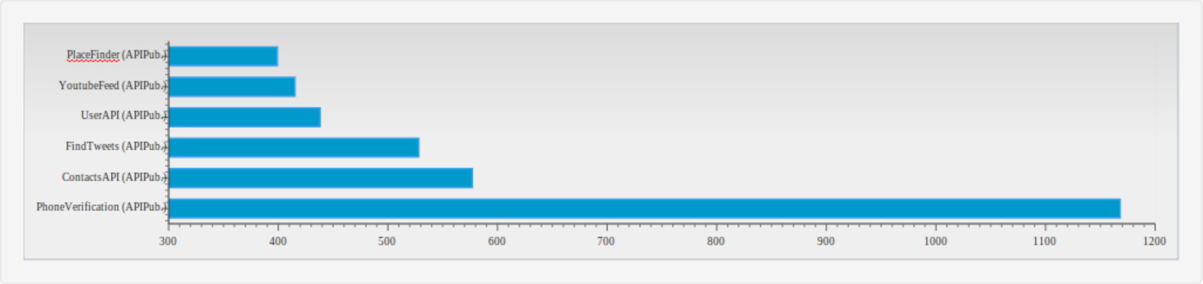
YoutubeFeed (admin)
1 subscription(s)



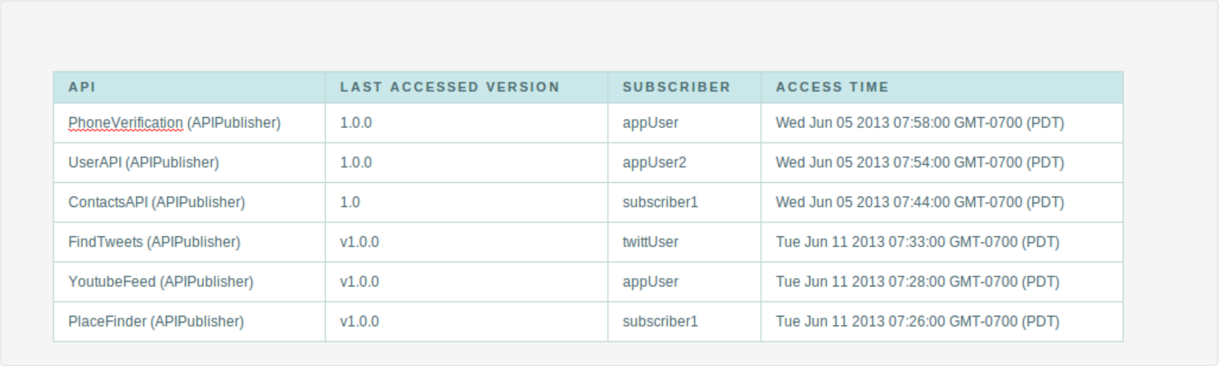
Overall API Usage (Across All Versions)

API NAME	NUMBER OF API CALLS
YoutubeFeed (APIPublisher)	23
PlaceFinder (APIPublisher)	20
ContactsAPI (APIPublisher)	8
PhoneVerification (APIPublisher)	7
FindTweets (APIPublisher)	5
UserAPI (APIPublisher)	4

API Response Times (Across All Versions / Last 10 invocations)

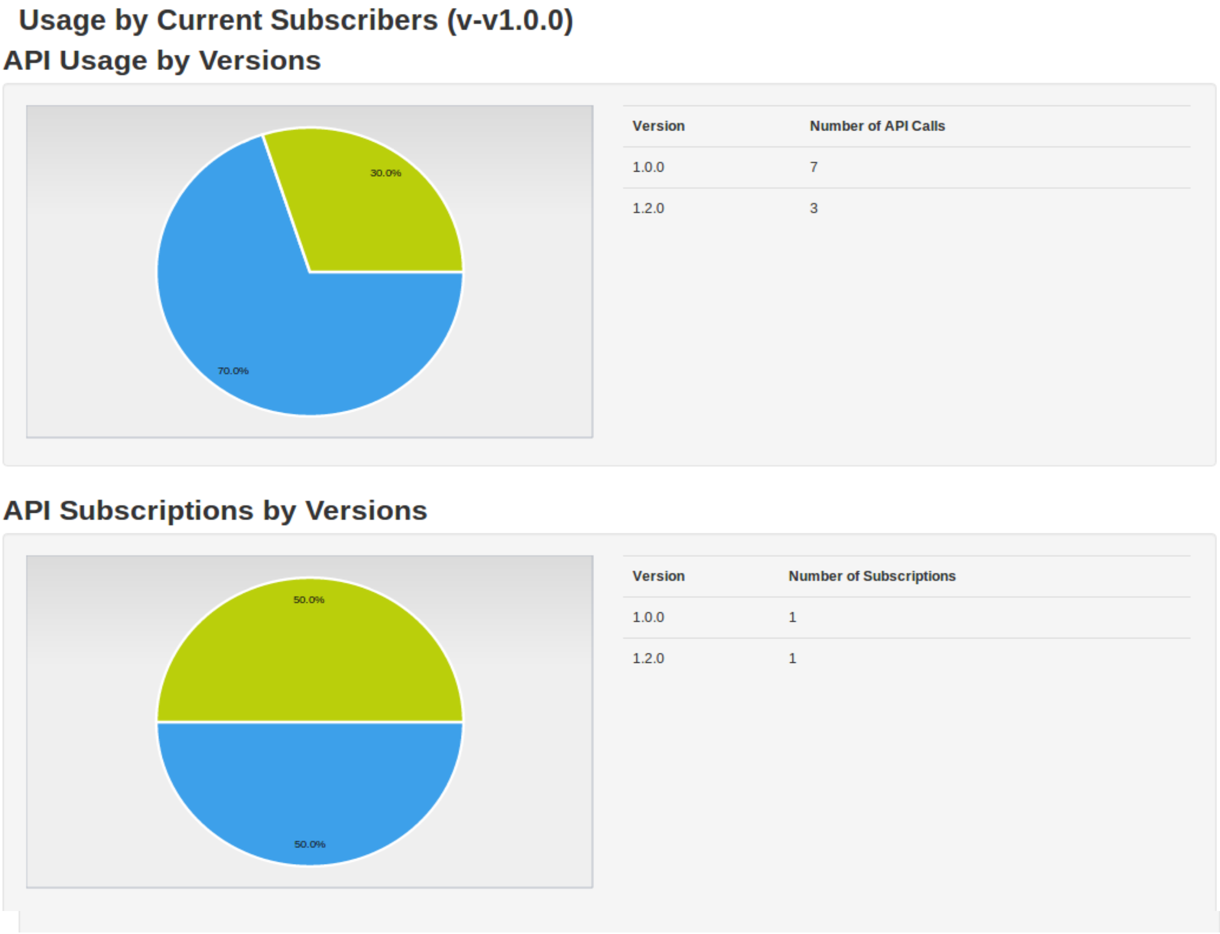


API Last Access Times (Across All Versions / Last 10 invocations)



API	LAST ACCESSED VERSION	SUBSCRIBER	ACCESS TIME
PhoneVerification (APIPublisher)	1.0.0	appUser	Wed Jun 05 2013 07:58:00 GMT-0700 (PDT)
UserAPI (APIPublisher)	1.0.0	appUser2	Wed Jun 05 2013 07:54:00 GMT-0700 (PDT)
ContactsAPI (APIPublisher)	1.0	subscriber1	Wed Jun 05 2013 07:44:00 GMT-0700 (PDT)
FindTweets (APIPublisher)	v1.0.0	twittUser	Tue Jun 11 2013 07:33:00 GMT-0700 (PDT)
YoutubeFeed (APIPublisher)	v1.0.0	appUser	Tue Jun 11 2013 07:28:00 GMT-0700 (PDT)
PlaceFinder (APIPublisher)	v1.0.0	subscriber1	Tue Jun 11 2013 07:26:00 GMT-0700 (PDT)

If you select a particular API, you can see additional statistics by version and by subscriber.



Usage by Current Subscribers (v-v1.0.0)

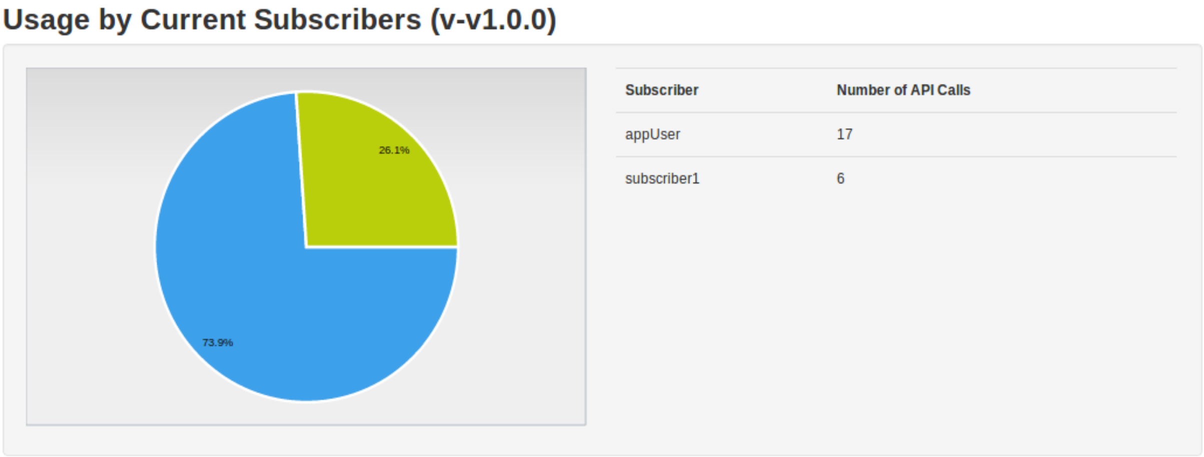
API Usage by Versions

Version	Number of API Calls
1.0.0	7
1.2.0	3

API Subscriptions by Versions

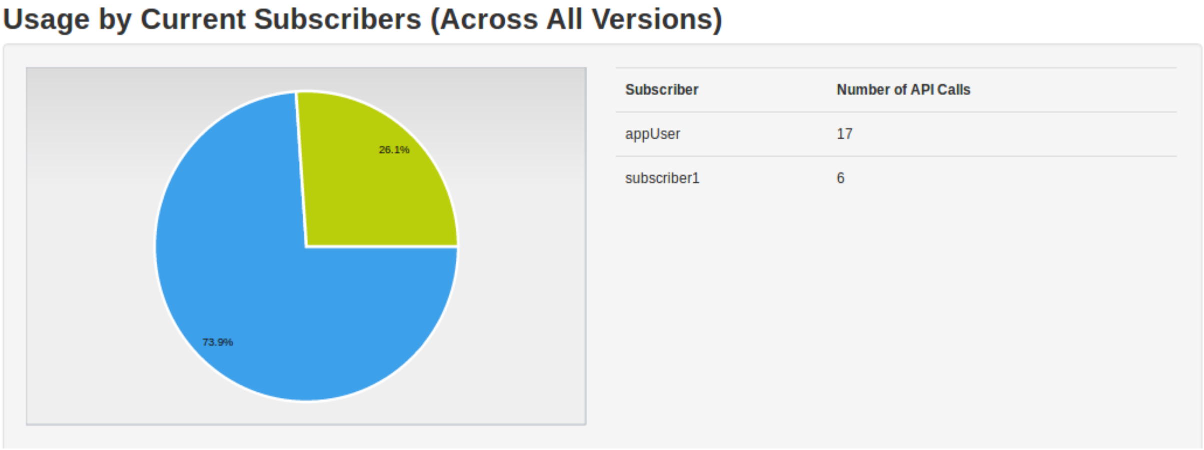
Version	Number of Subscriptions
1.0.0	1
1.2.0	1

Finally, you can also see specific statistics for API subscribers from the Users tab.



Usage by Current Subscribers (v-v1.0.0)

Subscriber	Number of API Calls
appUser	17
subscriber1	6



Usage by Current Subscribers (Across All Versions)

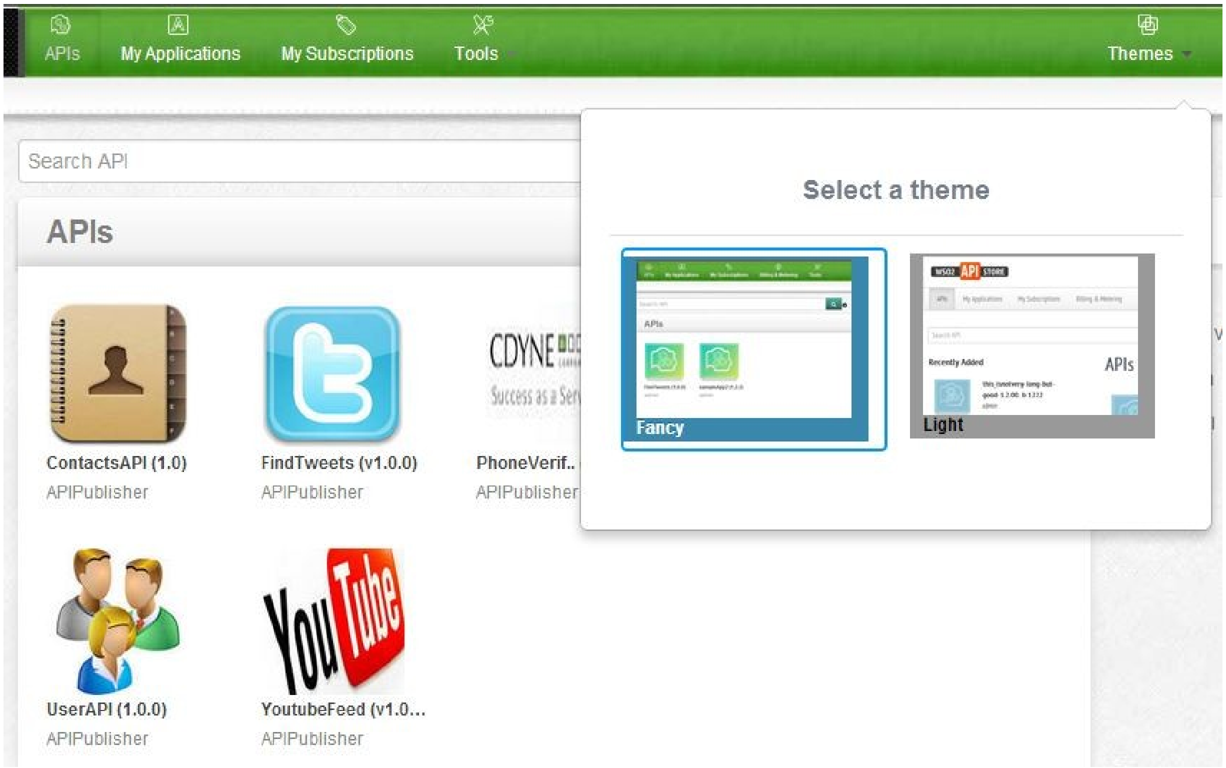
Subscriber	Number of API Calls
appUser	17
subscriber1	6

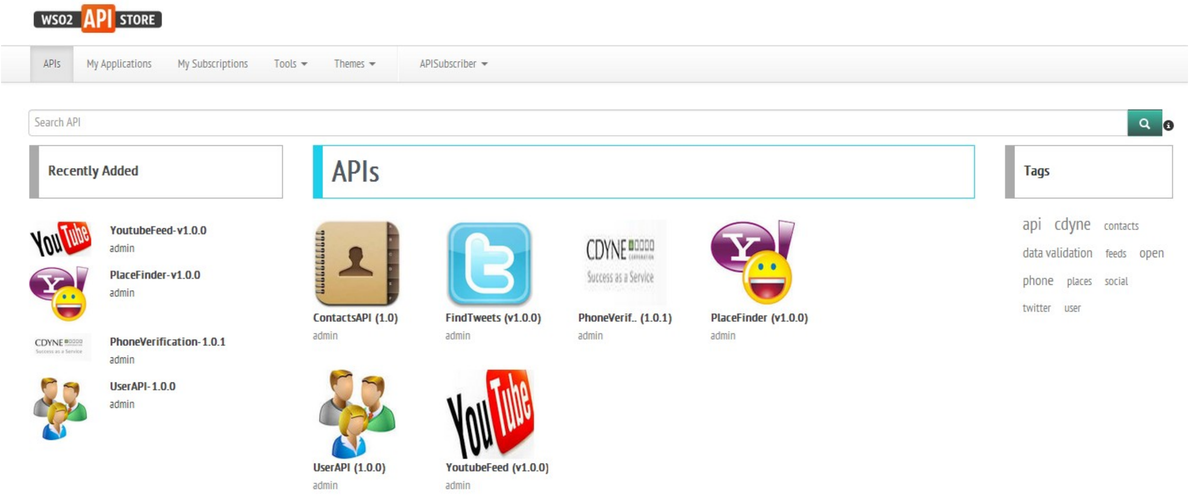
API Store Themes

API store theme can be changed by following given steps:

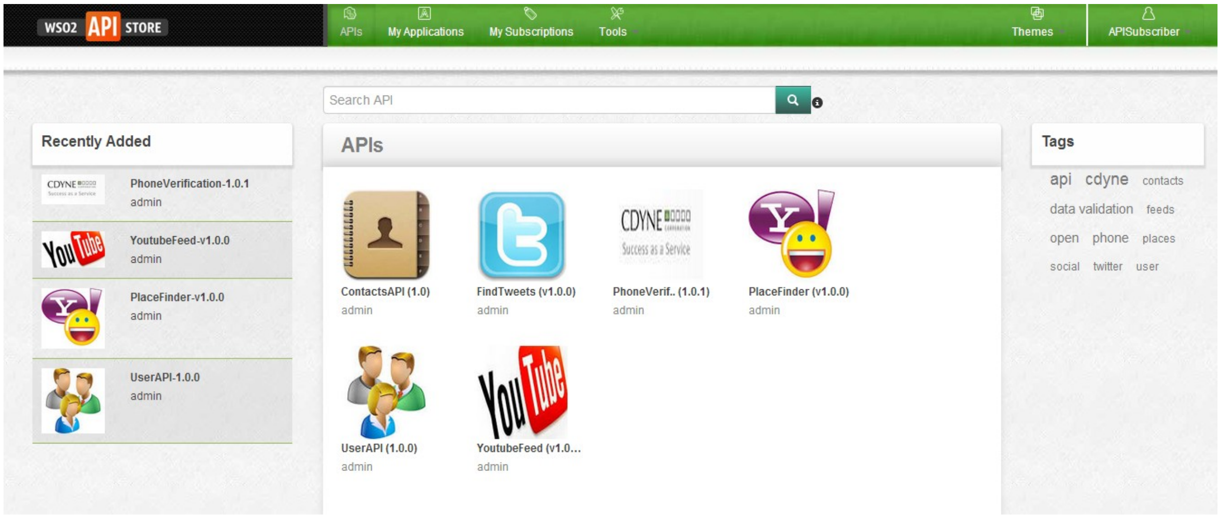
1. Login in the store using a user in the subscriber role. If you don't have any, you can self-sign from the same page.

2. Click on the top right settings icon and select a Theme from available Themes.





Light Theme



Fancy Theme

Installing the API Manager on Amazon EC2

You can install the API manager on an instance hosted on Amazon EC2. In order to force the various components of the product to use the instance public DNS address, you need to make the following changes to the API Manager installation. Following instructions are for Linux OS.

A) You need to add the following instructions to the .bashrc (or equivalent) of the user which will execute the API manager process (we recommend you use a specific user and never run as root, for security purposes) - The goal is to retrieve the public DNS address and set it as an environment variable (amazon.pub.hostname) which can be used later.

```
# set AMAZON Hostname for WSO2 AM

pubname=$(curl http://169.254.169.254/latest/meta-data/public-hostname 2>/dev/null);

export JAVA_OPTS=-Damazon.pub.hostname=$pubname

export JAVA_HOME=/usr/lib/jvm

export PATH=$PATH:$JAVA_HOME/bin
```

B) You need to edit the <install_root>/repository/conf/carbon.xml file and update the following 2 lines (which are by default commented out) .

```
<HostName>${amazon.pub.hostname}</HostName>

<MgtHostName>${amazon.pub.hostname}</MgtHostName>
```

C) You need to edit the <install_root>/repository/conf/api-manager.xml file and update the following entries, replacing certain occurrences of ${carbon.local.ip} by the Amazon public DNS address we defined in step 1. Just replace the occurrences listed below!

```
 <AuthManager>

        <!--Server URL of the Authentication service -->

<ServerURL>https://${amazon.pub.hostname}:$
{mgt.transport.https.port}/services/</ServerURL>

....

<APIGateway>

        <!--Server URL of the API gateway.-->
        <ServerURL>https://${amazon.pub.hostname}:${mgt.transport.https.port}/services/</ServerURL>

...

<APIEndpointURL>http://${amazon.pub.hostname}:${http.nio.port},https://${amazon.pub.hostname}:${https.nio.port}</APIEndpointURL>

...

<APIKeyManager>

<!--Server URL of the API key manager-->

<ServerURL>https://${amazon.pub.hostname}:${mgt.transport.https.port}/services/</ServerURL>
```

Advanced Topics

Changing API Store Branding

Please read the following blog entry for more details:
http://wso2.org/library/tutorials/2012/09/customizing-api-store-publisher-part1

http://wso2.org/library/articles/2012/06/api-store-themes

Adding a Throttling Tiers

Please read the following blog entry for more details:

http://sumedha.blogspot.fr/2012/06/how-to-add-new-throttling-tier-to-wso2.html

Interactive Documentation Support with Swagger Integration

Please read the following blog entry and documentation for more details:

http://docs.wso2.org/wiki/display/AM140/Adding+Documentation+Using+Swagger

http://blog.cobia.net/cobiacomm/2013/05/31/swagger-with-wso2-api-manager/

Google Analytics Tracking Support

Please read the following blog entry and documentation for more details:

http://docs.wso2.org/wiki/display/AM140/Google+Analytics+Tracking http://blog.lakmali.com/2012/12/google-analytics-tracking-for-wso2-api.html

Internationalization and Localization

Please read the following documentation for more details: http://docs.wso2.org/wiki/display/AM140/Internationalization+and+Localization

Additional Configuration Option

Additional configuration options are documented here:

http://sumedha.blogspot.fr/2012/06/guide-to-advance-configuration-options.html

Appendix

Installing the Samples

The API manager comes with a certain number of samples, including APIs from Twitter, YouTube and Yahoo. Samples are located under <apimgr_install_root>/samples. Each sample comes with an APIPopulator script which drive the API manager via a REST API.

By default, the samples installation uses the admin user to create samples.

Configuring Libraries

You need to configure various libraries before installing examples. To do this, you must run ant inside the <apimgr_install_root>/bin directory.

Installing the samples

To install the samples as the admin user:

1. Make sure the API manager is started
2. Execute the APIPopulator.sh script for each sample.

Starting and Stopping the API gateway

To start the API manager product once you unzipped it, you only need to make sure you have a Java runtime at 1.6 or 1.7 level installed on your machine and the JAVA_HOME environment variable set. Then:

1. Open a command line window
2. Go to <install_root>/bin
3. Start wso2server.bat or wso2server.sh
4. Wait until you see the message:

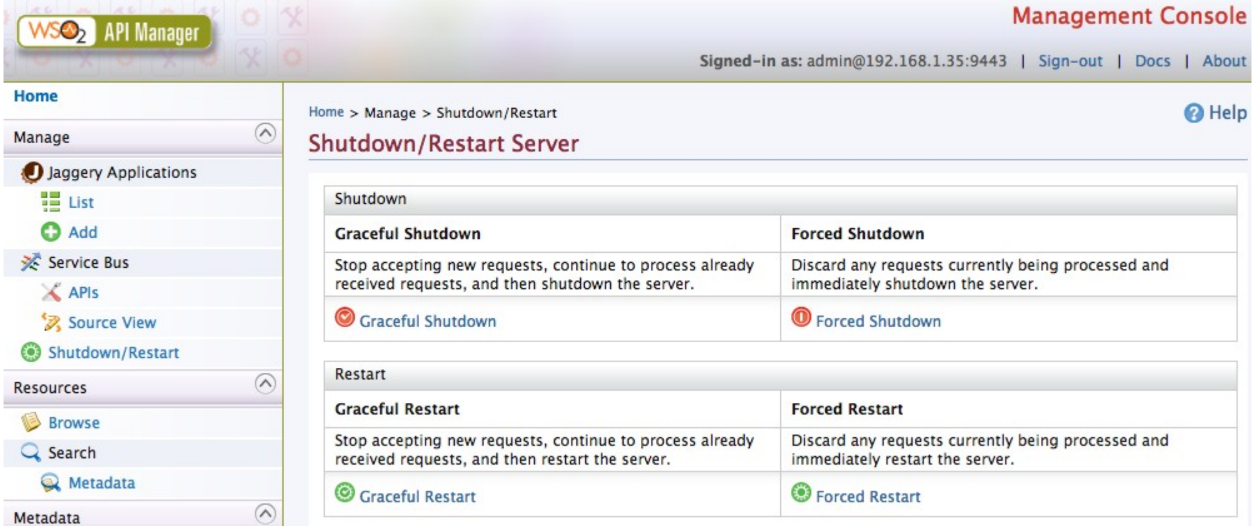
```
[2012-10-30 10:13:24,933]  INFO - StartupFinalizerServiceComponent Server           :  WSO2
    API Manager-1.1.0

[2012-10-30 10:13:24,933]  INFO - StartupFinalizerServiceComponent WSO2 Carbon started in
    63 sec

[2012-10-30  10:13:25,408]   INFO  -  CarbonUIServiceComponent  Mgt  Console  URL   :
    https://192.168.1.38:9443/carbon/
```

5. If you need further installation help, check this video: http://www.youtube.com/watch?v=mNQlrpMsAbE&feature=player_embedded

To stop the API gateway, simply hit Ctrl-C in the command window or choose Shutdown/Restart from the API manager administration console (in the Manage section).



Shutdown/Restart Server

Shutdown	
Graceful Shutdown	**Forced Shutdown**
Stop accepting new requests, continue to process already received requests, and then shutdown the server.	Discard any requests currently being processed and immediately shutdown the server.
Graceful Shutdown	Forced Shutdown

Restart	
Graceful Restart	**Forced Restart**
Stop accepting new requests, continue to process already received requests, and then restart the server.	Discard any requests currently being processed and immediately restart the server.
Graceful Restart	Forced Restart