Setting up AM

1. Download and extract the wso2am-2.5.0.zip.
2. Download the mysql-connector-java-5.1.44.tar.gz mysql connector jar and copy into {API-M_HOME}/repository/components/lib directory
3. Set lower_case_table_names = 1 in /etc/mysql/my.cnf

Database configurations

4. If Db is remotely setup please check the connectivity mysql -uroot -p -h <ip>
5. You need to run the database scripts to setup the required databases as follows

a). AM Database setup

mysql> create database apimgtdb;
mysql> use apimgtdb;
mysql> source <API-M_HOME>/dbscripts/apimgt/mysql5.7.sql;
mysql> grant all on apimgtdb. TO 'wso2user'@'%' identified by 'wso2123';*

mysql> create database userdb;
mysql> use userdb;
mysql> source <API-M_HOME>/dbscripts/mysql5.7.sql;
mysql> grant all on userdb. TO 'wso2user'@'%' identified by 'wso2123';*

mysql> create database regdb;
mysql> use regdb;
mysql> source <API-M_HOME>/dbscripts/mysql5.7.sql;
mysql> grant all on regdb. TO 'wso2user'@'%' identified by 'wso2123';*

mysql> create database statdb;
mysql> use statdb;
mysql> source wso2telcohub-3.1.1-SNAPSHOT/dbscripts/dep-hub/mysql/stats_db.sql;
mysql> grant all on statdb. TO 'wso2user'@'%' identified by 'wso2123';*

mysql> create database mbstoredb;
mysql> use mbstoredb;
mysql> source <API-M_HOME>/dbscripts/mb-store/mysql-mb.sql;
mysql> grant all on mbstoredb. TO 'wso2user'@'%' identified by 'wso2123'*

7. The databases are defined in <API-M_HOME>/repository/conf/datasources/master-datacontensource.xml file. Change the Username, password and databases names. Change host and port names of dbs.

Change configuration files

8.Go to <API-M_HOME>/repository/conf folder and add the following files from that of the previous built pack.

- Workflow.properties
- workflow-configuration.xml
- spendLimit.xml
- Oneapi-validation-conf.properties
- MobileCountryConfig.xml

Deploying Artifacts (Jars) on APIM 2.5.0

Go to <API-M_HOME>/repository/components/plugins folder of the previous built pack and filter the jar files with the “com.wso2telco.” prefix. Add those to the <API-M_HOME>/repository/components/dropins folder of the new pack.

- com.wso2telco.core.dbutils_2.4.5.SNAPSHOT.jar
- com.wso2telco.core.mnc-resolver_2.4.5.SNAPSHOT.jar
- com.wso2telco.core.msisdn-validator_2.4.5.SNAPSHOT.jar
- com.wso2telco.core.user-profile_2.4.5.SNAPSHOT.jar
- com.wso2telco.dep.billing-extension_2.4.5.SNAPSHOT.jar
- com.wso2telco.dep.billing-service_2.4.5.SNAPSHOT.jar
- com.wso2telco.dep.logging-extension_2.4.5.SNAPSHOT.jar
- com.wso2telco.dep.mediator_2.4.5.SNAPSHOT.jar
- com.wso2telco.dep.nashorn-mediator_2.4.5.SNAPSHOT.jar
- com.wso2telco.dep.oneapi-validation_2.4.5.SNAPSHOT.jar
- com.wso2telco.dep.operator-service_2.4.5.SNAPSHOT.jar
- com.wso2telco.dep.reporting-service_2.4.5.SNAPSHOT.jar
- com.wso2telco.dep.server.startup.observer_2.4.5.SNAPSHOT.jar
- com.wso2telco.dep.subscription-validator_2.4.5.SNAPSHOT.jar
- com.wso2telco.dep.verification-handler_2.4.5.SNAPSHOT.jar
- com.wso2telco.dep.workflow.extensions_2.4.5.SNAPSHOT.jar

Deploying Webapps on APIM 2.5.0

Go to <API-M_HOME>/repository/deployment/server/webapps folder of the previous built pack and add the following war files to the new pack.

- activiti-explorer.war
- activiti-rest.war
- aggregator-blacklist.war
- blacklist-whitelist-service.war
- manage-service.war
- quota-service.war
- ratecard-service.war
- workflow-service.war

Errors :
[2018-12-27 09:51:23,959] ERROR - ApplicationContext StandardWrapper.Throwable
java.lang.NoSuchMethodError:
org.springframework.web.context.support.XmlWebApplicationContext.getEnvironment()Lorg/springframework/core/env/ConfigurableEnvironment;

[2018-12-27 09:51:23,960] ERROR - StandardContext Servlet [cxf] in web application [/aggregator-blacklist] threw load() exception
java.lang.NoSuchMethodError:
org.springframework.web.context.support.XmlWebApplicationContext.getEnvironment()Lorg/springframework/core/env/ConfigurableEnvironment;

[2018-12-27 09:51:33,681] ERROR - ApplicationContext StandardWrapper.Throwable
java.lang.NoSuchMethodError:
org.springframework.web.context.support.XmlWebApplicationContext.getEnvironment()Lorg/springframework/core/env/ConfigurableEnvironment;

[2018-12-27 09:51:33,681] ERROR - StandardContext Servlet [cxf] in web application [/blacklist-whitelist-service] threw load() exception
java.lang.NoSuchMethodError:
org.springframework.web.context.support.XmlWebApplicationContext.getEnvironment()Lorg/springframework/core/env/ConfigurableEnvironment;

[2018-12-27 09:51:39,730] ERROR - ApplicationContext StandardWrapper.Throwable
java.lang.NoSuchMethodError:
org.springframework.web.context.support.XmlWebApplicationContext.getEnvironment()Lorg/springframework/core/env/ConfigurableEnvironment;

[2018-12-27 09:51:39,734] ERROR - StandardContext Servlet [cxf] in web application [/quota-service] threw load() exception
java.lang.NoSuchMethodError:
org.springframework.web.context.support.XmlWebApplicationContext.getEnvironment()Lorg/springframework/core/env/ConfigurableEnvironment;
at
org.springframework.web.context.support.XmlWebApplicationContext.loadBeanDefinitions(Xml

WebApplicationContext.java:87)

Error encountered while deploying

- quota-service.war
- aggregator-blacklist.war
- blacklist-whitelist-service.war

Take a pull from https://github.com/Rajithkonara/component-dep/commit/2833298f4cfb536c42073f71d5d605c025dc1aa2

Build component-dep.

1)

replace the following war files to /home/chamani/Documents/APIM25/wso2am-2.5.0/repository/deployment/server/webapps

/home/chamani/Documents/projects/component-dep/components/webapps/blacklist-whitelist-service/target/blacklist-whitelist-service.war

/home/chamani/Documents/projects/component-dep/components/webapps/aggregator-blacklist/target/aggregator-blacklist.war

/home/chamani/Documents/projects/component-dep/components/webapps/quota-service/target/quota-service.war

2) Replace /home/chamani/Documents/APIM25/wso2am-2.5.0/repository/deployment/server/webapps with

/home/chamani/Documents/projects/component-dep/components/webapps/aggregator-blacklist/target/aggregator-blacklist-2.4.5-SNAPSHOT and rename as aggregator-blacklist. Do the same for the other 3 war files

Getting Jaggery changes to APIM 2.5.0

Jaggery files changed:

1) **Related to Store:**

Add the changes of following files of wso2telcohub-3.1.1-SNAPSHOT pack to that of the

new pack

- store/jaggery.conf
- store/modules/manager/manager.jag and
- store/modules/manager/module.jag and
- store/modules/user/module.jag
- store/modules/user/user.jag
- store/site/blocks/api/api-info/block.jag
- store/site/blocks/api-doc/ajax/get.jag
- store/site/blocks/subscription/subscription-list/ajax/subscription-list.jag
- store/site/blocks/subscription/subscription-remove/ajax/subscription-remove.jag
- store/site/blocks/user/user-info/block.jag
- store/site/conf/locales/jaggery/locale_default.json
- store/site/conf/locales/jaggery/locale_en.json
- store/site/themes/wso2/css/custom.css and
- store/site/themes/wso2/js/applications.js
- store/site/themes/wso2/js/script.js
- store/site/themes/wso2/libs/theme-wso2_1.0/css/theme-wso2.css
- store/site/themes/wso2/libs/theme-wso2_1.0/images/logo-inverse.svg
- store/site/themes/wso2/templates/api/api-info/js/api-info.js and
- store/site/themes/wso2/templates/api/api-info/template.jag and
- store/site/themes/wso2/templates/application/application-add/template.jag
- store/site/themes/wso2/templates/application/application-edit/template.jag
- store/site/themes/wso2/templates/application/application-list/template.jag
- store/site/themes/wso2/templates/menu/header/template.jag
- store/site/themes/wso2/templates/menu/primary/template.jag
- store/site/themes/wso2/templates/page/base/template.jag
- store/site/themes/wso2/templates/sso/filter/template.jag
- store/site/themes/wso2/templates/sso/logout/template.jag and
- store/site/themes/wso2/templates/subscription/subscribed-apis/template.jag
- store/site/themes/wso2/templates/user/login/template.jag
- store/site/themes/wso2/templates/user/user-info/template.jag

Add the following files of wso2telcohub-3.1.1-SNAPSHOT pack to that of the new pack

- Add store/modules: approval-history
- Add store/site/blocks: footer-page
- Add store/site/blocks: home
- Add store/site/blocks/workflow: workflow-operator
- Add store/site/pages: footer-page.jag
- Add store/site/pages: home.jag
- Add store/site/themes/wso2/templates: footer-page
- Add store/site/themes/wso2/templates: home
- Add store/site/themes/wso2/templates/page/base/js: jssor.js
- Add store/site/themes/wso2/templates/page/base/js: jssor.slider.js
-

Copy following images from wso2telcohub-3.1.1-SNAPSHOT/repository/deployment/server/jaggeryapps/store/site/themes/wso2/images to that of the new pack

- A-Radica-Open-Source-Digital-Enablement-Platform.png
- BizBannerWeb2.png
- Breakthrough-Technology.png
- Connected-Ecosystem.png
- facebook_16.png
- GSMA-Mobile-Connect-Accelerator.png
- home-btn-back.png
- linkedin_16.png
- logo-inverse.svg
- logo-white-custom.png
- logo-white.png (Replace the existing image with this image)
- poweredBy.png
- Telco-micro-slider-left-arrow.png
- Telco-micro-slider-right-arrow.png
- The-Perfect-Hybrid.png
- twitter_16.png

2) Related to Publisher:

Add the changes of following files of wso2telcohub-3.1.1-SNAPSHOT pack to that of the new pack

- publisher/site/themes/wso2/css/custom.css
- publisher/site/themes/wso2/images: logo-white-custom.png
- publisher/site/themes/wso2/images/logo-white.png
- publisher/site/themes/wso2/libs/theme-wso2_1.0/css/theme-wso2.css
- publisher/site/themes/wso2/libs/theme-wso2_1.0/images/logo-inverse.svg
- publisher/site/themes/wso2/templates/footer/template.jag
- publisher/site/themes/wso2/templates/listing/template.jag
- publisher/site/themes/wso2/templates/menu/actions/template.jag
- publisher/site/themes/wso2/templates/sso/filter/template.jag
- publisher/site/themes/wso2/templates/sso/logout/template.jag

Add the following files of wso2telcohub-3.1.1-SNAPSHOT pack to that of the new pack

- Add publisher/site/blocks: branding
- Add publisher/site/pages: branding.jag
- Add publisher/site/themes/wso2/images: logo-inverse.svg
- Add publisher/site/themes/wso2/images: poweredBy.png
- Add publisher/site/themes/wso2/templates: branding

Adding Claims

Navigate to https://192.168.2.96:9443/carbon
Go to Claims
Go to Add local claims

- Have to add Department, Operator name, OperatorID
- In APIM 2.5 version Department will be already there. Have to edit it
- Claim uri as follows.

http://wso2.org/claims/operatorName

http://wso2.org/claims/operatorID
http://wso2.org/claims/department

- For display Name, description use, Mapped attribute Department, Operator name, OperatorID
- Tick as below

Supported by Default	true
Required	false
Read-only	false

Creating users and roles

Navigate to https://192.168.2.96:9443/carbon
Go to Users and Roles
Create new Role
Ex: hub-dia-publisher

Note : dia is the department.
Set manage permissions of this role as :

- Subscription→ Tiers, VIsible
- Admin permission→ Login
- Manage→ API → Create, Publish, Subscribe

Create new user
Ex:publisher

Assign role to hub-dia-publisher

Change user profile as :

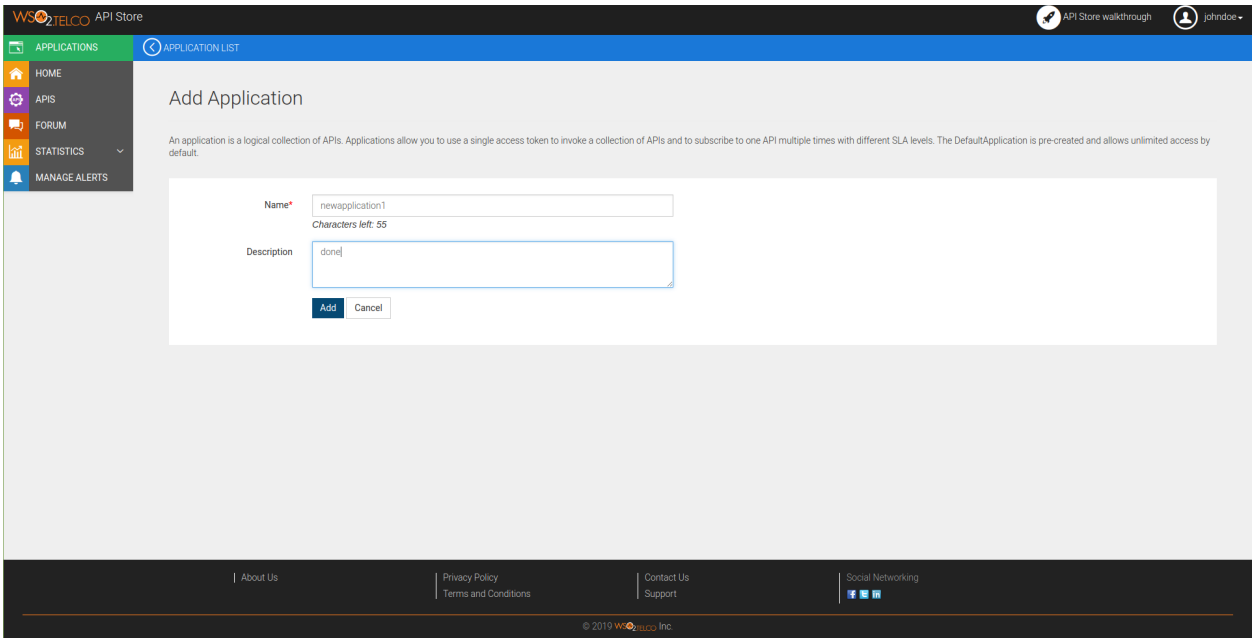
operatorName - operator1
operatorID - 1
Department - dia

Assign ui module permission of manage-app-admin

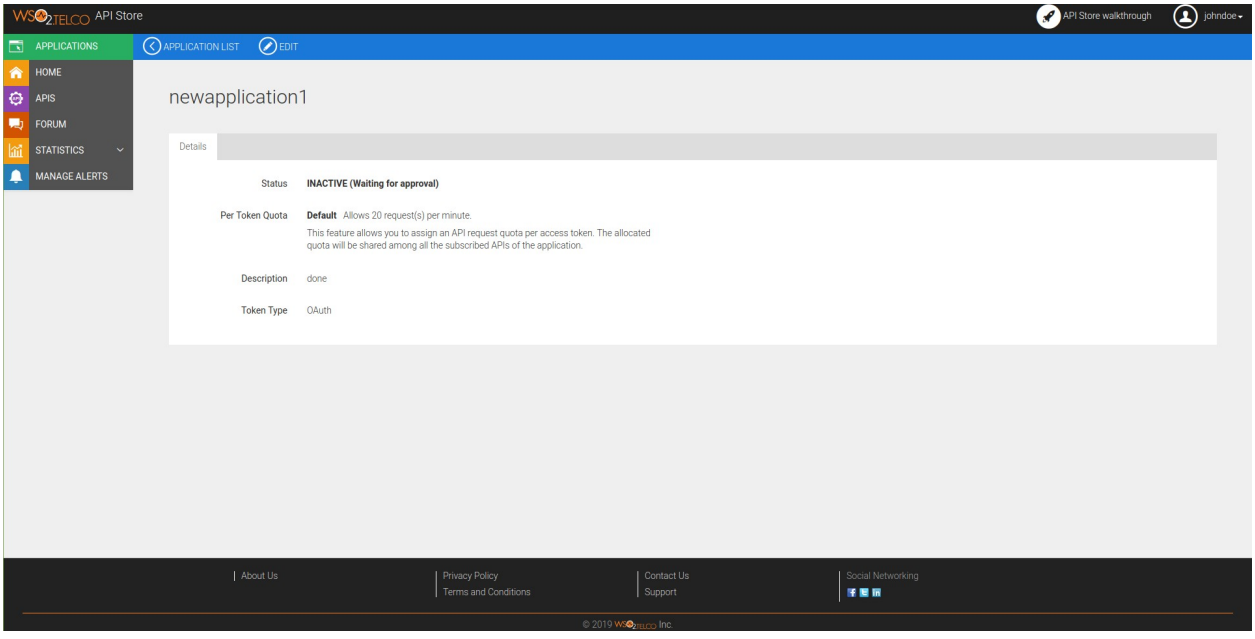
- UI Module Permission → application → tiers,visible
- Admin permissions → login

Steps to create an application in store

Create new user account and login to store.
Click add new application.



Status display as :**INACTIVE (Waiting for approval)**



Application Approval

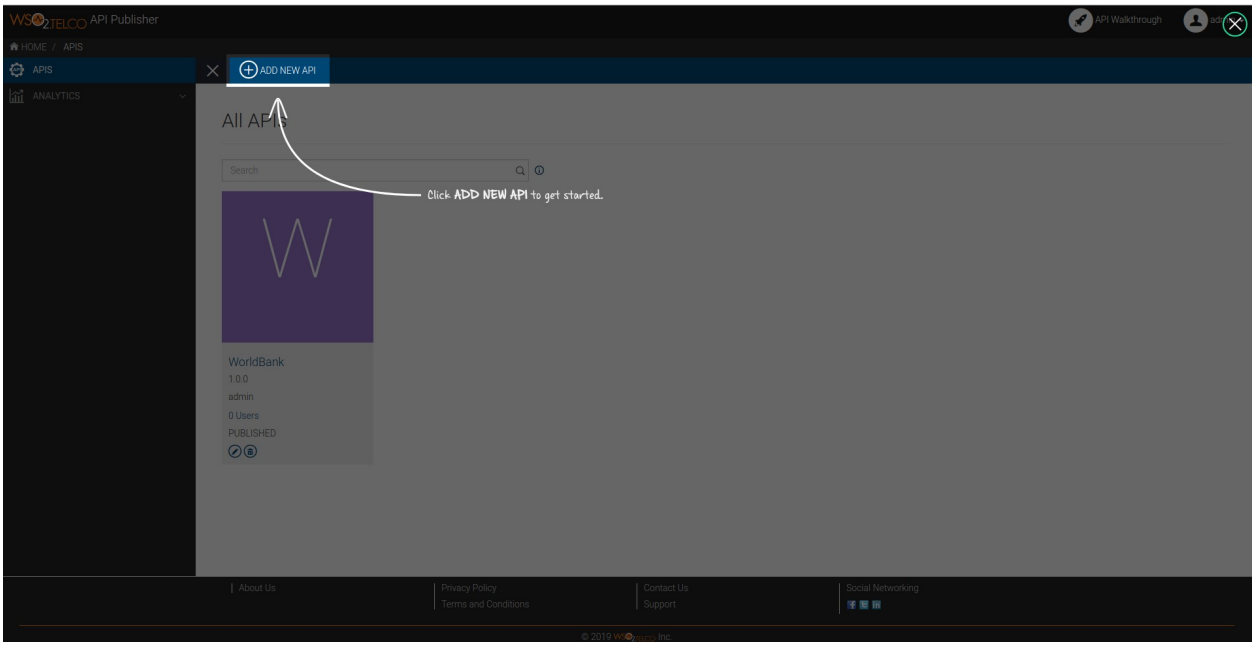
Go to https://192.168.2.96:9443/manage-service

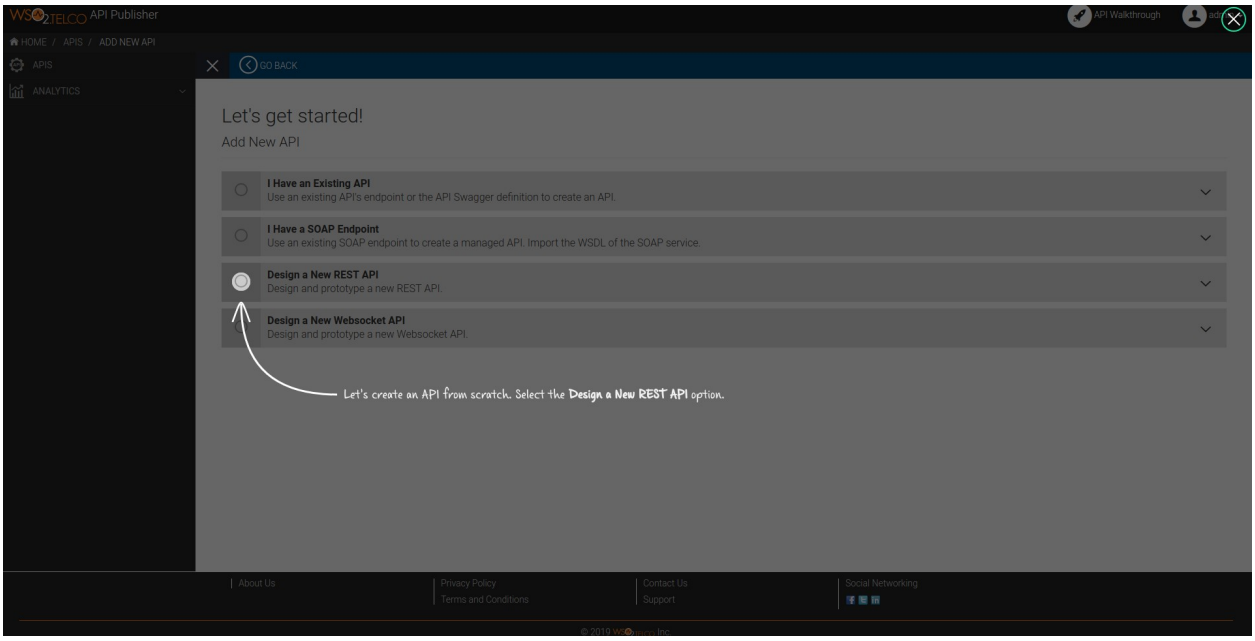
Login with admin,admin credentials

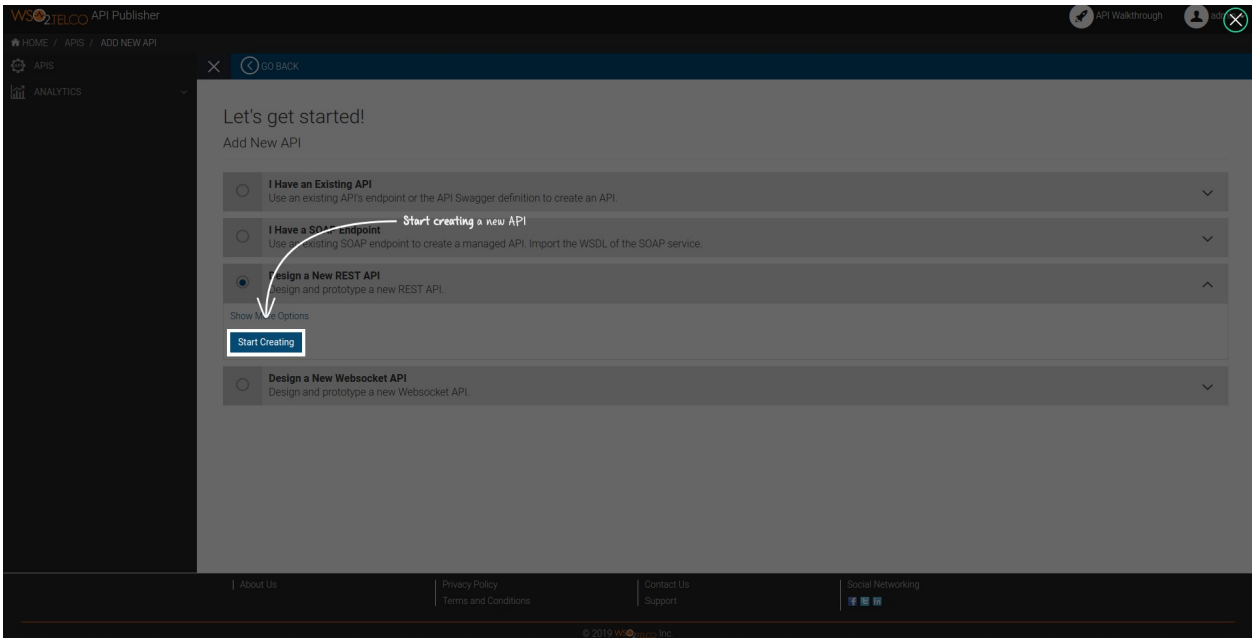
Approve the application

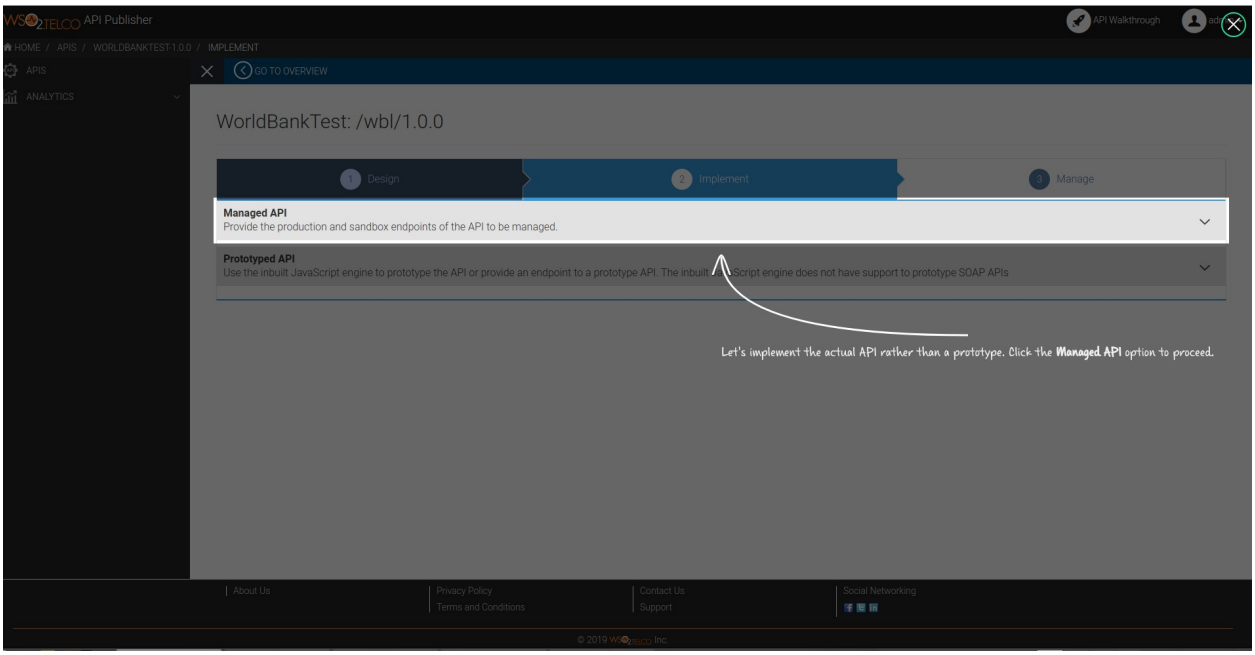
Steps to create an api in publisher

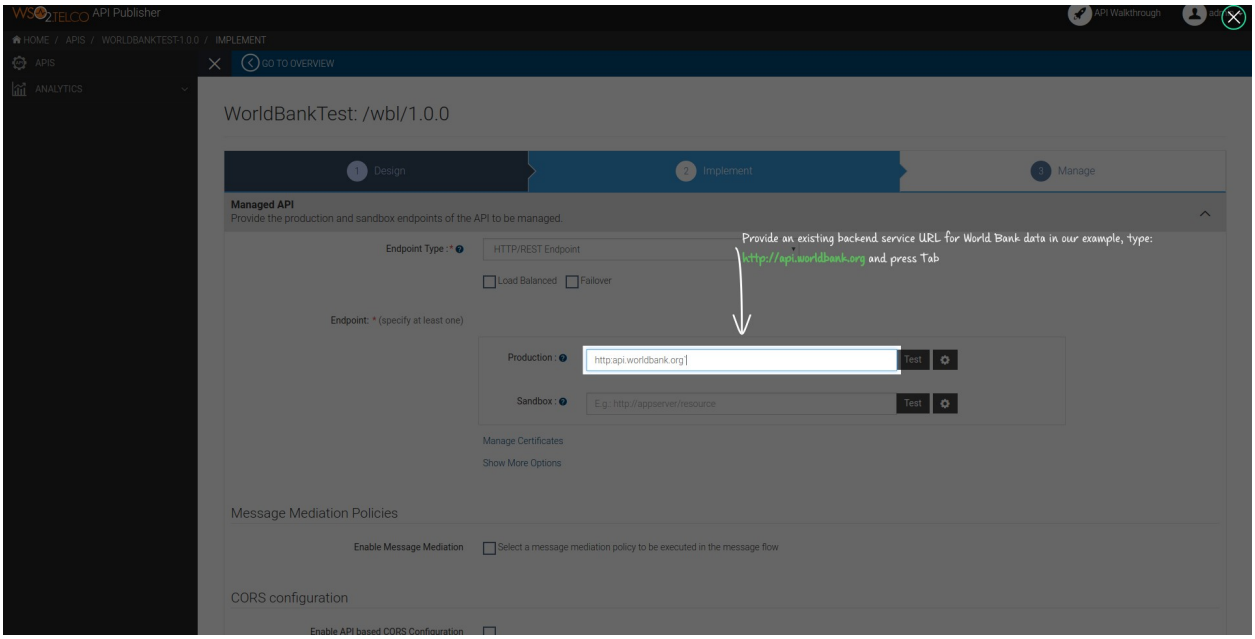
Login to publisher with publisher credentials
Create new API

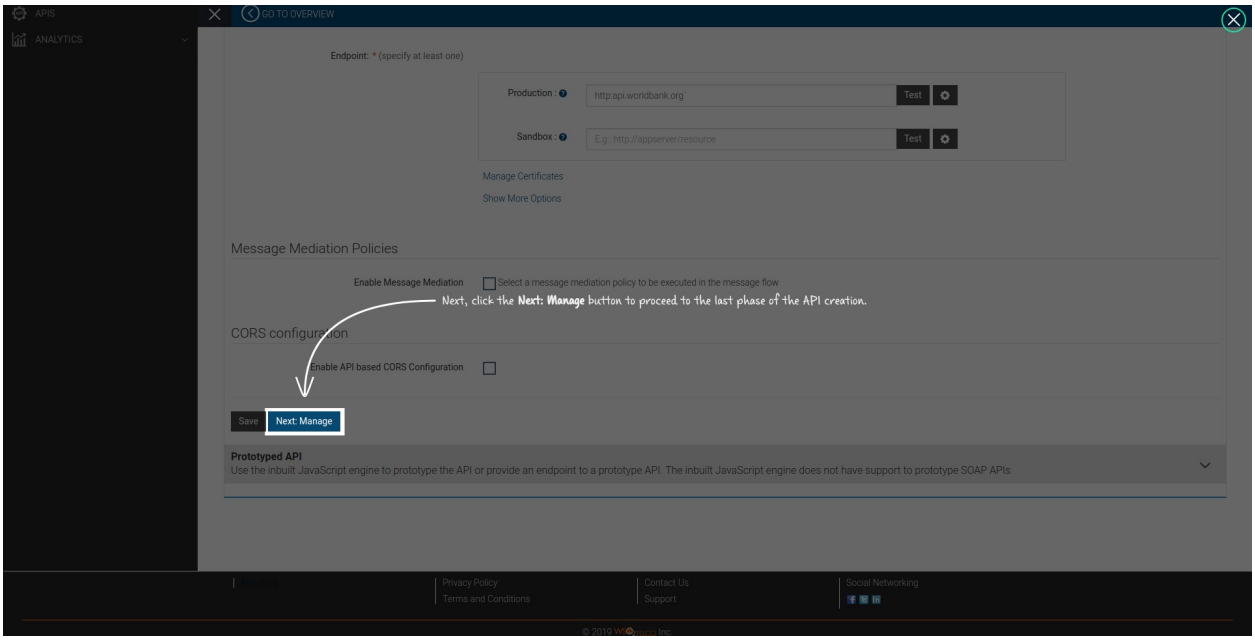


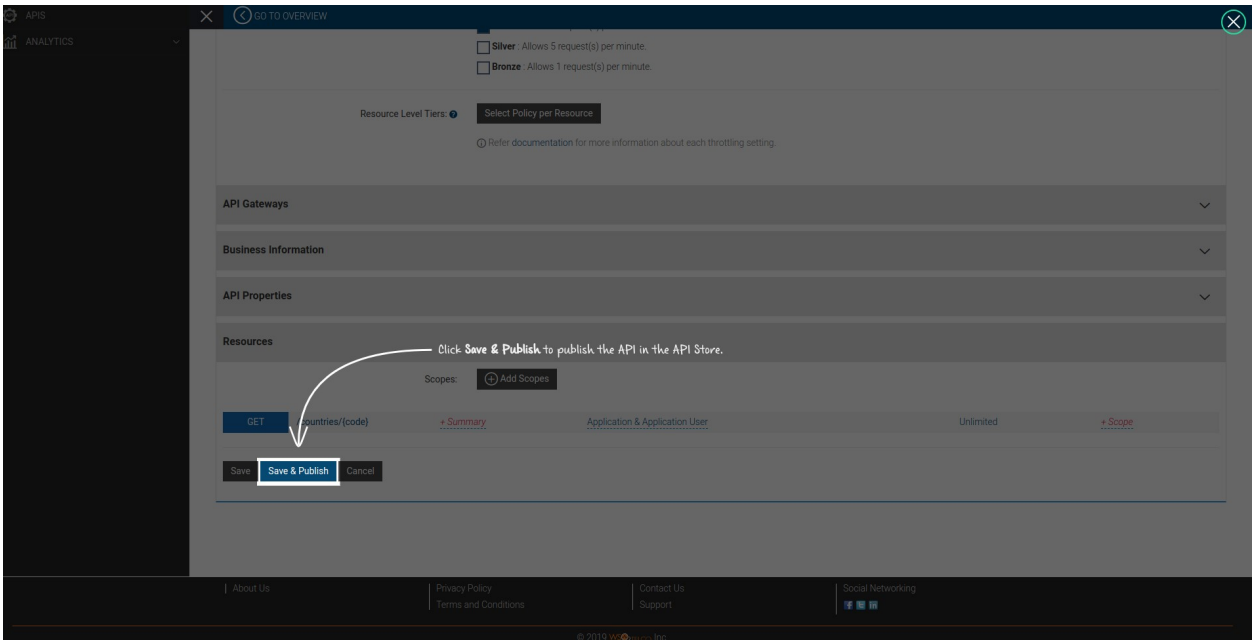


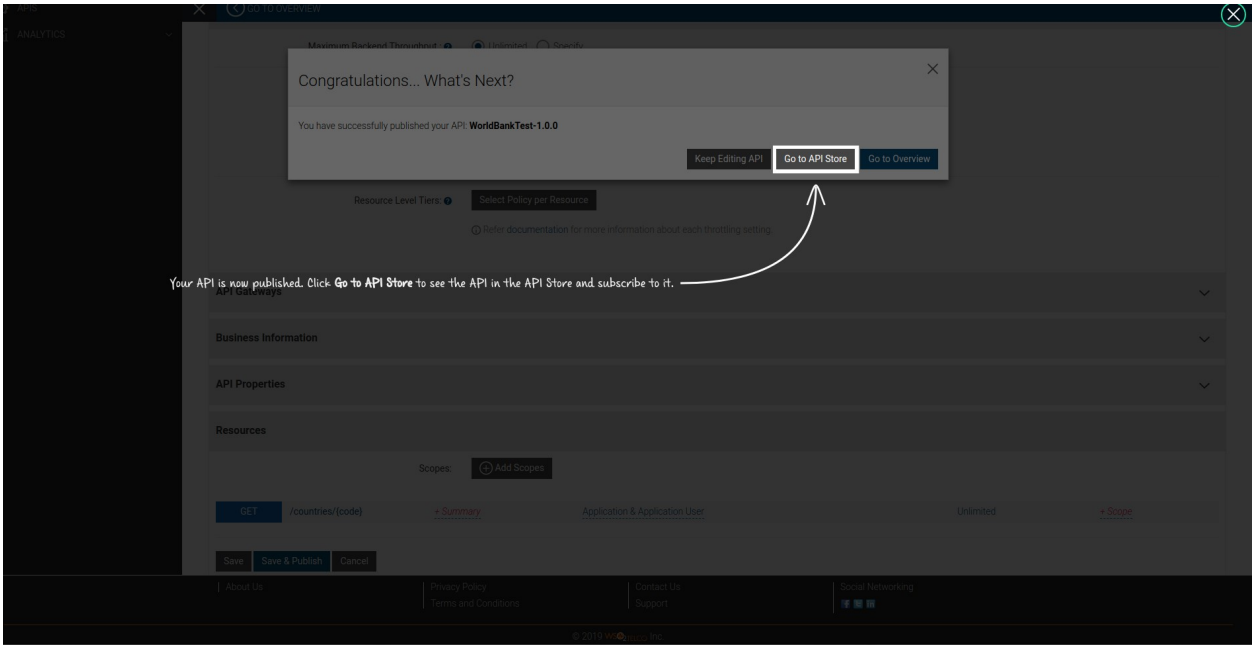

API Publisher
API Walkthrough
HOME / APIS / ADD NEW API
APIS
ANALYTICS
GO BACK
Let's get started!
Add New API
I Have an Existing API
Use an existing API's endpoint or the API Swagger definition to create an API.
I Have a SOAP Endpoint
Use an existing SOAP endpoint to create a managed API. Import the WSDL of the SOAP service.
Design a New REST API
Design and prototype a new REST API.
Start creating a new API
Show More Options
Start Creating
Design a New Websocket API
Design and prototype a new Websocket API.
About Us
Privacy Policy
Terms and Conditions
Contact Us
Support
Social Networking


API Publisher
API Walkthrough
HOME / APIS / WORLDBANKTEST-1.0.0 / IMPLEMENT
APIS
ANALYTICS
GO TO OVERVIEW
WorldBankTest: /wbl/1.0.0
1 Design
2 Implement
3 Manage
Managed API
Provide the production and sandbox endpoints of the API to be managed.
Prototyped API
Use the inbuilt JavaScript engine to prototype the API or provide an endpoint to a prototype API. The inbuilt JavaScript engine does not have support to prototype SOAP APIs
Let's implement the actual API rather than a prototype. Click the Managed API option to proceed.
About Us
Privacy Policy
Terms and Conditions
Contact Us
Support
Social Networking

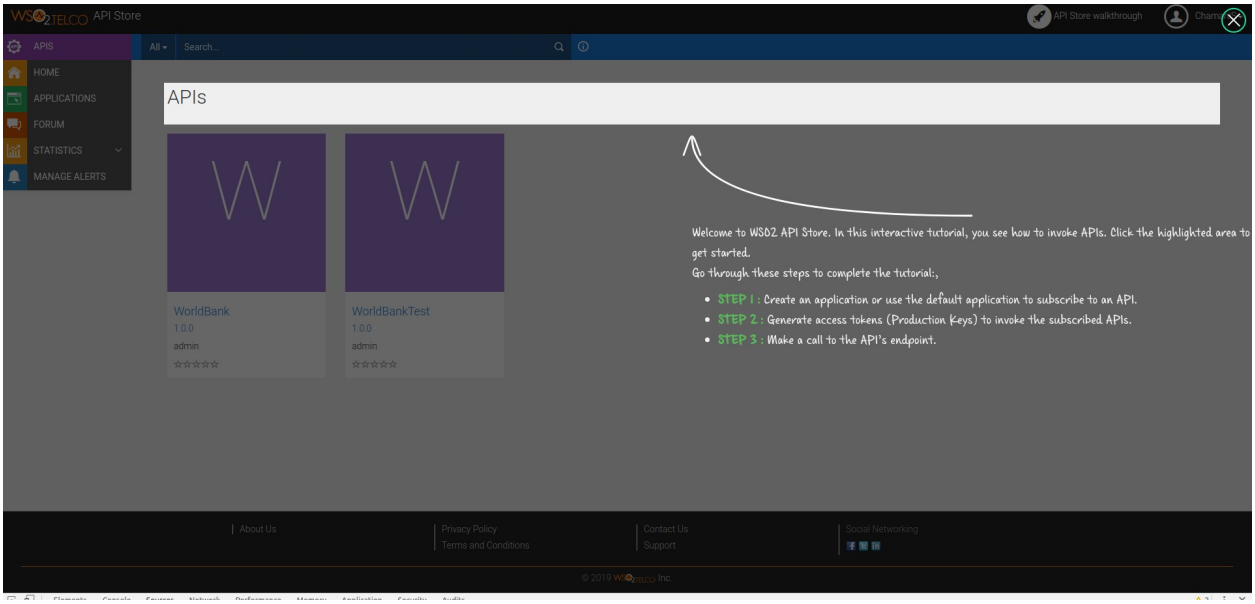

API Publisher
API Walkthrough
HOME / APIS / WORLDBANKTEST-1.0.0 / IMPLEMENT
APIS
ANALYTICS
GO TO OVERVIEW
WorldBankTest: /wbl/1.0.0
1 Design
2 Implement
3 Manage
Managed API
Provide the production and sandbox endpoints of the API to be managed.
Endpoint Type :*
HTTP/REST Endpoint
Load Balanced
Failover
Endpoint: * (specify at least one)
Production :
http://api.worldbank.org
Test
Sandbox :
E.g.: http://appserver/resource
Test
Manage Certificates
Show More Options
Provide an existing backend service URL for World Bank data in our example, type:
http://api.worldbank.org and press Tab
Message Mediation Policies
Enable Message Mediation
Select a message mediation policy to be executed in the message flow
CORS configuration

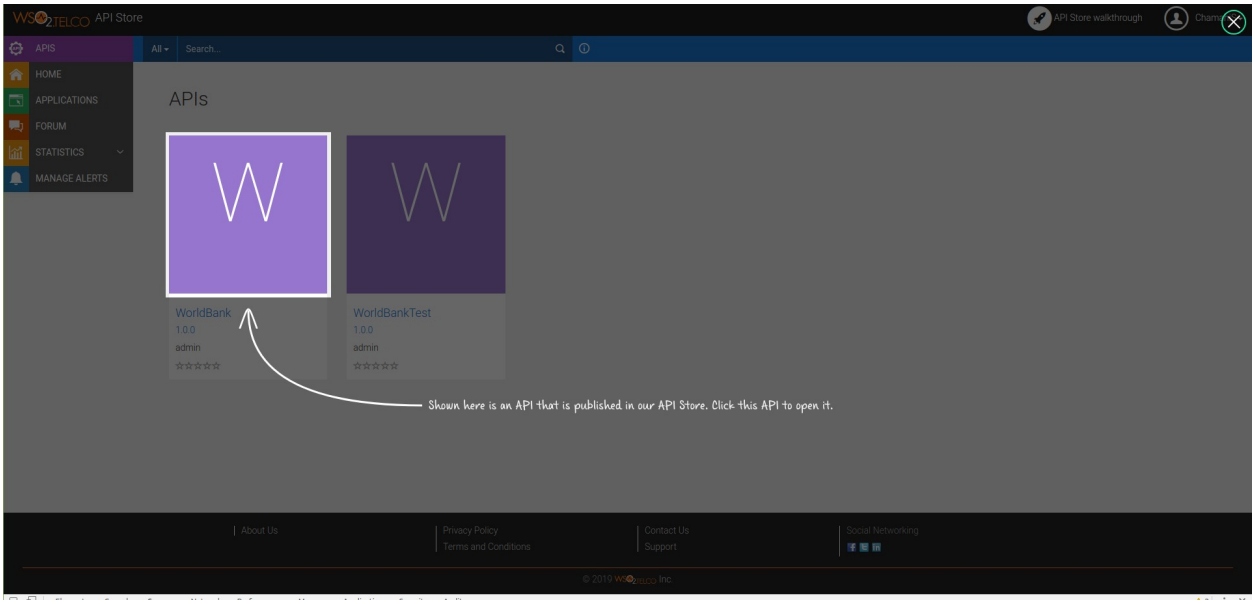

APIS
ANALYTICS
GO TO OVERVIEW
Endpoint: * (specify at least one)
Production :
http://api.worldbank.org
Test
Sandbox :
E.g.: http://appserver/resource
Test
Manage Certificates
Show More Options
Message Mediation Policies
Enable Message Mediation
Select a message mediation policy to be executed in the message flow
Next, click the Next: Manage button to proceed to the last phase of the API creation.
CORS configuration
Enable API based CORS Configuration
Save
Next: Manage
Prototyped API
Use the inbuilt JavaScript engine to prototype the API or provide an endpoint to a prototype API. The inbuilt JavaScript engine does not have support to prototype SOAP APIs
Privacy Policy
Terms and Conditions
Contact Us
Support
Social Networking

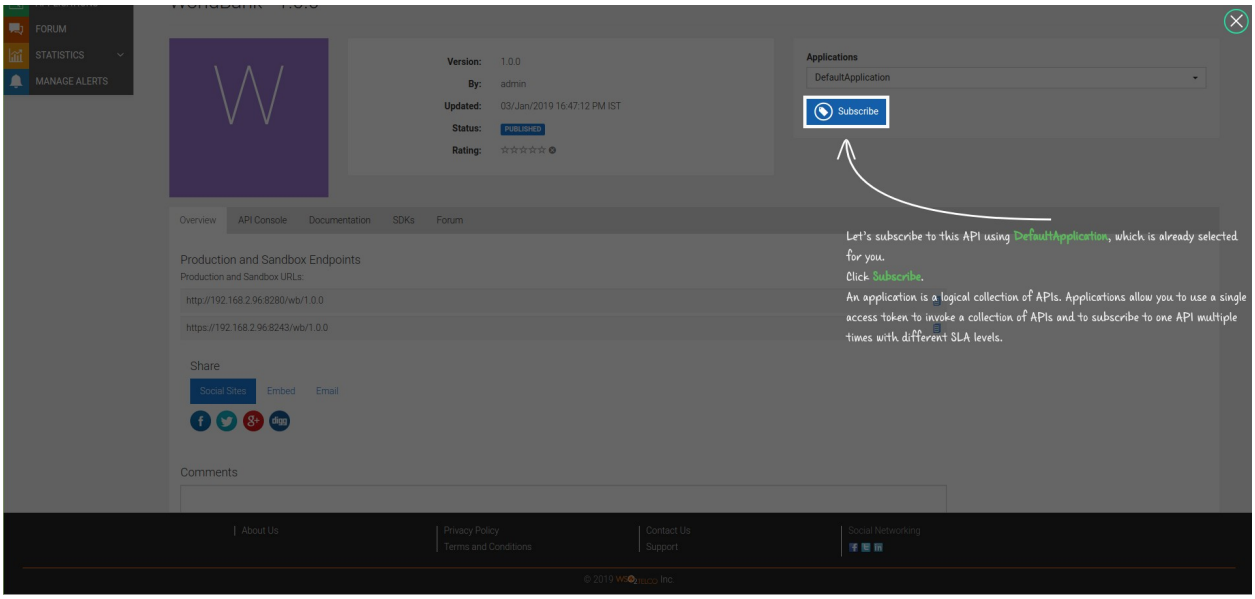

APIS
ANALYTICS
GO TO OVERVIEW
Silver : Allows 5 request(s) per minute.
Bronze : Allows 1 request(s) per minute.
Resource Level Tiers:
Select Policy per Resource
Refer documentation for more information about each throttling setting.
API Gateways
Business Information
API Properties
Resources
Click Save & Publish to publish the API in the API Store.
Scopes:
Add Scopes
GET
/countries/{code}
+ Summary
Application & Application User
Unlimited
+ Scope
Save
Save & Publish
Cancel
About Us
Privacy Policy
Terms and Conditions
Contact Us
Support
Social Networking

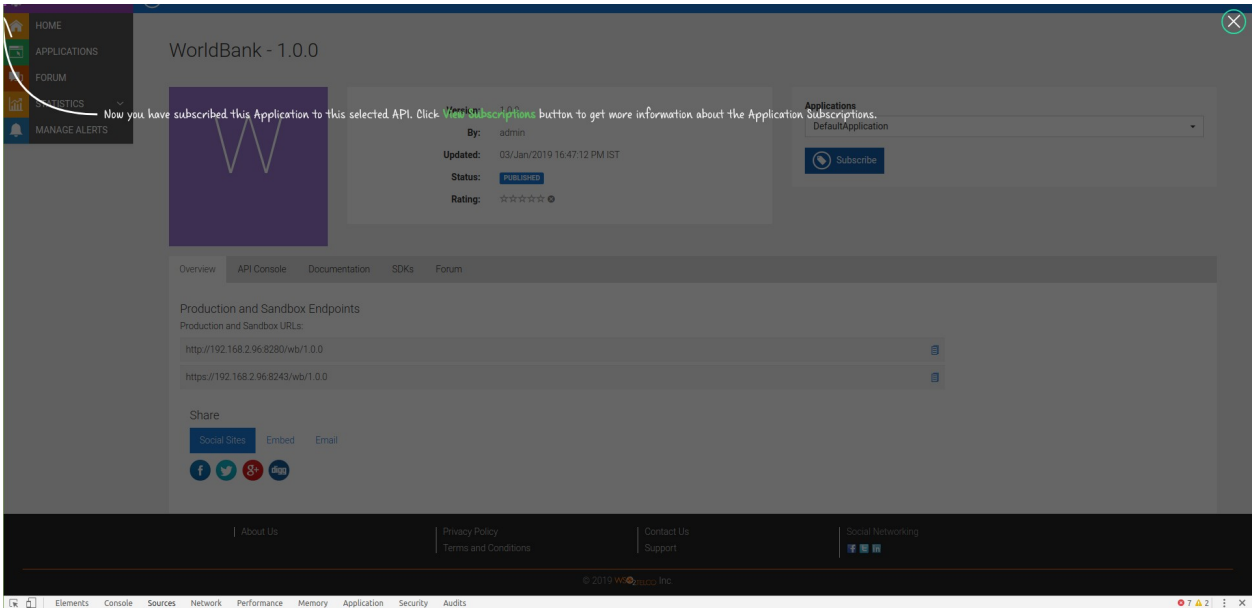

Congratulations... What's Next?
You have successfully published your API: WorldBankTest-1.0.0
Keep Editing API
Go to API Store
Go to Overview
Your API is now published. Click Go to API Store to see the API in the API Store and subscribe to it.
Business Information
API Properties
Resources
Scopes:
Add Scopes
GET
/countries/{code}
+ Summary
Application & Application User
Unlimited
+ Scope
Save
Save & Publish
Cancel

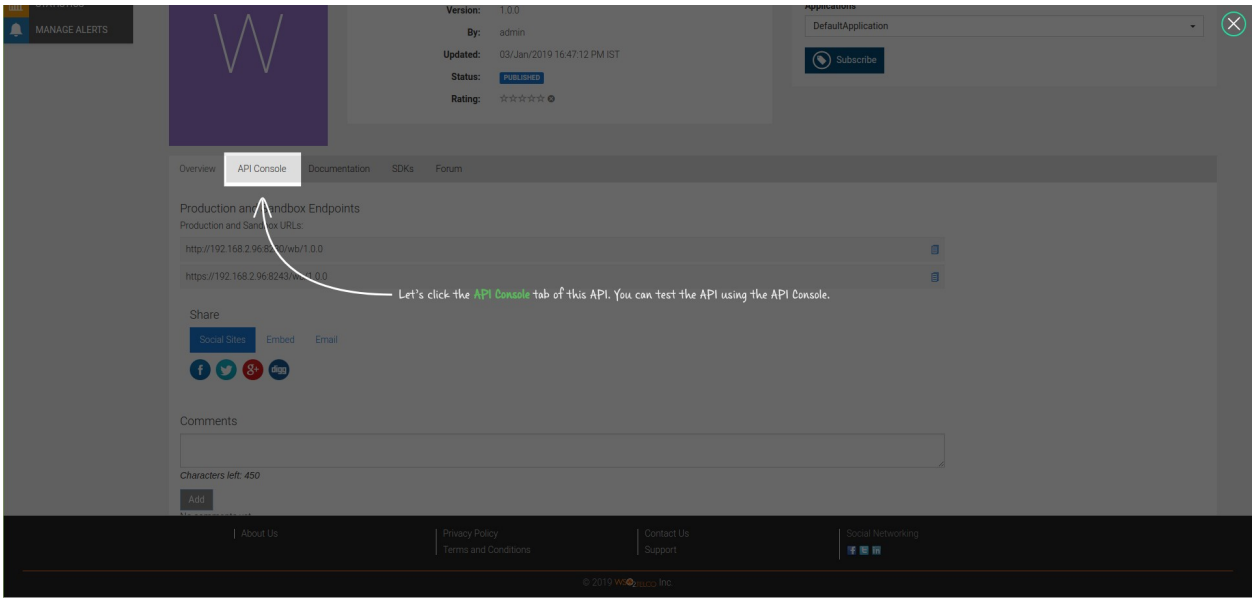
STORE
Invoking APIs

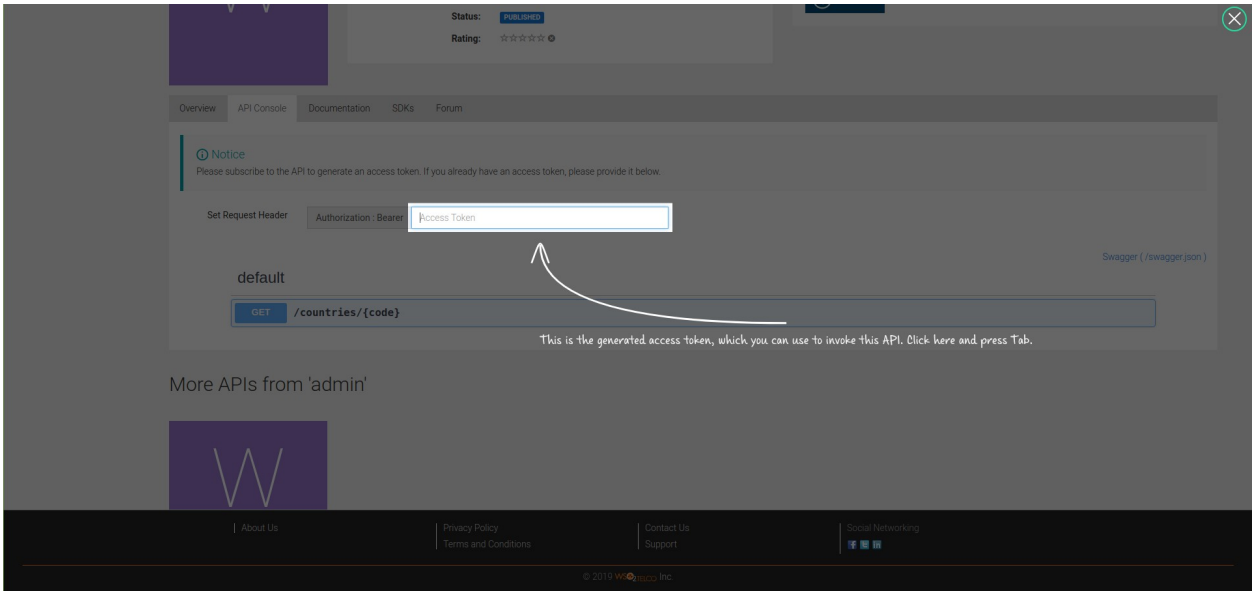





FORUM
STATISTICS
MANAGE ALERTS
Version: 1.0.0
By: admin
Updated: 03/Jan/2019 16:47:12 PM IST
Status: PUBLISHED
Rating:
Applications
DefaultApplication
Subscribe
Overview
API Console
Documentation
SDKs
Forum
Production and Sandbox Endpoints
Production and Sandbox URLs:
http://192.168.2.96:8280/wb/1.0.0
https://192.168.2.96:8243/wb/1.0.0
Share
Social Sites
Embed
Email
Comments
About Us
Privacy Policy
Terms and Conditions
Contact Us
Support
Social Networking
Let's subscribe to this API using DefaultApplication, which is already selected for you.
Click Subscribe.
An application is a logical collection of APIs. Applications allow you to use a single access token to invoke a collection of APIs and to subscribe to one API multiple times with different SLA levels.


HOME
APPLICATIONS
FORUM
STATISTICS
MANAGE ALERTS
WorldBank - 1.0.0
Version: 1.0.0
By: admin
Updated: 03/Jan/2019 16:47:12 PM IST
Status: PUBLISHED
Rating:
Applications
DefaultApplication
Subscribe
Overview
API Console
Documentation
SDKs
Forum
Production and Sandbox Endpoints
Production and Sandbox URLs:
http://192.168.2.96:8280/wb/1.0.0
https://192.168.2.96:8243/wb/1.0.0
Share
Social Sites
Embed
Email
About Us
Privacy Policy
Terms and Conditions
Contact Us
Support
Social Networking
Now you have subscribed this Application to this selected API. Click View Subscriptions button to get more information about the Application Subscriptions.
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Audits



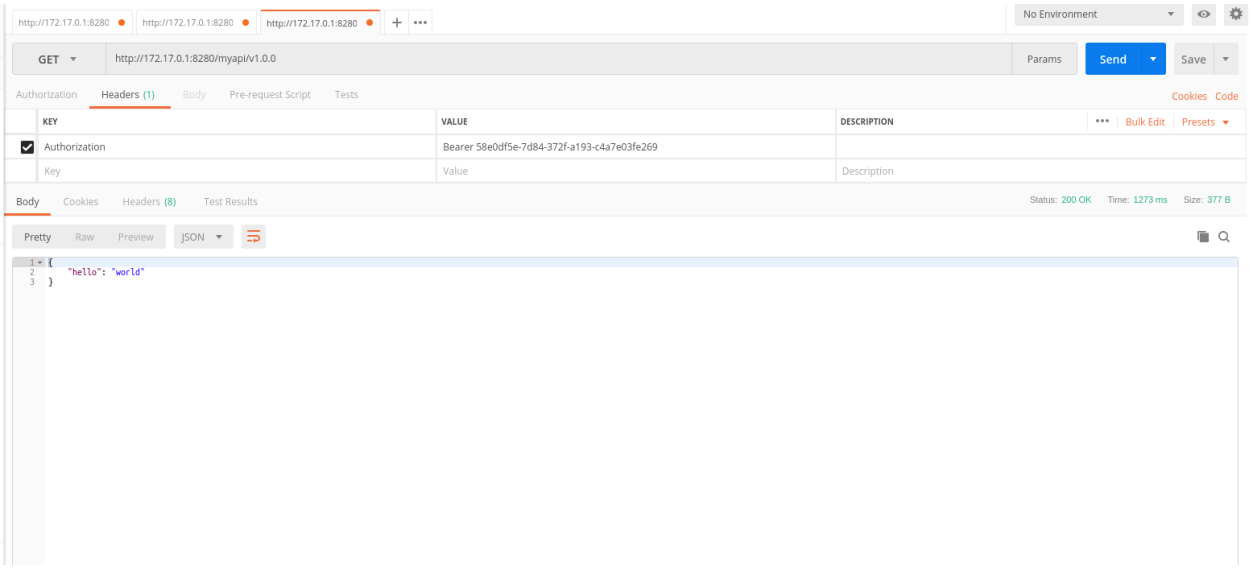


Subscription Approval

Go to https://192.168.2.96:9443/manage-service
Login with publisher credentials
Approve the subscription

GO to the created application
Go to production keys
Generate a token

Check it with postman



Setting up IS

Step 1 - Download the prepackaged WSO2 IS as the Key Manager
Download the prepackaged WSO2 IS as the Key Manager from the location: https://product-dist.wso2.com/products/api-manager/2.5.0/wum-updated-packs/wso2is-km-5.6.0.zip

Step 2 - Optionally, configure port offset for WSO2 IS
Open the <IS_HOME>/repository/conf/carbon.xml file and change the offset to 1.
<Offset>1</Offset>

Step 3 - Install and configure the databases
Since we have already created the databases, we need to only configure the data source configurations. = add the data

Step 4 - Configure the Key Manager (WSO2 IS) with WSO2 API-M
1. Open the <IS_HOME>/repository/conf/api-manager.xml [<ServerURL> element that appears under the <APIGateway> section.]

```
<ServerURL>https://localhost:8243/services/</ServerURL>
<RevokeAPIURL>https://localhost:8243/revoke</RevokeAPIURL>
```

2. Change the datasource in the <IS_HOME>/repository/conf/user-mgt.xml file to point to the WSO2UM_DB datasource

```
<UserManager>
   <Realm>
      <Configuration>
         ...
         <Property name="dataSource">jdbc/WSO2UM_DB</Property>
      </Configuration>
      ...
   </Realm>
</UserManager>
```

3. Change the datasource in the <IS_HOME>/repository/conf/identity/identity.xml file to point to the WSO2AM_DB datasource.

```
<JDBCPersistenceManager>
   <DataSource>
      <Name>jdbc/WSO2AM_DB</Name>
   </DataSource>
   ...
 <JDBCPersistenceManager>
```

Step 5 - Configure WSO2 API-M with the Key Manager (WSO2 IS)

1.Change the ServerURL of the **AuthManager** and the ServerURL of the **APIKeyValidator** to point to WSO2 IS in the <API-M_HOME>/repository/conf/api-manager.xml file.

<AuthManager>
<ServerURL>https://localhost:9444/services/</ServerURL>
</AuthManager>

<APIKeyValidator>
<ServerURL>https://localhost:9444/services/</ServerURL>
</APIKeyValidator>

2. Enable WS Client and disable the Thrift Client.

<APIKeyValidator>
<KeyValidatorClientType>WSClient</KeyValidatorClientType>
<EnableThriftServer>false</EnableThriftServer>
</APIKeyValidator>

3. Change the data source name in **JDBCPersistenceManager** is jdbc/WSO2AM_DB in the <API-M_HOME>/repository/conf/identity/identity.xml file.

<JDBCPersistenceManager>
<DataSource>
<Name>jdbc/WSO2AM_DB</Name>
</DataSource>
<JDBCPersistenceManager

Step 6 - Configuring the Databases for IS as the Key Manager
1.Add the mysql-connector-java-5.1.44.tar.gz mysql connector jar into {IS_HOME}/repository/components/lib directory

2. Copy the db configurations related to WSO2AM_DB, WSO2UM_DB, WSO2REG_DB from <API-M_HOME>/../datasource.xml to that of <IS_HOME>/../datasource.xml

3. Change below in the <IS_HOME>/repository/conf/user-mgt.xml file.
a)
<configuration>
<Property name="dataSource">jdbc/WSO2UM_DB</Property>
</configuration>

b)
Add the user store configuration correctly so that both the WSO2 Identity Server and WSO2 API Manager server point to the same user store.

```
<UserStoreManager class="org.wso2.carbon.user.core.jdbc.JDBCUserStoreManager">
                                                                    <Property
name="TenantManager">org.wso2.carbon.user.core.tenant.JDBCTenantManager</Property>
                                        <Property name="ReadOnly">false</Property>
                                <Property name="MaxUserNameListLength">100</Property>
                                    <Property name="IsEmailUserName">false</Property>
                                  <Property name="DomainCalculation">default</Property>
                                  <Property name="PasswordDigest">SHA-256</Property>
                                  <Property name="StoreSaltedPassword">true</Property>
                                        <Property name="ReadGroups">true</Property>
                                        <Property name="WriteGroups">true</Property>
                        <Property name="UserNameUniqueAcrossTenants">false</Property>
                            <Property name="PasswordJavaRegEx">^[\S]{5,30}$</Property>
                        <Property name="PasswordJavaScriptRegEx">^[\S]{5,30}$</Property>
    <Property name="UsernameJavaRegEx">^[^~!#$;%^*+={}\\|\\\\&lt;&gt;,\'\"]{3,30}$</Property>
                        <Property name="UsernameJavaScriptRegEx">^[\S]{3,30}$</Property>
    <Property name="RolenameJavaRegEx">^[^~!#$;%^*+={}\\|\\\\&lt;&gt;,\'\"]{3,30}$</Property>
                        <Property name="RolenameJavaScriptRegEx">^[\S]{3,30}$</Property>
                                <Property name="UserRolesCacheEnabled">true</Property>
                                <Property name="MaxRoleNameListLength">100</Property>
                                <Property name="MaxUserNameListLength">100</Property>
                                  <Property name="SharedGroupEnabled">false</Property>
                                        <Property name="SCIMEnabled">false</Property>
</UserStoreManager>
```

4.To enable the Key Manager to access to the registry database, open the **<IS_HOME>/repository/conf/registry.xml** file in the Key Manager node and add or modify the dataSource attribute of the <dbConfig name="govregistry"> element as follows in order to mount the Key Manager to the governance registry space. Do the same for that of **api-manager.xml file of <API-M_HOME>**

Comment following in registry.xml

```
 <!-- <dbConfig name="sharedregistry">
      <dataSource>jdbc/WSO2REG_DB</dataSource>
   </dbConfig> -->
```

```
  <!--  <dbConfig name="govregistry">
      <dataSource>jdbc/WSO2REG_DB</dataSource>
   </dbConfig> -->

 <!--  <remoteInstance url="https://localhost:9443/registry">
       <id>mount</id>
       <dbConfig>sharedregistry</dbConfig>
       <readOnly>false</readOnly>
       <enableCache>true</enableCache>
       <registryRoot>/</registryRoot>
   </remoteInstance> -->
```

Add follwing

```
<dbConfig name="govregistry">
  <dataSource>jdbc/WSO2REG_DB</dataSource>
</dbConfig>
<remoteInstance url="https://localhost:9443/registry">
   <id>gov</id>
   <cacheId>user@jdbc:mysql://db.mysql-wso2.com:3306/regdb</cacheId>
   <dbConfig>govregistry</dbConfig>
   <readOnly>false</readOnly>
   <enableCache>true</enableCache>
   <registryRoot>/</registryRoot>
</remoteInstance>
<mount path="/_system/governance" overwrite="true">
   <instanceId>gov</instanceId>
   <targetPath>/_system/governance</targetPath>
</mount>
<mount path="/_system/config" overwrite="true">
   <instanceId>gov</instanceId>
   <targetPath>/_system/config</targetPath>
</mount>
```

Step 7: Start the IS

References :

- https://docs.wso2.com/display/AM250/Configuring+the+Databases+for+IS+as+the+Key+Manager

- https://docs.wso2.com/display/AM250/Configuring+WSO2+Identity+Server+as+a+Key+Manager#ConfiguringWSO2IdentityServerasaKeyManager-Step1-DownloadtheprepackagedWSO2ISastheKeyManager

Step 8: change URLs as
<http uri-template="https://localhost:9444/oauth2/authorize">
In /wso2am-2.5.0/repository/deployment/server/synapse-configs/default/api
- _AuthorizeAPI_.xml
- _RevokeAPI_.xml
- _TokenAPI_.xml

Component-dep

Add jaggery changes to component-dep

Product-hub

Managing Throttling

Throttling allows you to limit the number of successful hits to an API during a given period of time,for situations like below.

- To protect your APIs from common types of security attacks such as denial of service (DOS)

- To regulate traffic according to infrastructure availability.

There are 5 main throttling policies implemented in wso2AM Admin portal.

1. Advanced policies
2. Blacklist policies
3. Custom policies
4. Application policies
5. Subscription policies

Policy 1 : Advanced policies

Advanced Throttling : Advanced throttling policies allow an API Publisher to control access to his API resources using advanced rules. Advanced policies include the ability to apply limits by filtering requests based on the following properties and their combinations.

1. IP address and address range
2. HTTP request headers
3. JWT claims
4. Query parameters

Enable advanced Throttling in api-manager.xml

<EnableAdvanceThrottling>true</EnableAdvanceThrottling>

1. **IP address and address range**

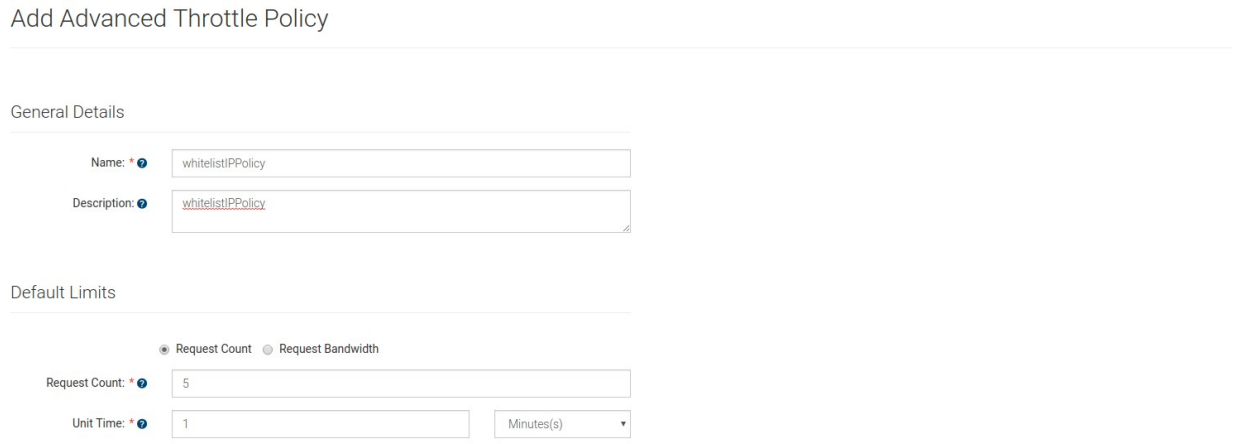
Control/restrict access to your API or its selected resources for a given IP address or address range
Ex:- IP Whitelisting (have described below)

IP Whitelisting

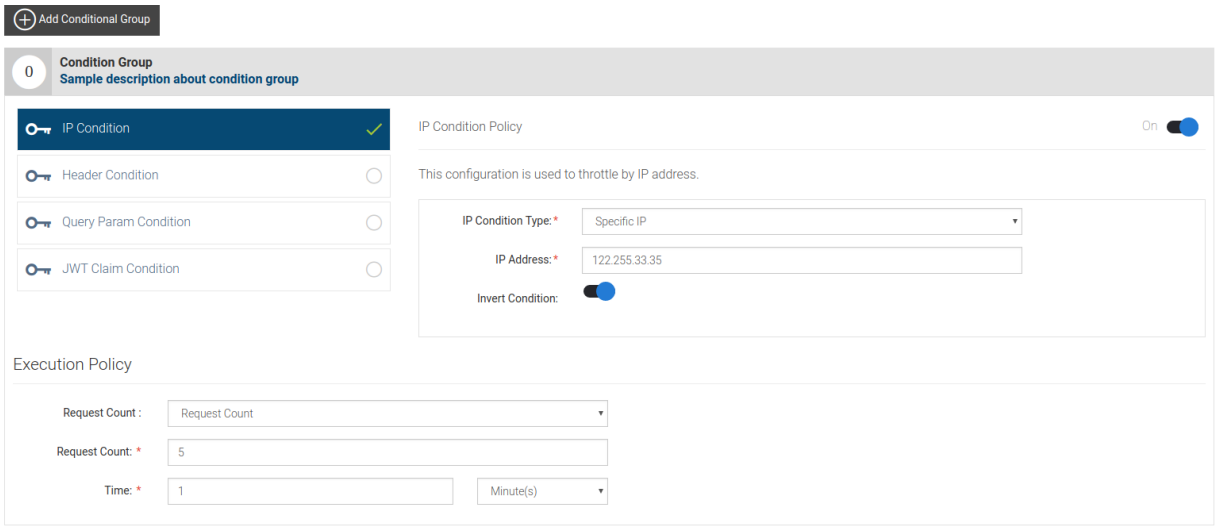
Way of configuring a filter to extract a particular set of known IP addresses and grant the access to the given assets for requests comes from those IPs only.

1.Login to the admin portal of WSO2 API Manager](https://<ip_address>:9443/admin).
2.Open Throttling Policies tab and navigate to Advanced Throttling.
3.Click ADD NEW POLICY to add a new Throttling tier.

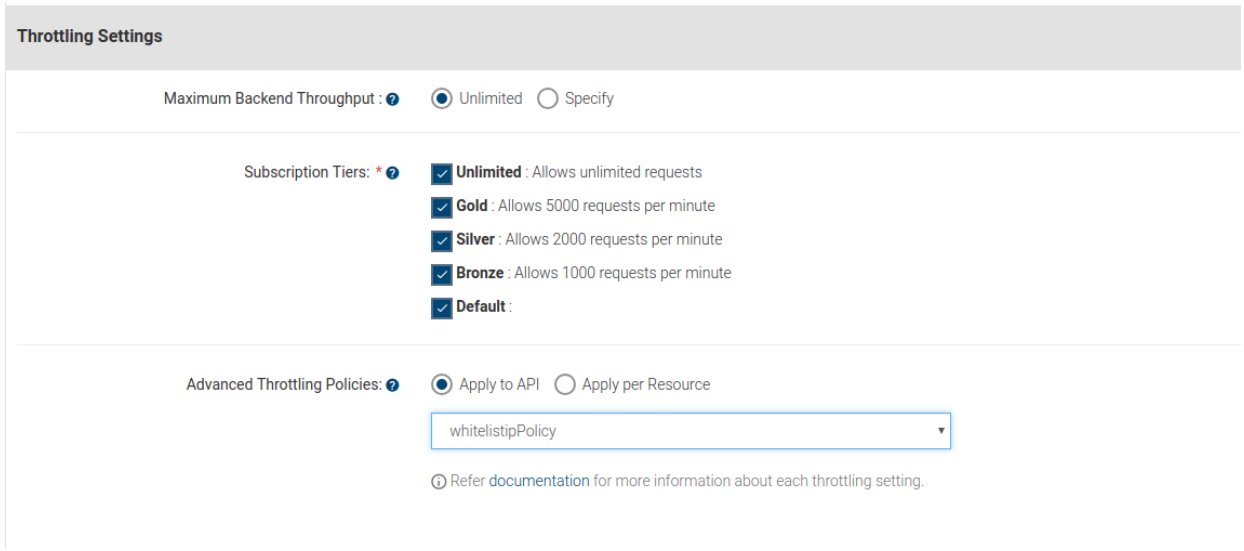
Creating the Advanced Throttling policy



4. Open the Conditional Group added and fill the details.

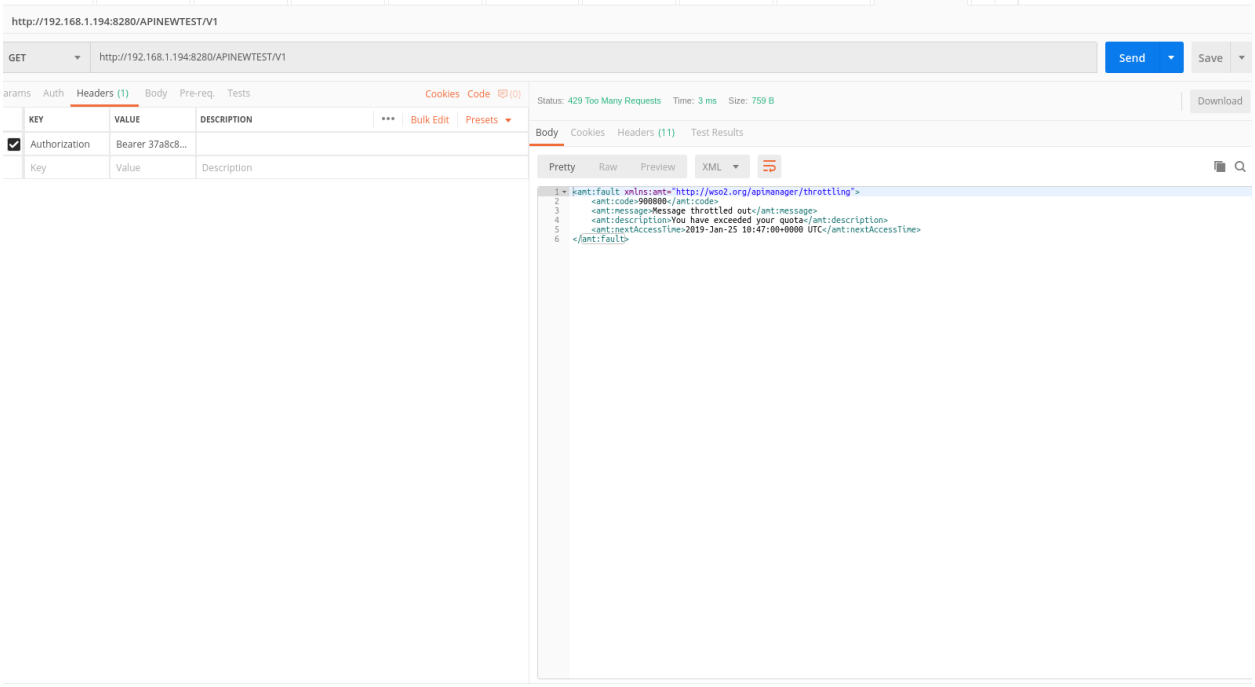


5. Create a new API and select advanced throttling policies and select the policy created.



6. Subscribe in to a previously created application.

7.Check the workflow by sending a request.



Since we set the request per minute as 5, it will set a throttling error as follows.

```
<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:nextAccessTime>2019-Jan-25 10:47:00+0000 UTC</amt:nextAccessTime>
</amt:fault>

If we changed the network/ip we are able to send as much as requests.

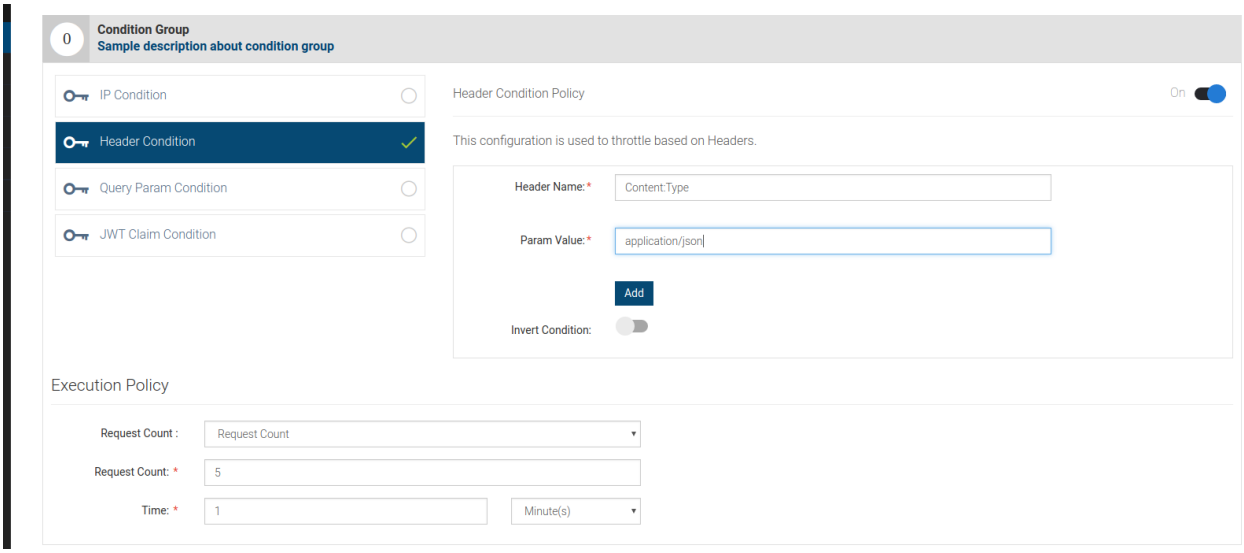
2. **HTTP request headers**

Allow you to apply limits to APIs by filtering requests based on HTTP headers.

Enable HeaderConditions in api-manager.xml

<EnableHeaderConditions>true</EnableHeaderConditions>

Add Advanced Throttle Policy with following condition group for an example.



Here we need to apply a special limit for JSON requests.There we can filter JSON messages by using a policy that inspects the HTTP request headers and checks if the Content-Type header is application/json and apply a special limit for those requests while allowing a default value for the rest.

When we send requests of type application/json, following throttle out error comes when requests are greater than 5.
code=900800&message=Message%20throttled%20out&description=You%20have%20exceeded%20your%20quota&nextAccessTime=2019-Jan-28%2012%3A23%3A00%2B0000%20UTC

3. **JWT claims**

A JWT claim contains meta information of an API request. It can include application details, API details, user claims. Advanced throttling policies based on JWT claims allow you to filter requests by JWT claim values and apply limits for requests.

Enable EnableJWTClaimConditions in api-manager.xml

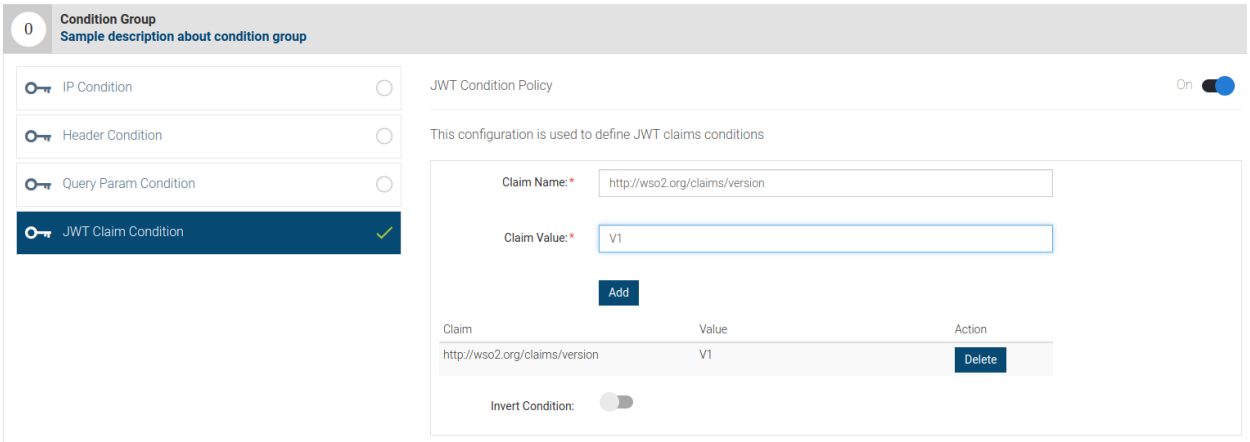
<EnableJWTClaimConditions>true</EnableJWTClaimConditions>

Generate a JWT Token

JWT PAYLOAD:DATA


```
{
 "http://wso2.org/claims/applicationtier": "100PerMin",
 "http://wso2.org/claims/keytype": "PRODUCTION",
 "http://wso2.org/claims/version": "V1",
 "iss": "wso2.org/products/am",
 "http://wso2.org/claims/applicationname": "newapp",
 "http://wso2.org/claims/enduser": "ChamaniNew@carbon.super",
 "http://wso2.org/claims/enduserTenantId": "-1234",
 "http://wso2.org/claims/subscriber": "ChamaniNew",
 "http://wso2.org/claims/tier": "Unlimited",
 "exp": 1548748400,
 "http://wso2.org/claims/applicationid": "9",
 "http://wso2.org/claims/usertype": "APPLICATION",
 "http://wso2.org/claims/apicontext": "/NEWAPI1/V1"
}
```


Use above claim names and claim values to generate a restriction.



When the limit(5) exceed it will give below error.
code=900800&message=Message%20throttled%20out&description=You%20have%20exceeded%20your%20quota&nextAccessTime=2019-Jan-29%2008%3A19%3A00%2B0000%20UTC

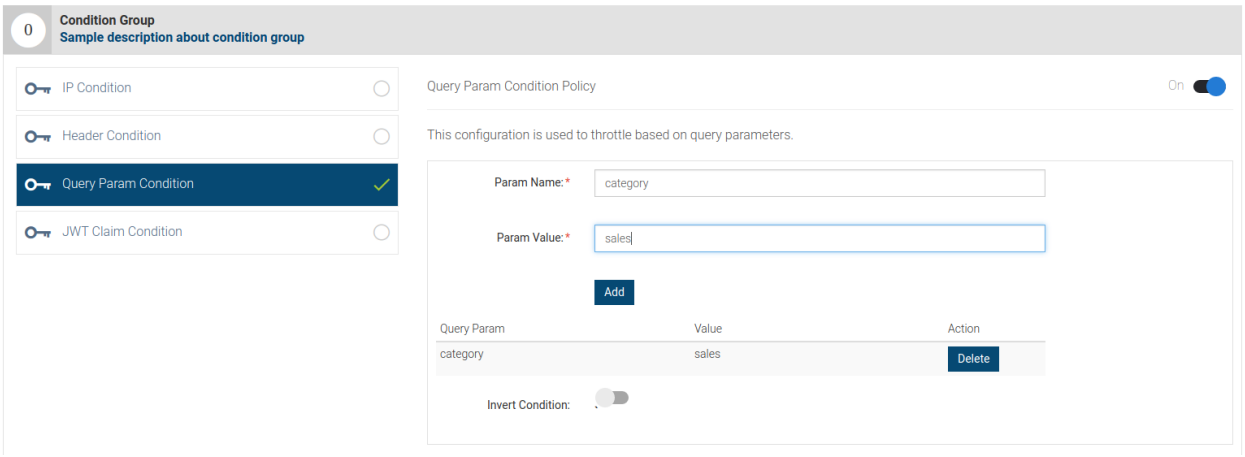
4. **Query parameters**

Filtering based on query parameters almost always apply to HTTP GET requests when doing search type of operations.

Enable QueryParamConditions in api-manager.xml

<EnableQueryParamConditions>true</EnableQueryParamConditions>

Ex: Search API
We can have Param Name : category, Param value: sales



Send a request as :
http://192.168.1.194:8280/NEWAPI1/V1?category=sales

Policy 2 : Blacklist policies

By blacklisting requests, we can protect servers from common attacks and abuse by users. For example, if a malicious user misuses the system, all requests received from that particular user can be completely blocked.

- Block calls to specific APIs
- Block all calls from a given application

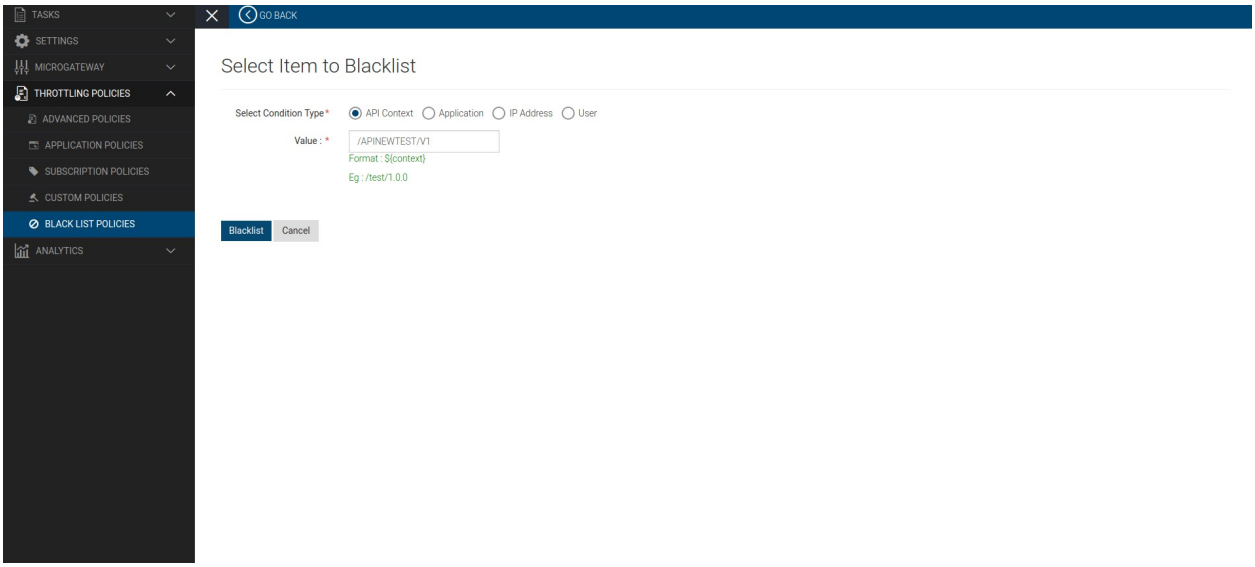
- Block requests coming from a specific IP address
- Block a specific user from accessing APIs

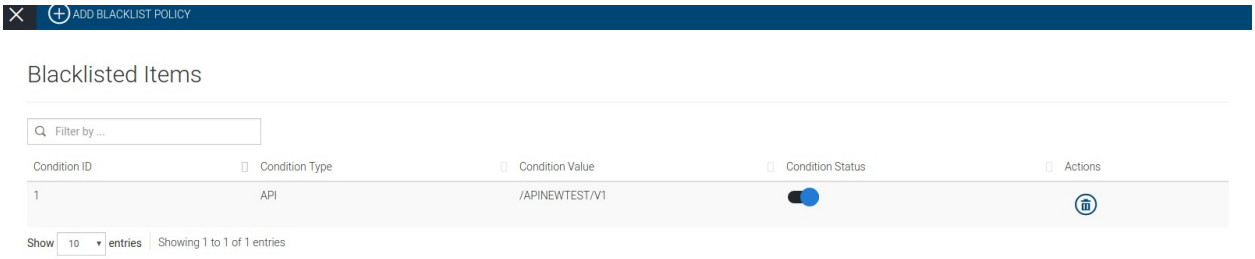
1. Log in to the Admin Portal using the URL https://localhost:9443/admin and your admin credentials.
2. Click BlackList under the Throttle Policies section and click Add Item.

Here we can black list API Context / API / IP address or User.

- **Blocking a API Context**

Click **API Context**
Add the resource as **/APINAME/VERSION**
Click Blacklist



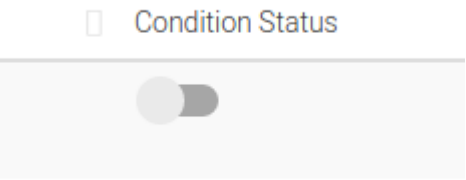


3.Login to API Store using the URL https://localhost:9443/store and invoke the API.
You will see the following response

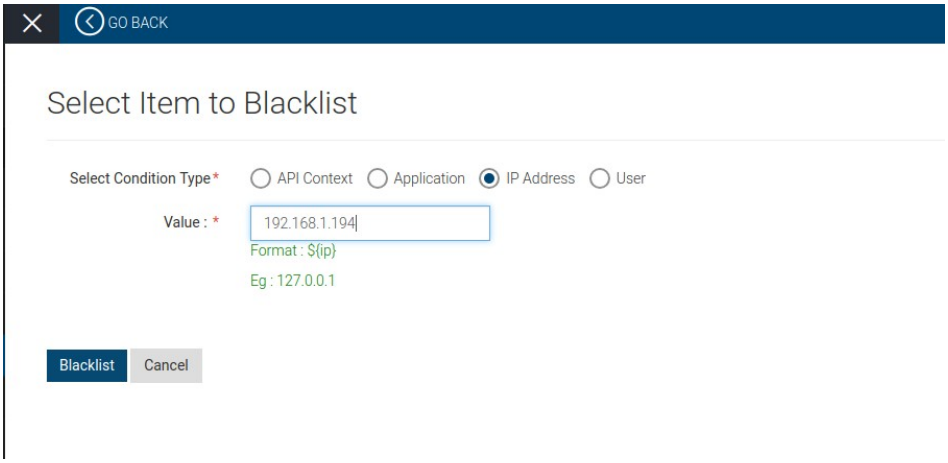
4.Check the workflow by sending a request.
It will display as follows.
<amt:fault xmlns:amt="http://wso2.org/apimanager/throttling">

```
    <amt:code>900805</amt:code>
    <amt:message>Message blocked</amt:message>
    <amt:description>You have been blocked from accessing the resource</amt:description>
</amt:fault>
```

To disable this untick **condition status**.



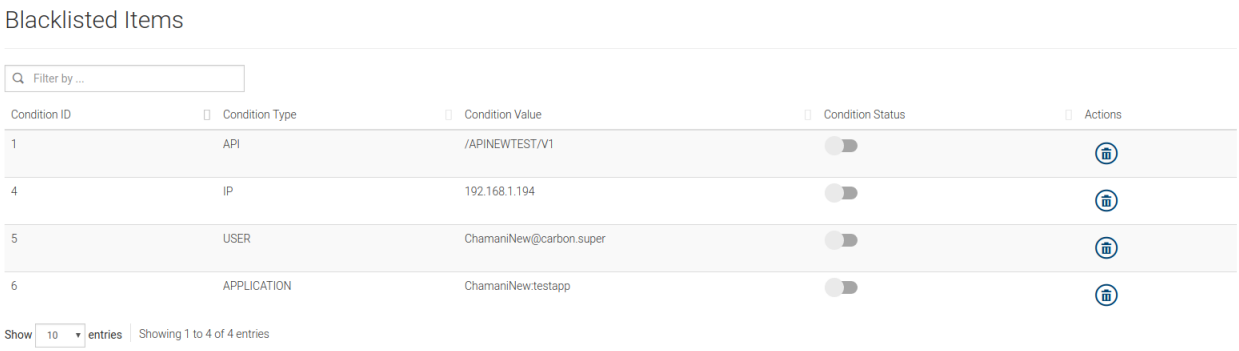
- **Blocking an IP address**



4.Check the workflow by sending a request.
It will display as follows.

```
<amt:fault xmlns:amt="http://wso2.org/apimanager/throttling">
    <amt:code>900805</amt:code>
    <amt:message>Message blocked</amt:message>
    <amt:description>You have been blocked from accessing the resource</amt:description>
</amt:fault>
```

Likewise can blacklist application and user also.



Policy 3 : Custom policies

- Custom throttling allows system administrators to define dynamic rules for specific use cases, which are applied globally across all tenants.
- uses the Siddhi query language

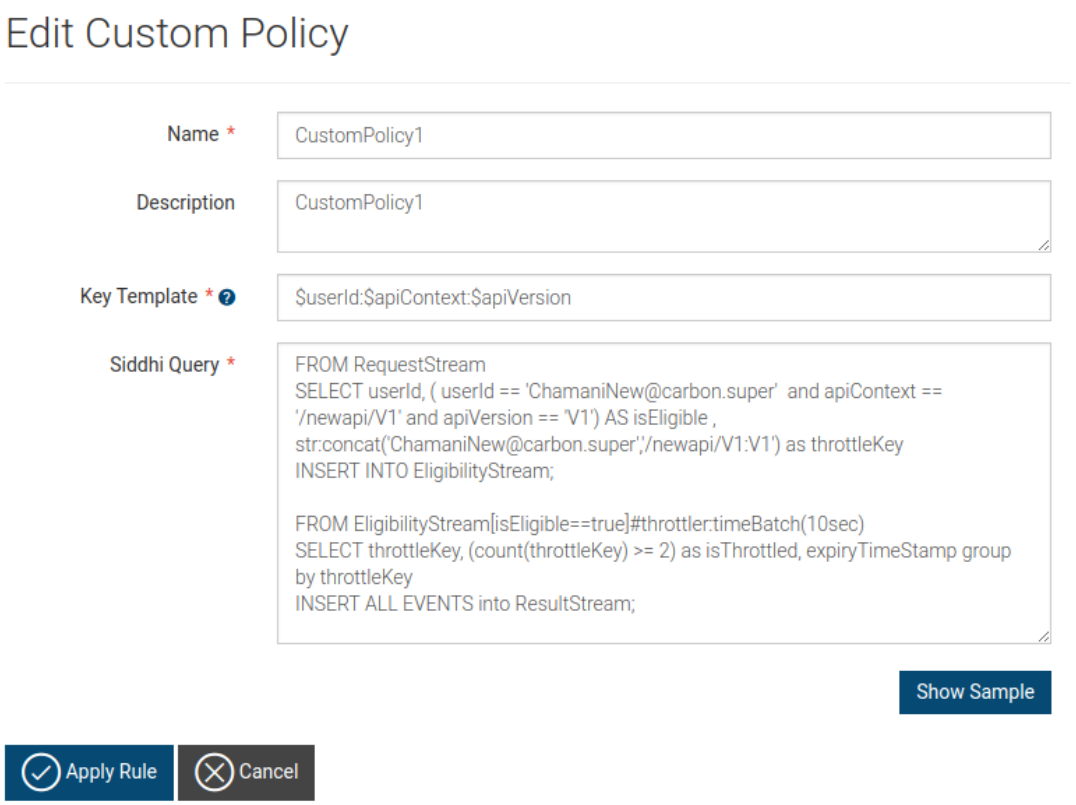
The following keys can be used to create custom throttling policies:
resourceKey, userId, apiContext, apiVersion, appTenant, apiTenant, appId

Key template:
$userId:$apiContext:$apiVersion

Siddhi Query:

```
FROM RequestStream
SELECT userId, ( userId == 'ChamaniNew@carbon.super'  and apiContext == '/newapi/V1' and apiVersion == 'V1') AS isEligible ,
str:concat('ChamaniNew@carbon.super','/newapi/V1:V1') as throttleKey
INSERT INTO EligibilityStream;

FROM EligibilityStream[isEligible==true]#throttler:timeBatch(15sec)
SELECT throttleKey, (count(throttleKey) >= 5) as isThrottled, expiryTimeStamp group by throttleKey
INSERT ALL EVENTS into ResultStream;
```



Here in each 10 seconds when we send requests, 2 will be given 200 response while others will give 429 responses.

Error logs as this :

[2019-01-28 16:24:29,431] INFO - ConnectionStartOkMethodHandler SASL Mechanism selected: PLAIN
[2019-01-28 16:24:29,431] INFO - ConnectionStartOkMethodHandler Locale selected: en_US
[2019-01-28 16:24:29,433] INFO - ConnectionStartOkMethodHandler Connected as: admin
[2019-01-28 16:24:29,433] INFO - ConnectionStartOkMethodHandler Framesize set to 65535
[2019-01-28 16:24:29,455] INFO - ChannelOpenHandler Connecting to: carbon
[2019-01-28 16:24:29,455] INFO - AndesChannel Channel created (ID: 127.0.0.1:48480)

Policy 4 : Application policies
Policy 5 : Subscription policies

Application-level throttling tiers are defined at the time an application is created.

Note: If error comes when creating application as “Specified application tier not exists”, go to admin portal and “Application Throttling Policies” and add Default tier.

Adding application level tiers and subscription level tiers

1. Sign in to the Admin Portal using the URL `https://localhost:9443/admin` and your admin credentials (admin/admin by default).
2. Click Application Tiers under the Throttle Policies section to see the set of existing throttling tiers.
3. To add a new tier, click Add New Policy. (Ex: 100PerMin)

Similarly go to Subscription Tiers and add a new policy. (Ex: Platinum)

Then go to manage module and select the **created application tier** when approving the application.

Similarly when subscribing the application to api select the **created subscription tier.**

Send requests and check if the throttling policies work as expected.