Exercise 12

API Management and Governance including Business Activity Monitoring

Prior Knowledge
RESTful services
Previous ESB exercises

Objectives
Understand API management and key issuing. Understand Business Activity Monitoring. Be able to configure the API Manager and Business Activity Monitor, and use OAuth2 Bearer Tokens

Software Requirements
Oracle Java 1.7.0_67
WSO2 API Manager 1.7.0 (WSO2 AM)
WSO2 Business Activity Monitor 2.4.1 (WSO2 BAM)
Node.js and npm

The installation process was to follow the instructions here on linking the two products:
https://docs.wso2.com/display/AM170/Publishing+API+Runtime+Statistics

1) We have a simple backend service written in node.js. If you want to use a previous ESB or JAX-RS service you have created you can do that, but these instructions are based on the node service.

   This is super-simple backend node.js API that just returns "hello world":

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200);
  res.end("hello world\n");
}).listen(8001);
http.createServer(function (req, res) {
  res.writeHead(200);
  res.end("hello sandbox\n");
}).listen(8002);
```

   This is available in the VM at /home/ox-soa/nodeserver/server.js or at this URL:
   http://freo.me/1yLN9kM

2) Start the node service in a fresh terminal window by:
   a. `cd ~/nodeserver`
   b. `nodejs server.js`
   c. Test the service by browsing http://localhost:8001/ and http://localhost:8002/

3) Start the WSO2 BAM Server:
   a. `cd ~/servers/wso2bam-2.4.1/`
   b. `bin/wso2server.sh`

4) Start the WSO2 API Manager:
   a. `cd ~/servers/wso2am-1.7.0`
   b. `bin/wso2server.sh`

5) Wait until both are started and then check that you can access the admin screens:
   a. https://localhost:9447/carbon (AM)

b. https://localhost:9448 (BAM)
Because we have not got a "real" TLS/SSL certificate, your browser will complain about these websites. You will need to persuade your browser to move on! (Browser dependent).

6) To create the users and roles in the API Manager, you log in to the management console as an administration user (default credentials: **admin/admin**).

To speed things up we have already created the following users and roles.

Username	Role	Password
charlie	creator	password
peter	publisher	password

To see what we did go look at the Appendix A.

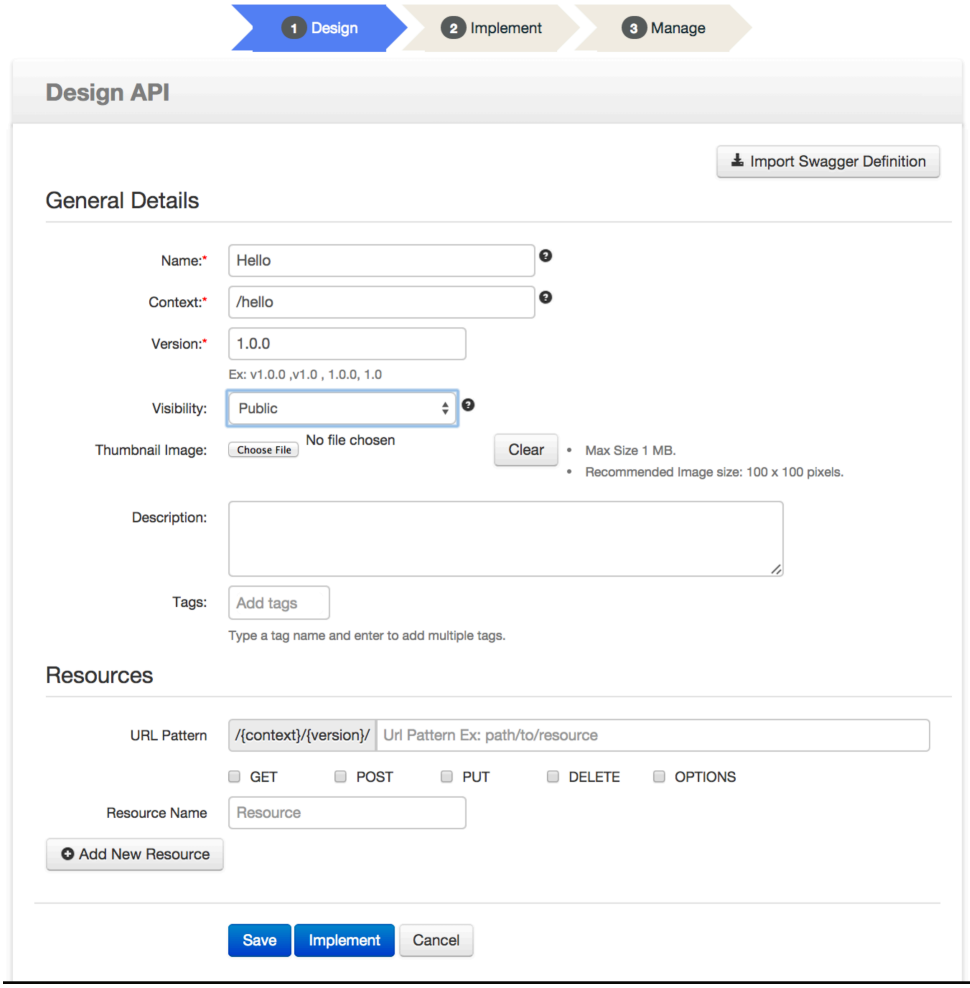
7) An API creator uses the API provider Web application to create and publish APIs into the API Store. In this section, we explain how to create an API and attach documentation to it.

In this guide, we work with a service exposed by the node.js server running on port 8001 on the VM. Let's create this API and add it to the API Store.

Open the API Publisher system (https://localhost:9447/publisher) and log in as **charlie/password**

8) Click the **Add** link and provide the information given in the table below.

Field	Value	Description
Name	Hello	Name of API as you want it to appear in the API store
Context	/hello	URI context path that is used by to API consumers
Version	1.0.0	API version (in the form of version.major.minor)
Visibility	Public	You can require users to be authorized into a role or domain before they can see this API. We are making it fully visible: even to users who are not signed in.
Thumbnail Image	Your choice or leave blank	I like kittens :-)
Description	Up to you	
Tags	Again up to you	
Resources	Leave for the moment	This area allows you to define specific RESTful resources which can then have permissions applied, improved documentation, etc



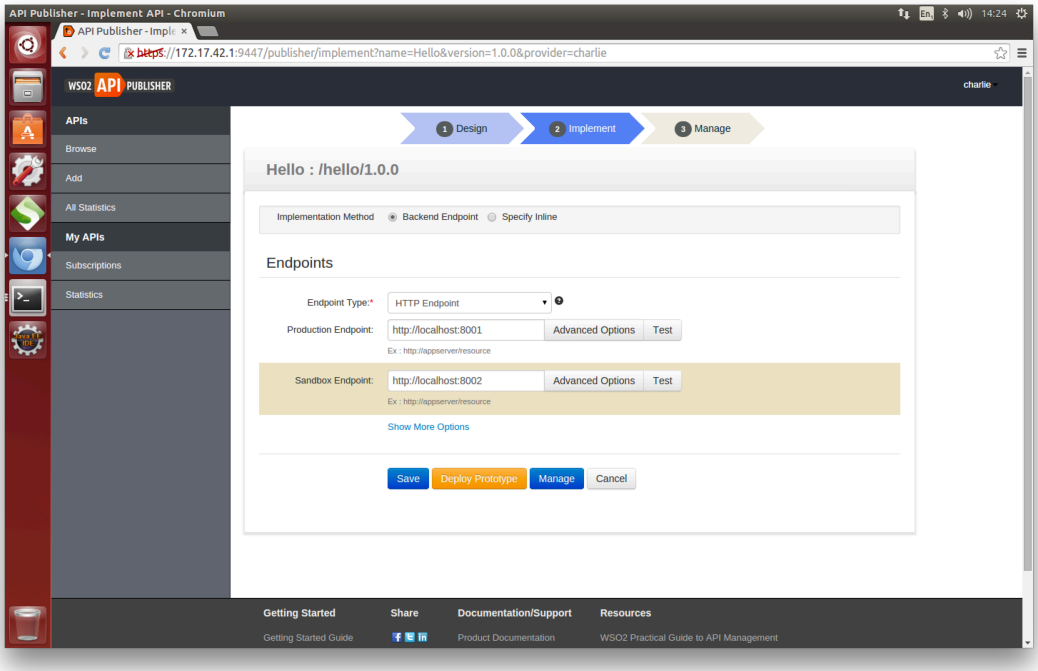
9) Click **Implement**.

10) It asks you to create a resource with wildcard characters (/*). Click **Yes**

Note that a resource by the name default gets created as follows.

10) Click **Implement** again to go to the Implement tab and provide the following information.

Field	Value	Description
Implementation method	Backend endpoint	If you have a real backend implementation to your API, select that option. Else, you can specify implementation in-line. The latter approach is usually used in mock-up implementation for prototyped APIs.
Endpoint type	HTTP endpoint	
Production endpoint	http://localhost:8001/	This is the URL of the Node.js service
Sandbox endpoint	http://localhost:8002/	This is the URL of the Node.js sandbox service



11) Now Click **Manage** to go to the Manage tab and provide the following information:

Field	Value	Description
Default Version	Ticked	There can be a default version of every API, which can be routed to whichever version the administrator or API owner chooses.
Tier Availability	Select all of 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.
Transports	HTTP/HTTPS	This allows users to use HTTP to access this URL. In practice this is a bad idea for a production system because the API Key (Bearer Token - which we'll see later) needs to be protected.

11) Now click **Save.**

*If you try **Save and Publish** you won't succeed because you are only a creator not a publisher! If you do try this, check the WSO2 AM logs in the terminal window.*

12) On the left hand menu click APIs->Browse.

13) Have a look at the API

14) Adding Documentation: After creating the API, click on its icon to open its details. Select the Docs tab.

15) Click **Add New Document** link.

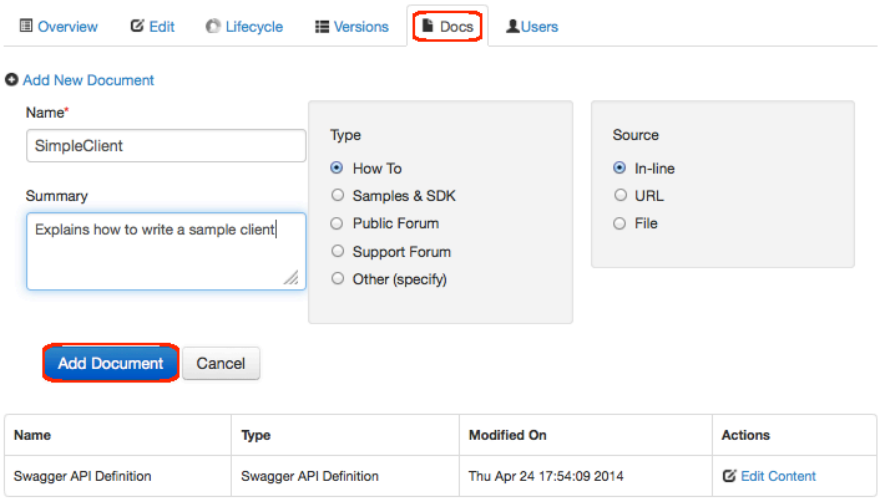
Documentation can be provided inline, via a URL or as a file. For inline documentation, you can edit the content directly from the API publisher interface. You get several documents types:

1. Swagger documents
2. How To
3. Samples and SDK

4. Public forum / Support forum (external link only)
5. API message formats
6. Other

16) Select the **How To** type, a name for the document and a short description, which will appear in the API Store. Select inline or provide a URL.

17) Click **Add Document**.



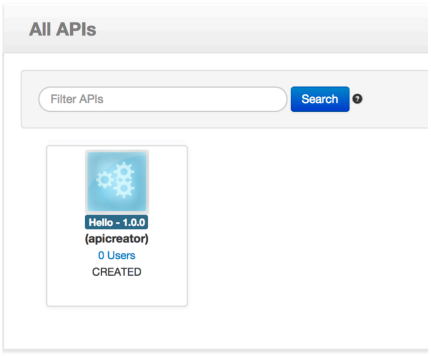
18) Once the document is added, click **Edit Content** link, which opens an embedded editor to edit the document contents.

Publishing!

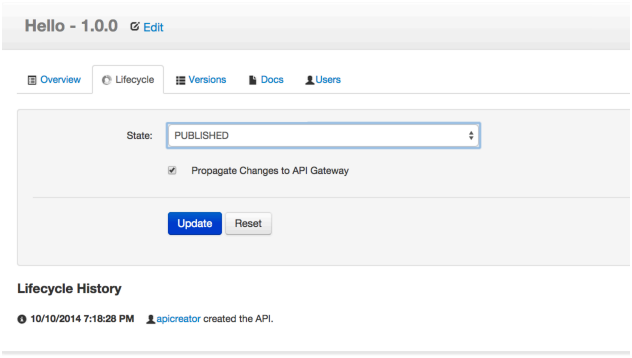
19) Log out as **Charlie**

20) Log in as **peter/password**

21) Now Click on the Hello Icon



22) Choose the **Lifecycle** tab. Select **Published**. Click **Update**



Subscription

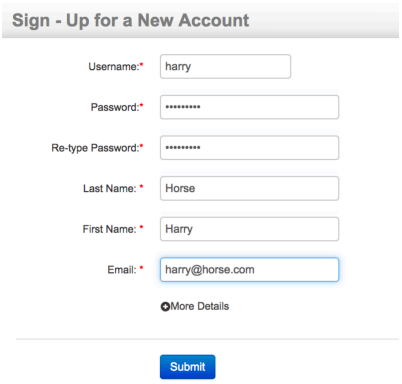
23) You subscribe to APIs using the API Store Web application

    Open the API Store (https://localhost:9447/store) using your browser. Using the API Store, you can,

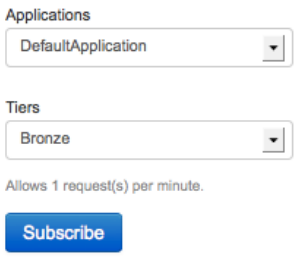
    - Search and browse APIs
    - Read documentation
    - Subscribe to APIs
    - Comment on, rate and share/advertize APIs
    - Take part in forums and request features etc.

The API you published earlier is available in the API Store.

24) Self sign up to the API Store using the **Sign-up** link. You can use any appropriate (or even inappropriate) details.



25) After signup, log in to the API Store and click the API that "peter" published earlier (Hello 1.0.0).
26) Note that you can see the subscription option in the right hand side of the UI after logging in. Select the default application, Bronze or Silver tier and click **Subscribe**.



Applications
An application is a logical collection of one or more APIs, and is required when subscribing to an API.

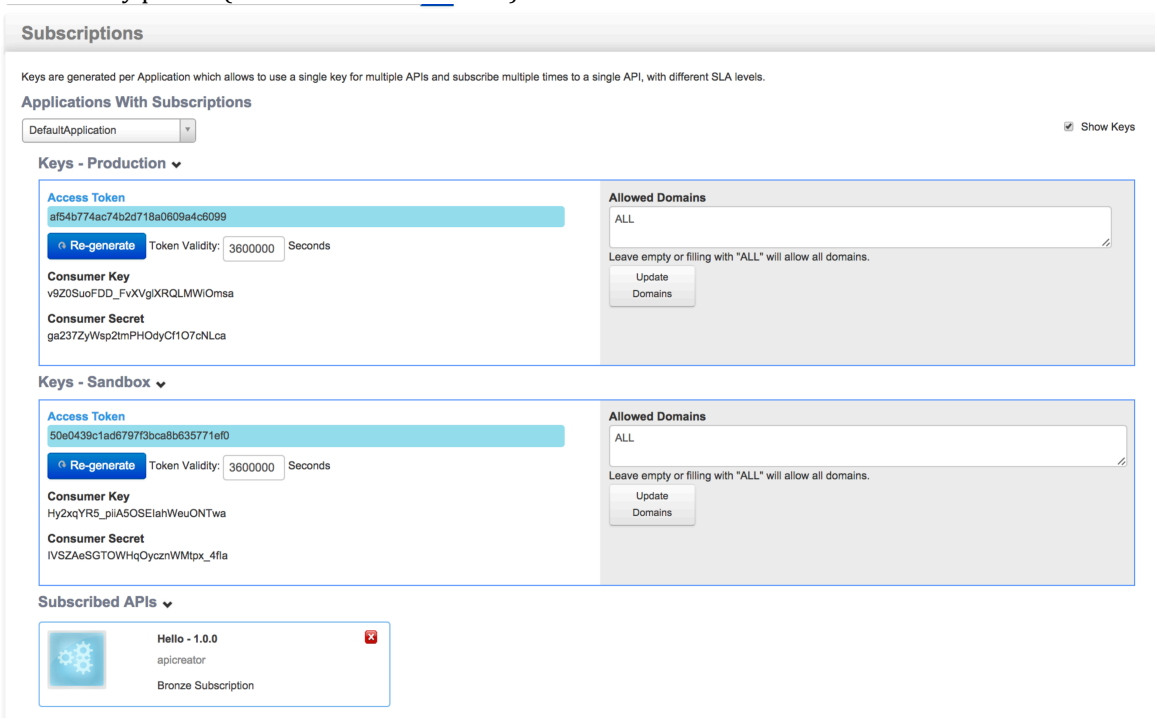
It represents your client application.

You can subscribe to multiple APIs using the same application. Instead of using the default application, you can also create your own by selecting the **New Application...** option in the above drop-down list or by going to the **My Applications** menu in the top menu bar.

27) Once the subscription is successful, go to **My Subscriptions** page.

    You can see that you are subscribed to the API.

28) Click Generate to create both a production and sandbox key. I recommend extending the validity period (add a few 00s to the end).



You can now test your API.

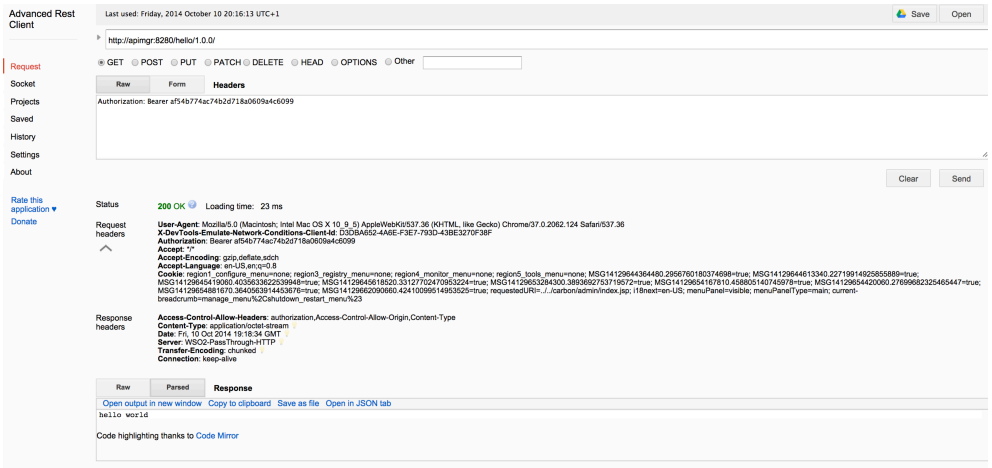
29) There are a few ways you can try this out. We like the Chrome Advanced Rest Client, and curl.

    You need to craft an HTTP GET request against http://localhost:8284/hello with a header:
    `Authorization: Bearer af54b774ac74b2d718a0609a4c6099`

    Of course you need your own Bearer token not mine! The URL is found in the API Store under the Hello API's listing. The token is in your subscription.

30) Here is a sample CURL request:
    `curl –H "Authorization: Bearer af54b774ac74b2d718a0609a4c6099" –X GET http://apimgr:8284/hello`
    (All one line!)

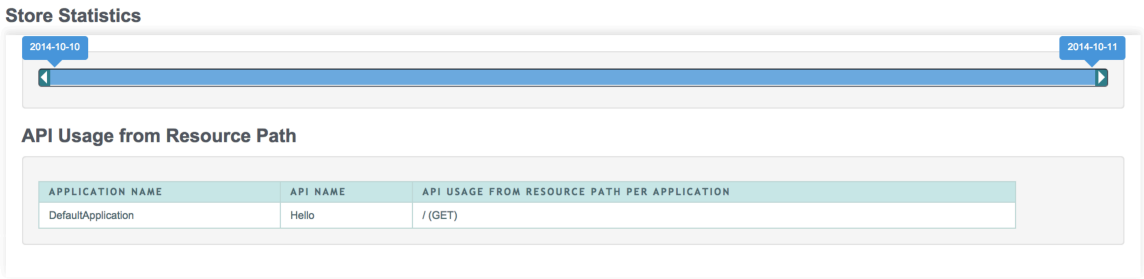
The Advanced REST client can be easily installed into Chrome and does a similar function.



Also try out hitting multiple times (to see throttling), and also try the sandbox token instead.

31) Analytics

You can see statistics both as a user and as a publisher. While you are still in the store, look at the Statistics.



32) Log back into the publisher and take a look at the stats there as well.

33) Extras

If you are finished early, there are lots more things to try. For example, see if you can Copy your existing API into a new version and publish that.

Send some requests into that and now check out the statistics.

That's all folks!

Appendix A

Setting up users and roles

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

- **Creator** : a creator is a person in a technical role who understands the technical aspects of the API (interfaces, documentation, versions, how it is exposed by API Gateway) and uses the API publisher to provision APIs into the API store. The creator uses the API Store to consult ratings and feedback provided by API users. Creator can add APIs to the store but cannot manage their lifecycle (i.e., make them visible to the outside world).
- **Publisher** : a publisher manages a set of APIs across the enterprise or business unit and controls the API lifecycle and monetization aspects. The publisher is also interested in usage patterns for APIs and as such has access to all API statistics.
- **Consumer** : a consumer uses the API store to discover APIs, see the documentation and forums and rate/comment on the APIs. S/he subscribes to APIs to obtain API keys.

The admin user can play the creator, publisher and subscriber roles described earlier. In this section, we explain how to set up these users or custom users and roles.

1. Log in to the management console user interface (https://apimgr:9443/carbon) of the API Manager using admin/admin credentials.
2. Select the **Users and Roles** menu under the **Configure** menu.
3. Click **Add New Role** and provide creator as the role name.
4. Click **Next**.
5. Select the following permissions from the list that opens and then click **Finish**.
    - Login
    - Manage > API > Create
    - Manage > Resources > Govern and all underlying permissions
6. Similarly, create the publisher role with the following permissions.
    - Login
    - Manage > API > Publish
7. You can now create users for each of those roles. To do so, click the **Users and Roles** menu under the **Configure** menu.
8. Click **Users**.
9. Click **Add New User**, provide the username/password and click **Next**.
10. Select the role you want to assign to the user (e.g., creator, publisher or subscriber) and click **Finish**. Given below is a list of usernames and the roles we assign to them in this guide.
11. Repeat the steps to create at least one user for all roles.