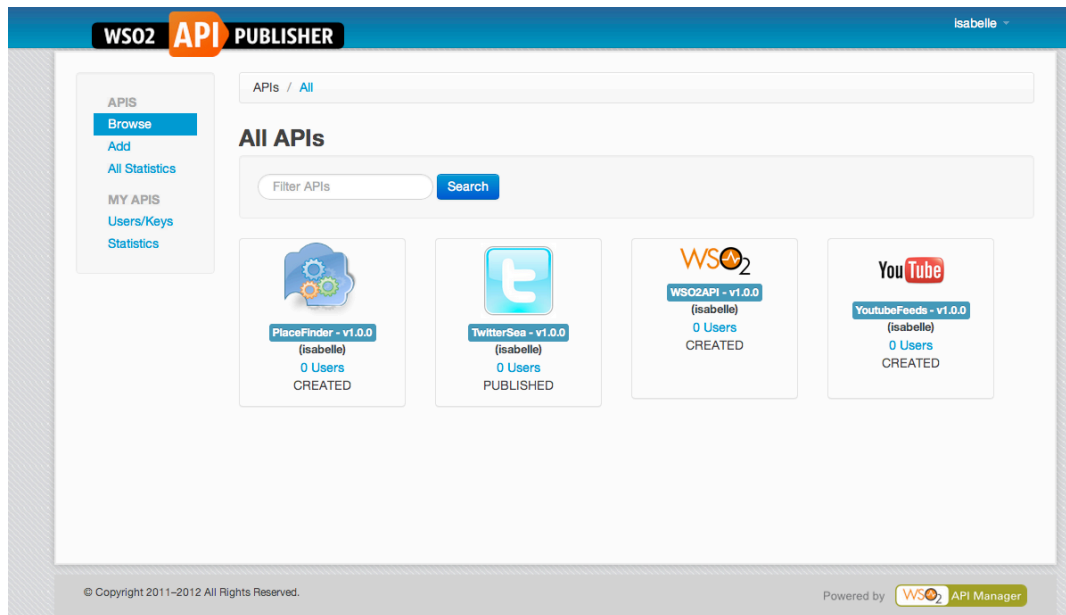


WSO2 API MANAGER 1.1

GETTING STARTED GUIDE



Introduction and Concepts	3
Components	3
Users and Roles	4
API Lifecycle	5
Throttling Tiers	6
API Keys	7
Defining Users and Roles	9
Defining roles	9
Defining users via the admin console	11
Defining users via self-registration	11
Publishing APIs	13
The Phone Number validation API	13
Adding an API to the Store	13
Adding Documentation	15
Publishing the API	17
API Versioning	18
Using the API store	20
Browsing the store	20
Subscribing to an API	21
Calling an API	23
Deleting an API	24
Monitoring and Statistics	25
Enabling and configuring statistics	25
Viewing Statistics	26
Installing the API Manager on Amazon EC2	31
Advanced Topics	32
Changing API Store Branding	32
Adding a Throttling Tiers	32
Additional Configuration Option	32
Appendix	33

Installing the Samples	33
Starting and Stopping the API gateway	33

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.

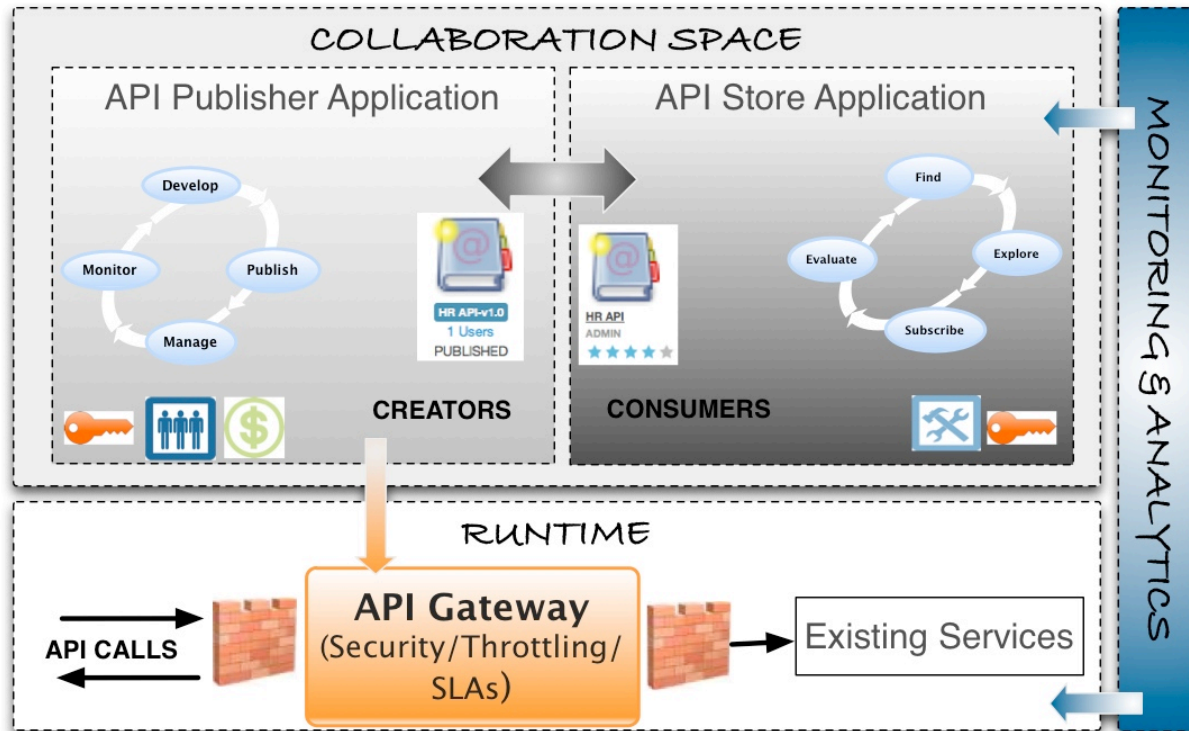
In addition, as it is powered by the WSO2 Business Activity Monitor, the WSO2 API Manager is ready for massively scalable deployment immediately.

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.

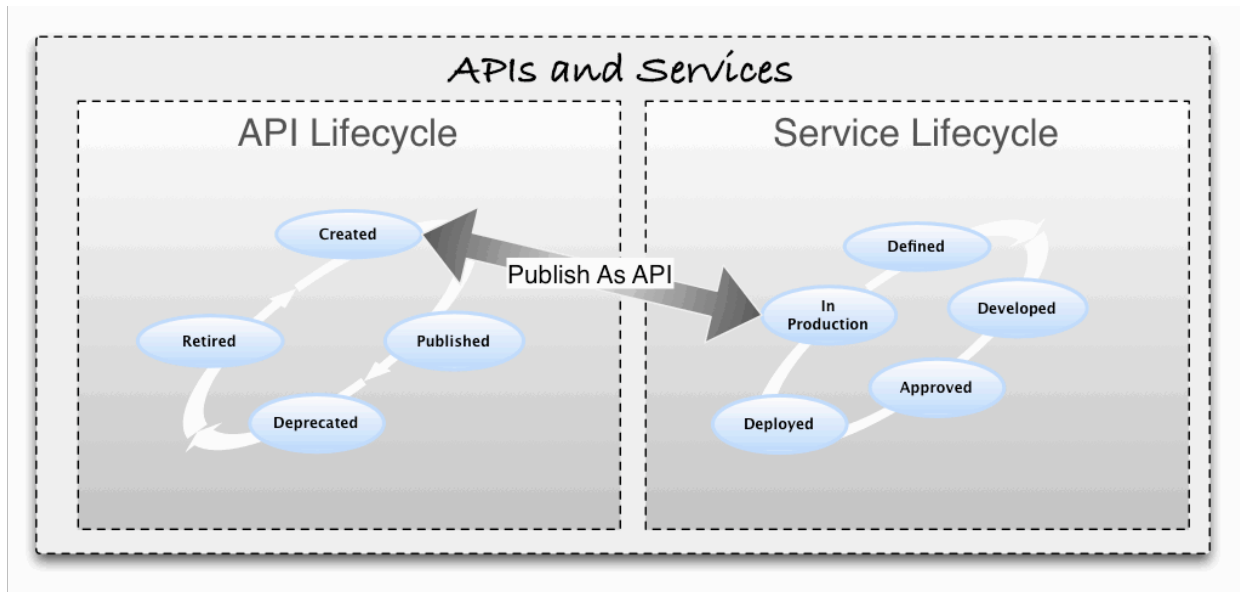
- **Consumer:** the consumer uses the API store to discover APIs, consult the documentation and forums as well as rate/comments on the API. He/she subscribes to APIs to obtain an API key.

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>

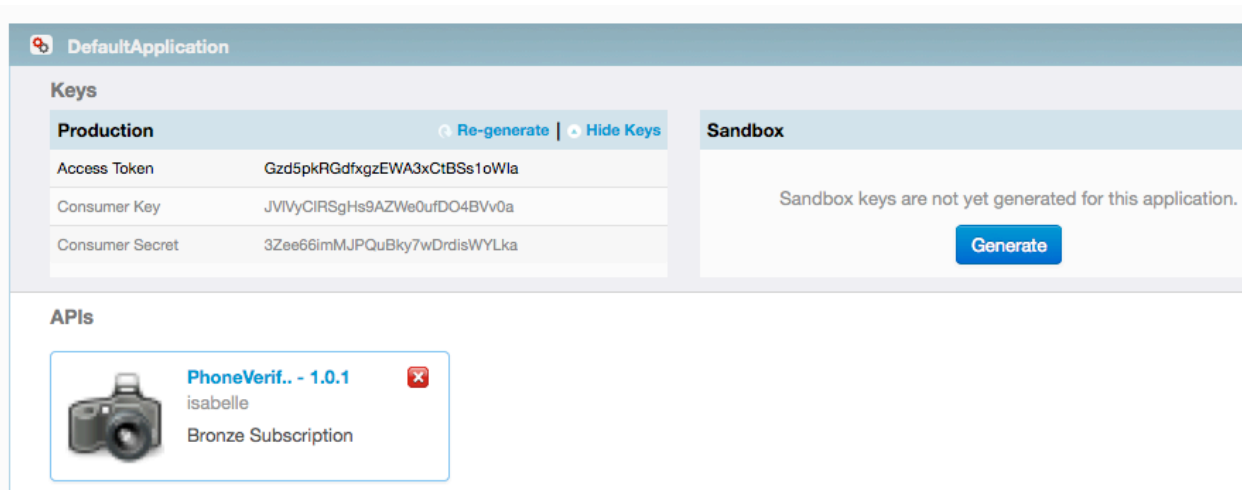
<http://wso2.com>

API Manager | Getting Started Guide

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).



Application Access Token

API keys 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.

API keys are generated at the application level and valid for all APIs which are associated to this application. All 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 <AccessTokenDefaultValidityPeriod>.

Application User Access Token

We also support the generation of access tokens on demand through a specific login API - This API takes 4 parameters:

- Userid
- Password
- Consumer Key
- Consumer Secret

The Login API returns an access token, which can then be stored in session on the client side (the application does not need to manage users and passwords). On the API gateway, the access token is validated for each API call. In case the token expires, the application will have to obtain a new token by re-issuing the login call again.

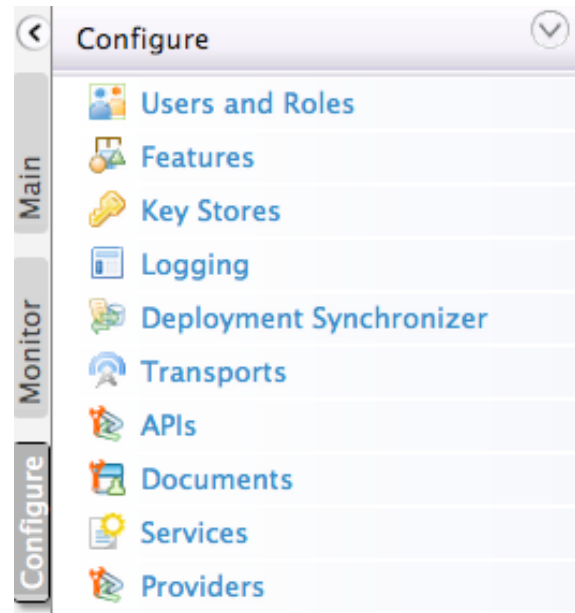
Defining Users and Roles

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.

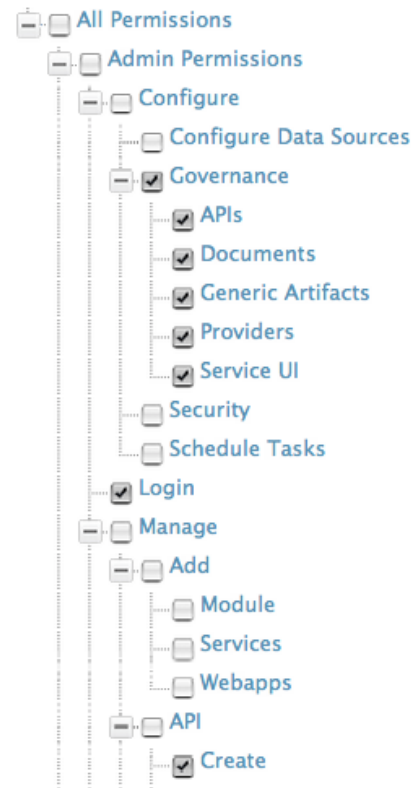
Defining roles

To define the *creator* role, you need to:

1. 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.
2. Select the configure tab on the left side
3. Select Users and Roles
4. Select Roles
5. Click **Add New Role**
6. Provide **creator** as the role name



7. Click **Next** - You will be presented with a list of permissions. For the creator role, you need to select the following permissions:
 - Configure > Governance and all underlying permissions
 - Login
 - Manage > API > Create
 - Manage > Resources > Govern and all underlying permissions.
8. Click **Finish** (at the bottom of the page)
9. Repeat steps 7 and 8 to create the *publisher* role, with the following permissions:
 - Login
 - 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
 - Login
 - 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:

1. Go to **Configure > Users and Roles**
2. Click **Users**
3. Click **Add New User**
4. Provide user name and password
5. Click **Next**
6. Select the role you want to assign to the user (creator, publisher or subscriber)
7. Click **Finish**.
8. Repeat those steps to create a user in the publisher and subscriber roles.

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

Add User

Step 1 : Enter user name

Enter user name

User Name*

Password*

Password Repeat*

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 userid and password. The password must follow certain validity rules, as per the image below:

Sign - up for a new account

Username:

Password:

For a more secure password:

- Use 6 to 32 characters
- Use both letters and numbers
- Add special characters (such as @, ?, %)
- Mix capital and lowercase letters

Re-type Password:

Medium

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.

The screenshot shows the 'Add New API' form in the API Manager web application. The left sidebar contains navigation links: APIS, Browse, Add (highlighted), All Statistics, MY APIS, Users/Keys, and Statistics. The main form area is titled 'Add New API' and contains the following fields:

- Name:** PhoneVerification
- Context:** /phoneverify
- Version:** 1.0.0
- Description:** The Phone Verification API allows you to enhance telemarketing effectiveness and increase the integrity of your contact database
- Thumbnail Image:** Choose File | tel-logo.jpeg
 - Max Size 1 MB.
 - Recommended Image size: 100 x 100 pixels.
- Endpoint URL:** http://ws.cdyne.com/phoneverify/phoneverify. Test URI

Below the Endpoint URL field, there is an example: 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
Thumbnail Image	Image file	Icon to be displayed in API store (can be jpeg, tiff, png format)
Endpoint 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)
Tags	String	One or 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:			
URL Prefix	URL Patterns	Allowed HTTP Verbs	
/phoneverify/1.0.0	<input type="text"/>	<input type="checkbox"/> GET <input type="checkbox"/> PUT <input type="checkbox"/> POST <input type="checkbox"/> DELETE	<button>+ Add</button>
/phoneverify/1.0.0	/	<input checked="" type="checkbox"/> GET <input checked="" type="checkbox"/> PUT <input checked="" type="checkbox"/> POST <input checked="" type="checkbox"/> DELETE	<button>Delete</button>

API resources can accept following optional attributes:

- **verbs:** Specifies the HTTP verbs a particular resource would accept. Allowed values are GET, POST, PUT, DELETE. Multiple values can be specified.
- **uri-template:** A URI template as defined in <http://tools.ietf.org/html/rfc6570> (eg: /phoneverify/{phoneNumber})
- **url-mapping:** A URL mapping as defined as per the servlet specification (extension mappings, path mappings and exact mappings)

Once a request has been accepted by a resource, it will be mediated through an in-sequence. Any response from the backend is handled through the out-sequence. Fault sequence is used to mediate 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:

APIs / All / Phone Verification-1.0.0

Phone Verification - 1.0.0

[Overview](#)
[Edit](#)
[Life Cycle](#)
[Versions](#)
[Docs](#)
[Users](#)


0 Users
 CREATED
 1.0.0
 Docs

The Phone Verification API allows you to enhance telemarketing effectiveness and increase the integrity of your contact database

Endpoint URL	http://ws.cdyne.com/phoneverify/phoneverify.asmx
WSDL	http://ws.cdyne.com/phoneverify/phoneverify.asmx?wsdl
Date Last Updated	Thu Aug 02 15:50:47 CEST 2012
Tier Availability	Bronze,Gold,Silver
Tags	open,cdyne,phone,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 or via a URL. For inline documentation, you can edit the contents directly from the API publisher interface.

Several documents types are available:

- How To
- Samples and SDK
- Public forum / Support forum (external link only)
- API message formats
- Other

To create a How-To document:

1. Select the **How To** type
2. Provide a name for the document
3. Provide a short description of the document (this will appear in the API store)
4. Select whether the document is stored inline or provided via a URL
5. Click **Add New Document**

<http://wso2.com>

API Manager | Getting Started Guide

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.

APIs / All / PhoneVerification-1.0.0

PhoneVerification-v1.0.0

Overview Edit Versions Docs Users

+ Add New Document

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.:

Add Document Cancel

Name	Type	Content	Modified On	Actions
README	How To	View	Wed Jun 20 17:21:11 CEST 2012	Edit Content Update Delete
Pricing	Other	View	Wed Jun 20 17:21:38 CEST 2012	Edit Content Update Delete

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.

PhoneVerification - 1.0.0

Overview Edit Life Cycle Versions Docs Users

State: PUBLISHED

☒ Propagate Changes to API Gateway

Update Reset

Life-Cycle History

- 2012-07-07 20:03:31.376 isabelle created the API.
- 2012-07-07 20:04:49.591 isabelle 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**.

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.

Phone Verification - 1.0.0

[Overview](#) [Edit](#) [Life Cycle](#) [Versions](#) [Docs](#) [Users](#)



0 Users
CREATED
1.0.0
Docs

The Phone Verification API allows you to enhance telemarketing effectiveness and increase the integrity of your contact database

Endpoint URL	http://ws.cdyne.com/phonerify/phonerify.asmx
WSDL	http://ws.cdyne.com/phonerify/phonerify.asmx?wsdl
Date Last Updated	Thu Aug 02 16:07:59 CEST 2012
Tier Availability	Bronze,Gold,Silver
Tags	open,cdyne,phone,data validation
Business Owner	Acme Corp [operations@acme.com]
Technical Owner	Cdyne [info@cdyne.com]

To Version

Ex:v1.0.1

[Done](#) [Cancel](#)

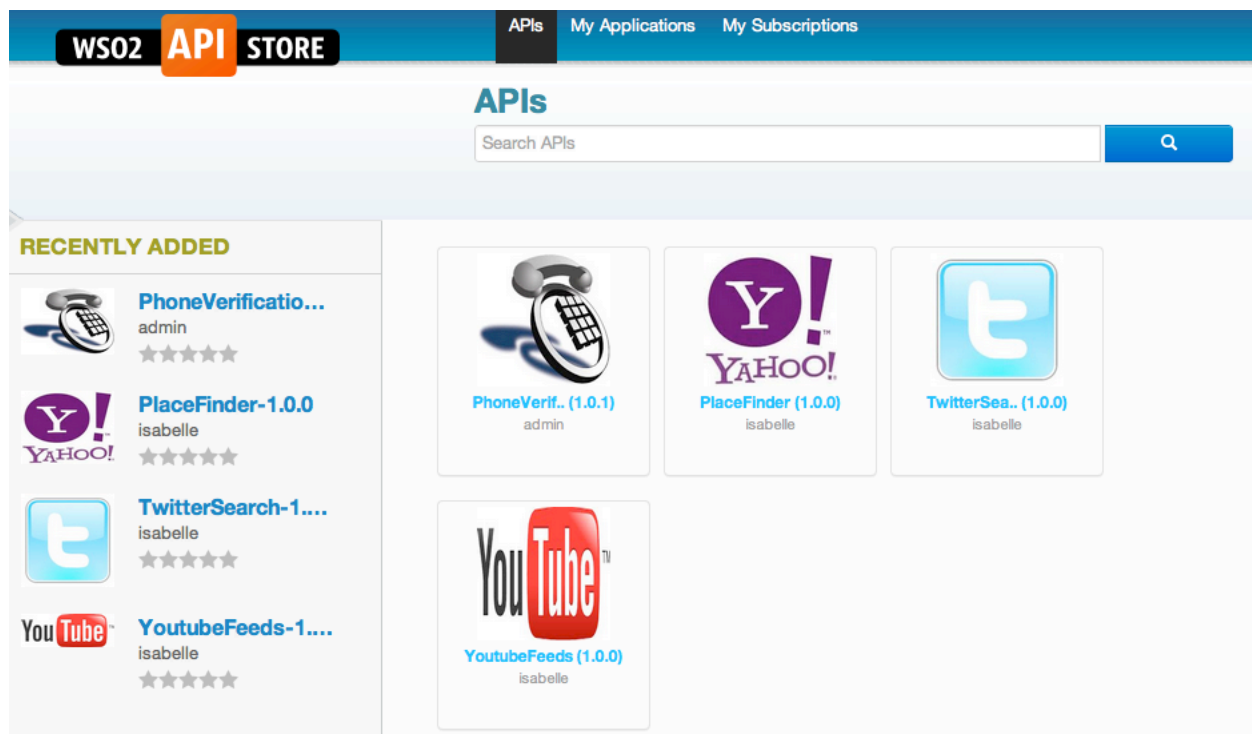
Using the API store

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:

PhoneVerification - 1.0.1

admin



Version:	1.0.1
Status:	PUBLISHED
Updated:	Thu Aug 02 17:48:18 CEST 2012

Overview

Documentation

URLs:

- <http://192.168.1.41:8280/phoneverify/1.0.1>
- <https://192.168.1.41:8243/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 combinations 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. Phone Verification will reduce data entry errors and provide the ability to verify any United States area code + prefix combination in Batch or Real-time mode. If the area code is incorrect or missing, Phone Verification can identify the error or supply a correct one, standardizing your phone numbers for consistency throughout your database. The Web Service does not check the last 4 digits of the phone number.

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 (this search is case sensitive at this point) or by clicking on the tags to the right.

You can consult the documentation from the Documentation tab.

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

APICREATOR



Rating:



Your rating: 0/5



Version:

1.0.1

Status:

PUBLISHED

Updated:

Wed Jun 20 18:53:15 CEST 2012

Applications

Select Application...

Tiers

Bronze

Allows 1 request per minute

Subscribe

- 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.
- Select the tiers (Service Level) you need - The description of the service level is shown below the Tiers field.
- Click **Subscribe**
- Once the subscription is successful , you can switch to My Subscriptions .
- 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.

DefaultApplication

Keys

Production		Re-generate	Hide Keys
Access Token	Gzd5pkRGdfxgzEWA3xCtBSs1oWla		
Consumer Key	JVIVyCIRSGHs9AZWe0ufDO4BVv0a		
Consumer Secret	3Zee66imMJPQuBky7wDrdisWYLka		

Sandbox

Sandbox keys are not yet generated for this application.

Generate

APIs

PhoneVerif.. - 1.0.1
✕

isabelle
Bronze Subscription

You have now successfully subscribed to the API and can start using it.

Calling an API

To test the API, you can use a simple REST client application (or `curl`). The Cdyne 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?PhoneNumber=XXXX&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 9oGKRz6HpGFqtMwfwHrCj9pOKowa`.

The screenshot shows a REST client interface with the following details:

- URL:** `http://192.168.1.41:8280/phoneverify/1.0.1/CheckPhoneNumber?PhoneNumber=+14158878177&LicenseKey=0`
- Method:** `GET` (selected)
- Headers:**
 - Raw input / Form:**

Authorization	Bearer 9oGKRz6HpGFqtMwfwHrCj9pOKowa	Remove
Add row		
- Buttons:** Clear form, Send request
- Status code:** **200 OK**
- Time:** 1201 ms

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.

The screenshot shows a REST client interface with the following details:

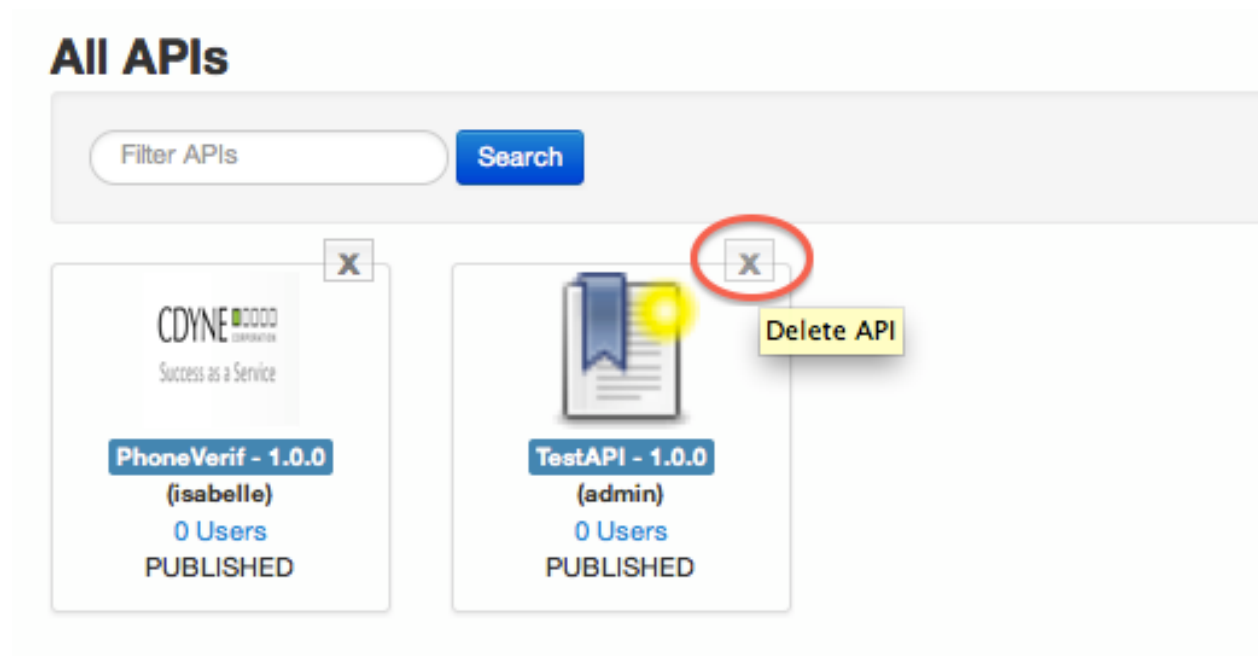
- Status code:** **503 Service Unavailable**
- Time:** 8 ms
- Headers:**
 - Raw response / XML:**

```
<?xml version='1.0' encoding='UTF-8'>
<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>
```


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:

- Usage of APIs per creator
- Usage of all APIs
- Average response times
- Usage of an API per subscriber
- Usage of an API per subscriber per version
- Number of subscriptions per API

Aside from the number of subscriptions per API, all other dashboards require to use WSO2 Business Activity Monitoring 2.0.0 for analytics - You need to use a WSO2 BAM 2.0.x, available for download on our web site. Simply unzip BAM2 to install it.

Enabling and configuring statistics

In order to use BAM2 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:

- a) Edit this file and locate the `APIUsageTracking` element
 - b) Make sure the value is set to **true**.
 - c) Restart the API manager
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/API_MGT_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) 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.
 - c) Start WSO2 BAM server from <BAM_HOME>/bin/wso2server.[sh/bat]

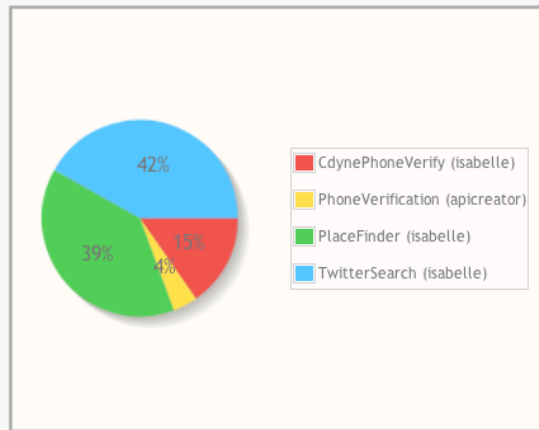
Viewing Statistics

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:

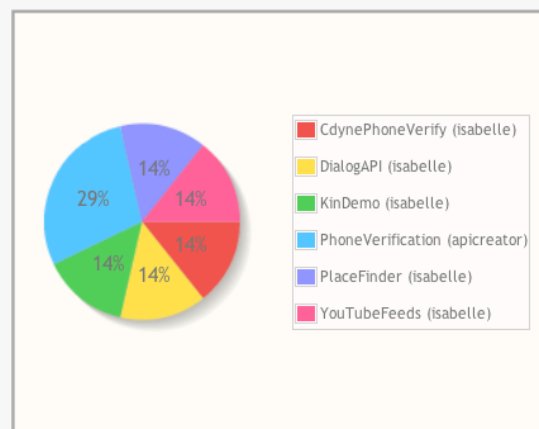
1. Connect to the API publisher web application as a creator or publisher. In publisher role, you are able to see all stats, as creator you also see specific stats for the APIs you created.
2. Go to **APIs > All Statistics**

Overall API Usage (Across All Versions)



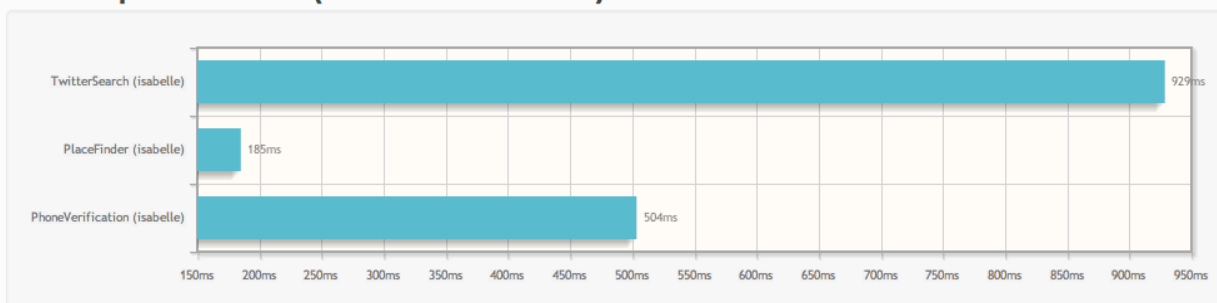
API Name	Number of API Calls
CdynePhoneVerify (isabelle)	15
PhoneVerification (apicreator)	4
PlaceFinder (isabelle)	38
TwitterSearch (isabelle)	41

Overall API Subscriptions (Across All Versions)



API Name	Number of Subscriptions
CdynePhoneVerify (isabelle)	1
DialogAPI (isabelle)	1
KinDemo (isabelle)	1
PhoneVerification (apicreator)	2
PlaceFinder (isabelle)	1
YouTubeFeeds (isabelle)	1

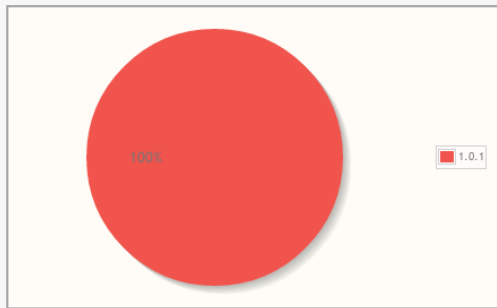
API Response Times (Across All Versions)



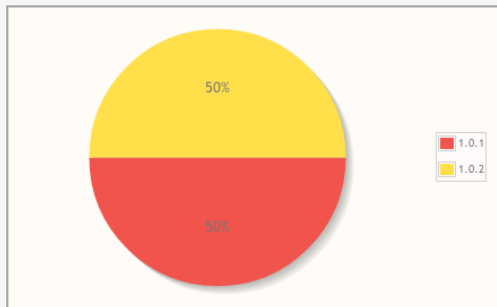
API Last Access Times (Across All Versions)

API	Last Accessed Version	Subscriber	Access Time
CdynePhoneVerify (isabelle)	1.0.0	martin	6/19/12 6:39 PM
PhoneVerification (apicreator)	1.0.1	charith	6/21/12 2:18 PM
PlaceFinder (isabelle)	1.0.0	martin	6/19/12 6:41 PM
TwitterSearch (isabelle)	1.0.0	subscriber1	6/14/12 5:41 PM

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

PhoneVerification-v1.0.1[Overview](#) [Edit](#) [Life Cycle](#) [Versions](#) [Docs](#) [Users](#)**API Usage by Versions**

Version	Number of API Calls
1.0.1	9

API Subscriptions by Versions

Version	Number of Subscriptions
1.0.1	2
1.0.2	2

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

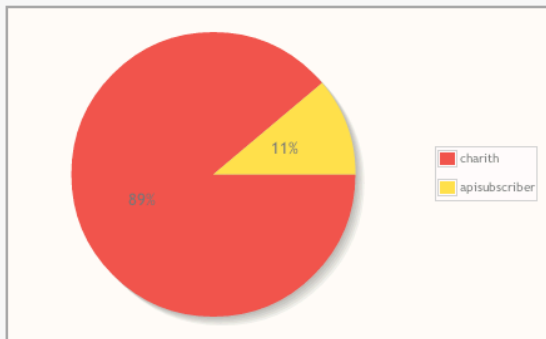
PhoneVerification-v1.0.1

[Overview](#) [Edit](#) [Life Cycle](#) [Versions](#) [Docs](#) [Users](#)

Active Subscriptions

Name	Date of Subscription
charith	2012-06-21 12:30:25.707
apisubscriber	2012-06-20 22:46:44.042

Usage by Current Subscribers (v-1.0.1)



Subscriber	Number of API Calls
charith	8
apisubscriber	1

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.

1. 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
```

2. 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>
```

3. 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}/service
s/</ServerURL>
```

...

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

...

```
<APIKeyManager>
```

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

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

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
: WS02 API Manager-1.1.0
[2012-10-30 10:13:24,933] INFO - StartupFinalizerServiceComponent WS02
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).

The screenshot shows the WSO2 API Manager Management Console. The top navigation bar includes the WSO2 logo, 'API Manager', and the title 'Management Console'. It also shows the user is signed in as 'admin@192.168.1.35:9443' with links for 'Sign-out', 'Docs', and 'About'. The left sidebar contains a 'Manage' section with options like 'Jaggery Applications', 'Service Bus', 'APIs', 'Source View', and 'Shutdown/Restart'. The main content area is titled 'Shutdown/Restart Server' and contains two tables: 'Shutdown' and 'Restart'. Each table has two columns: 'Graceful' and 'Forced'. The 'Shutdown' table shows 'Graceful Shutdown' (with a green checkmark icon) and 'Forced Shutdown' (with a red warning icon). The 'Restart' table shows 'Graceful Restart' (with a green checkmark icon) and 'Forced Restart' (with a green checkmark icon).

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

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