16L07A

IBM Integration Bus

Integration with IBM API Connect Advanced lab

Featuring:

In API Connect:
 add client id and secret to application using REST API
 add rate to product plan
 test rate limit
Run load test for application URL
Implement rate limit using IIB workload policies
Run another load test for application URL showing workload policies

November 2016
Hands-on lab built at product Version 10.0.0.6

Contents

1. Introduction

This lab guide builds on the *Integration with IBM API Connect* basic lab, by showing how rate limits defined for a product plan in API Connect can be used to limit traffic through the DataPower Gateway from an application subscribed to that plan.

In practice, plans with different rate limits may be created as tiered offerings with different pricing, so that plans with higher rate limits cost more to subscribe to than those with lower limits.

In order to apply rate limits, the REST API configuration in API Connect first needs to be changed by enforcing the use of credentials to identify and authenticate the application, which is done in this case by specifying the use of a client ID and client secret in the query.

Another way of enforcing rate limits is then shown, this time using a Workload Management Operational Policy within IIB.

2. Tasks

In this lab, you will perform the following tasks:

Within API Connect, as the API Manager:

1) Complete configuration of the REST API including requirement for credentials* to access the REST API
2) Create an application and subscribe it to a use plan within the product
3) Specify a rate limit for the use plan and test the REST API again.

As an API user

Run the following load tests using SOAP UI for the following:

1) Published REST API with (new) rate limit in the API Management use plan, to demonstrate limits enforced by the DataPower Gateway at the application level
2) Deployed REST API with rate limit enforced by WLM

***Applications' Credentials**

- Client ID: identifies applications that are making API calls so that application-specific quotas and statistics can be monitored, displayed, and enforced.
- Client Secret: used, like a password, in conjunction with Client ID.

Please note: this lab guide makes use of cloud-based IBM Bluemix services which are subject to change at any time. Whilst this document is updated from time to time to reflect these and other changes, it may not be fully up to date and therefore you may need to adapt the instructions it contains accordingly.

3. Add credentials to the REST API

In order to identify and authenticate subscribing applications, it is necessary to add this requirement in the API configuration.

1. If you are still logged into API Connect from before, then skip this step.

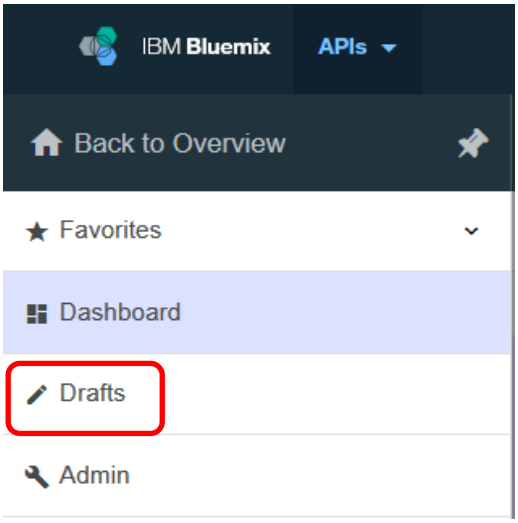
   Otherwise, go to https://new-console.eu-gb.bluemix.net/apis/apiconnect

   Log on with your IBM ID and password.

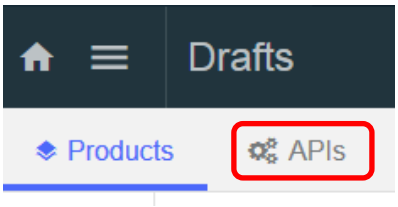
   You are now logged on to IBM API Connect as an API manager.

2. Click on the menu symbol ( ).

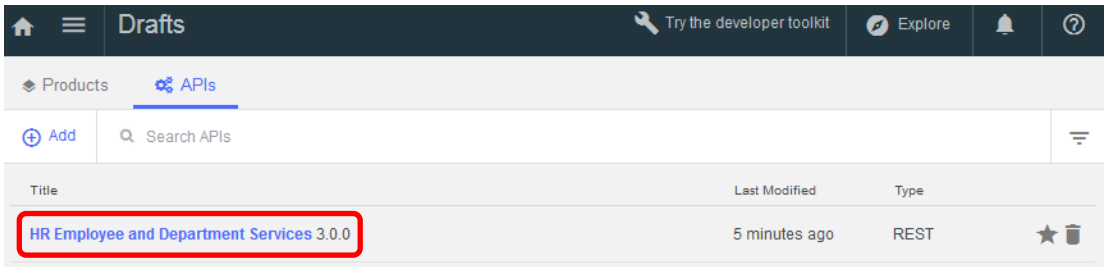
3. Then click **Drafts:**



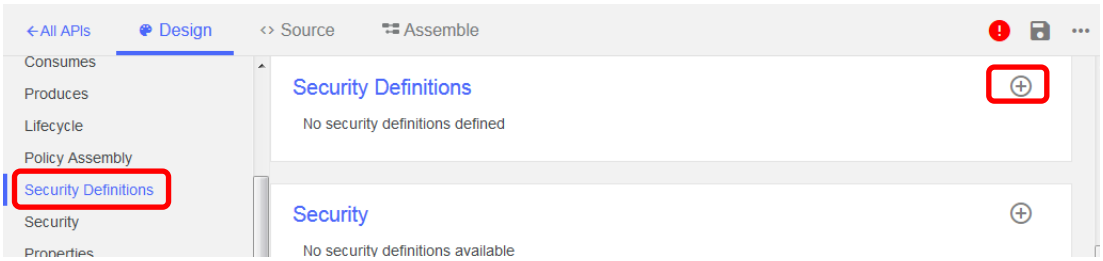
4. At the next screen, click **APIs:**



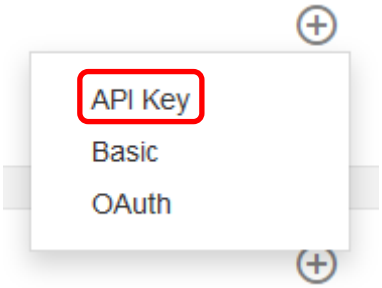
5. Click on the **HR Department and Employee Services** API.



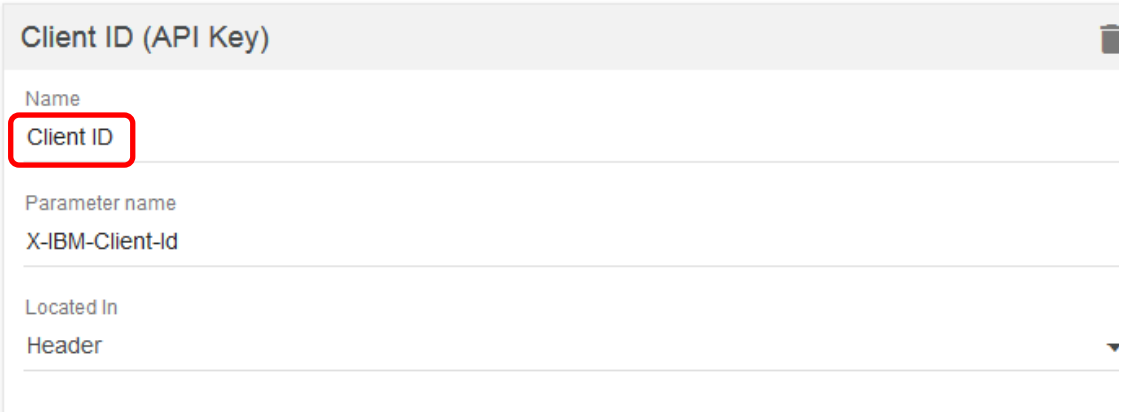
6. Navigate to the **Security Definitions** section and click the + sign



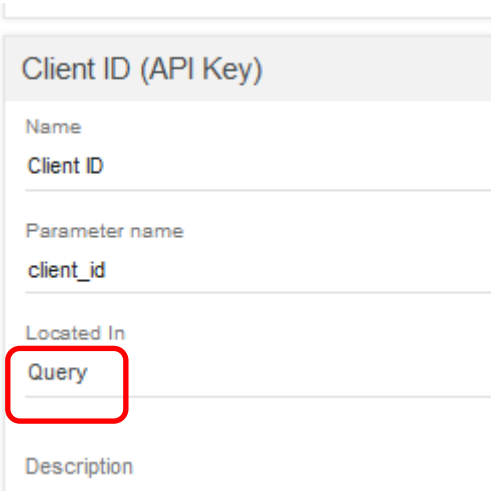
7. Select API Key



8. Check the Parameter name is as shown below and change the name the of the API Key to *Client ID*.



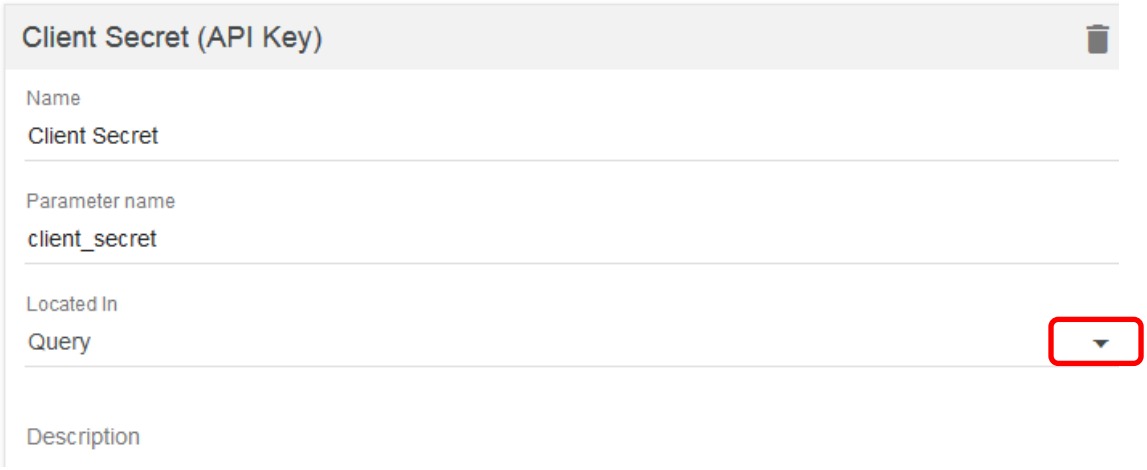
9. Change the location to **Query** so that credentials can be appended to the REST API request.



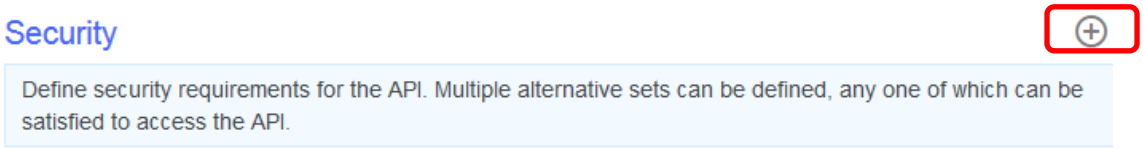
10. Add another security defintion:

11. Check the Parameter name is as shown and this time, call it *Client Secret*.

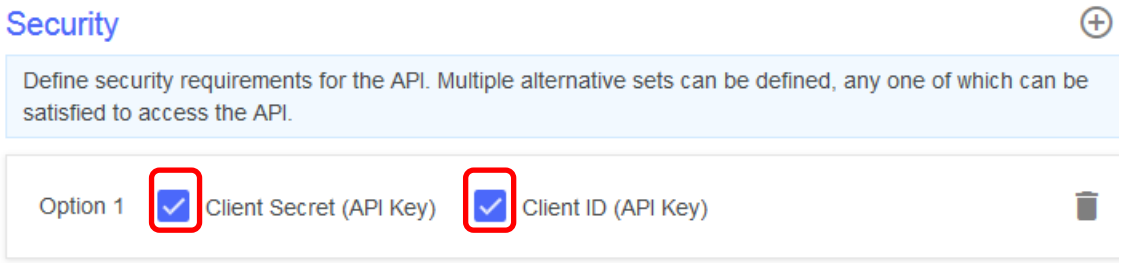
12. Change the **Located In** entry, as before, by selecting the down arrow and changing the setting from Header to Query.



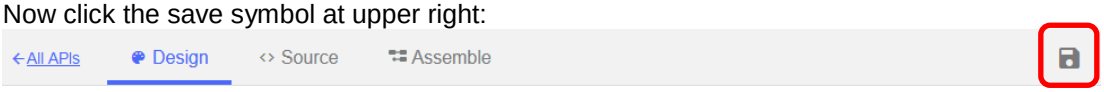
13. Navigate to Security and click + to add these Security Schemes.



14. Select these schemes

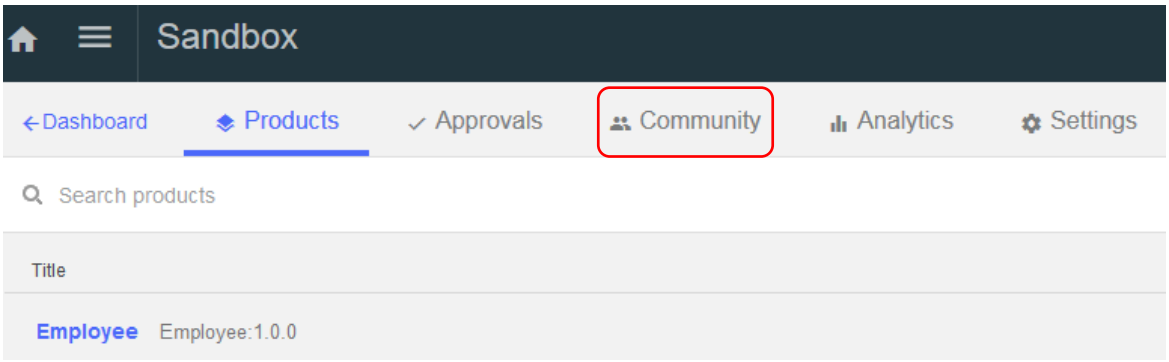


15. Now click the save symbol at upper right:

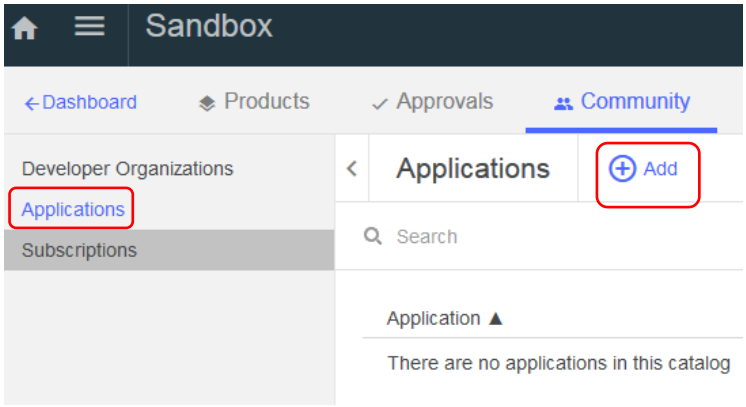


4. Create application and subscribe it to use plan

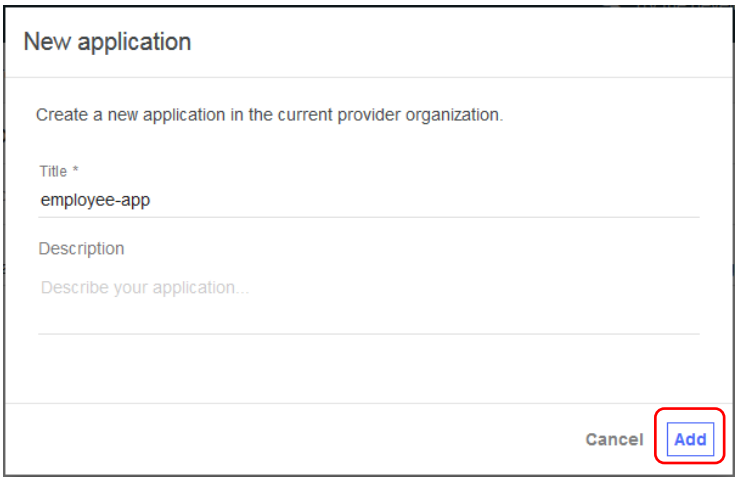
1. Navigate to the Dashboard and click on the Sandbox catalog icon.
2. Click on **Community**.



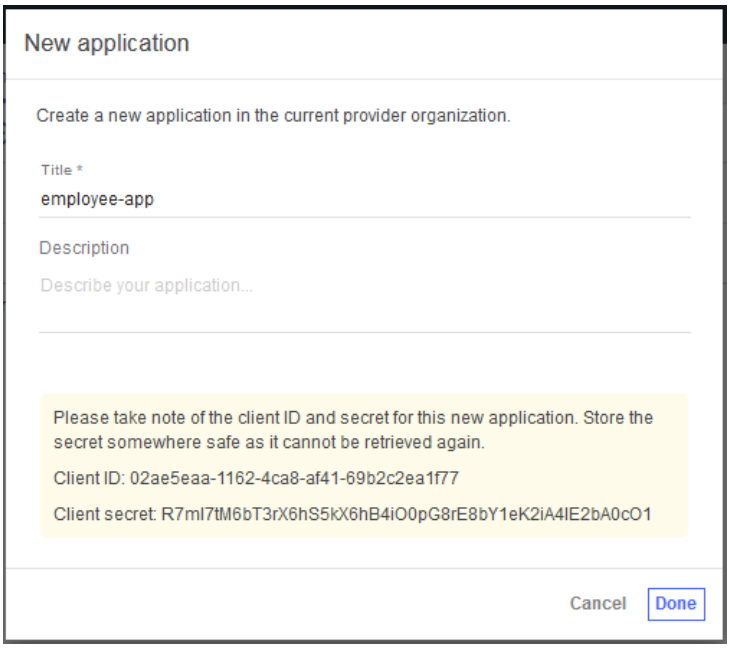
3. Then click **Applications** followed by **Add**.



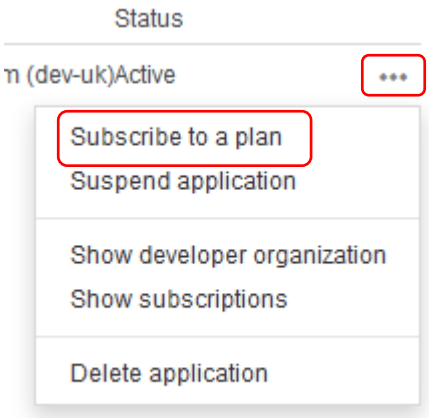
4. Provide a title for the new application and click **Add**.



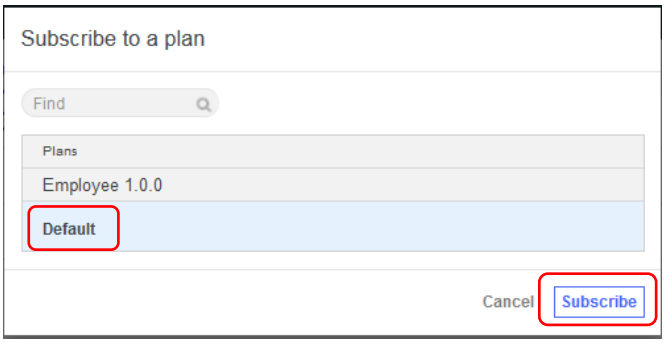
5. A client ID and secret identifying the application will be displayed. Copy these details to a Notepad You will not be able to view the secret again once this window is closed. Click **Done** when you have finished.



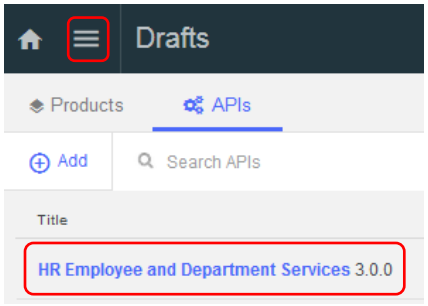
6. In the application view, click the 3 dots symbol next to the new application and select **Subscribe to a plan**.



7. Highlight the Default plan provided with the Employee product and click **Subscribe**.

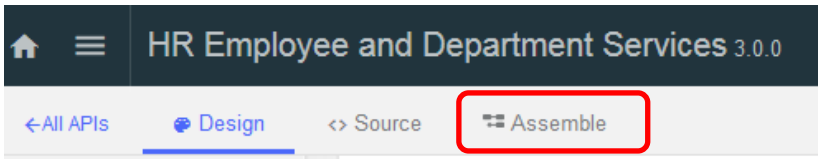


8. Click on the menu symbol at top left, then **Drafts** and navigate back to your REST API and click on it.

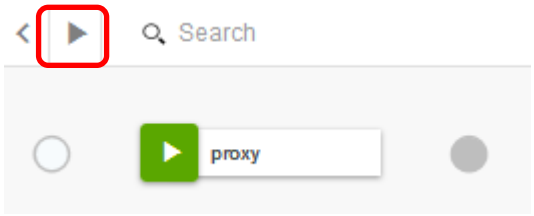


5. Test REST API with credentials

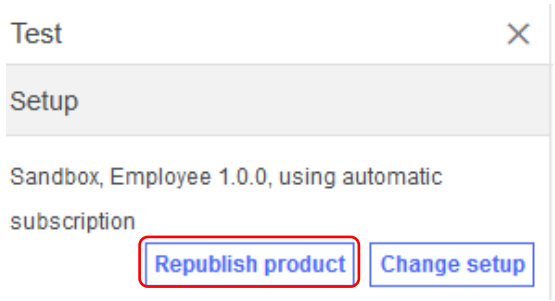
1. Click the **Assemble** tab.



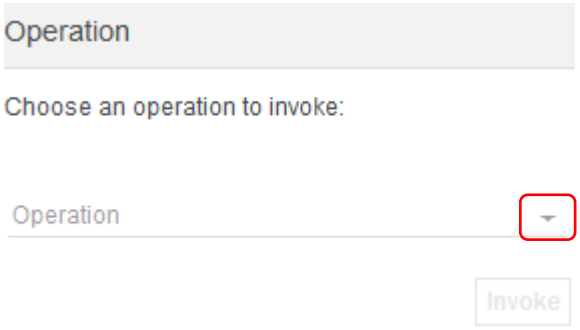
2. Click the arrow button to start the test.



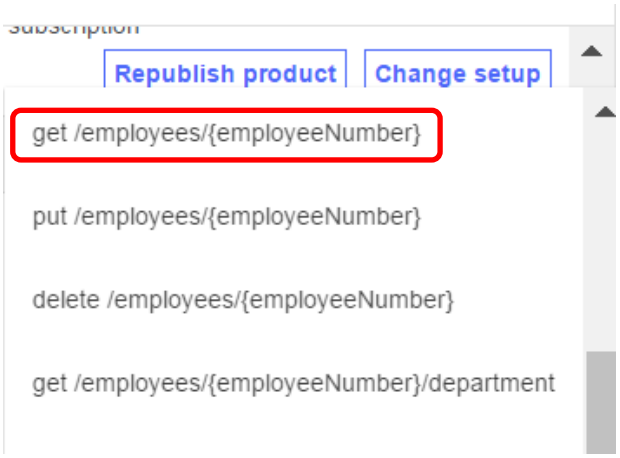
3. The test details are summarized. Click on **Republish Product**.



4. Now choose an operation to invoke by clicking the down arrow at the right.

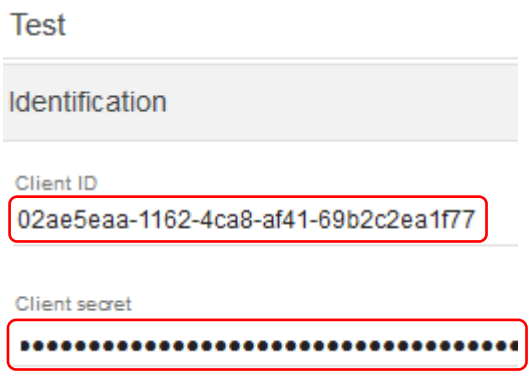


5. Scroll down and choose the **get /employees/{employeeNumber}** operation.

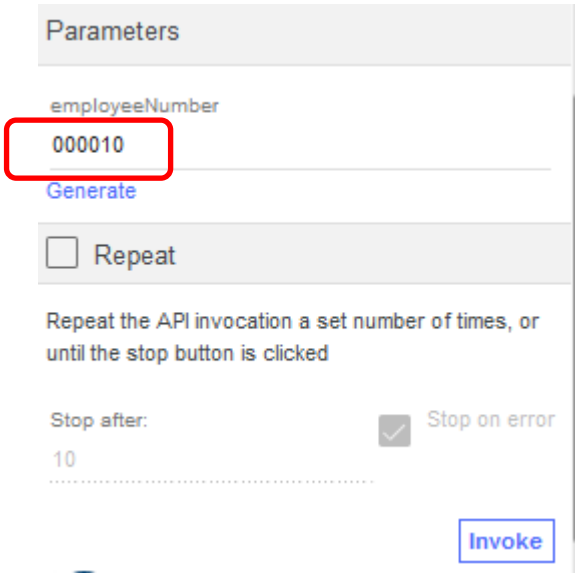


6. The Identification section is used to identify the calling application.

   A test client ID and secret are provided automatically, but we will not use them, because these are not for an application associated with the product. Delete these and enter the client ID and secret you saved earlier when you created your application.



7. Now, under the Parameters section, enter an employee number to retrieve, e.g. 000010 and click **Invoke**.

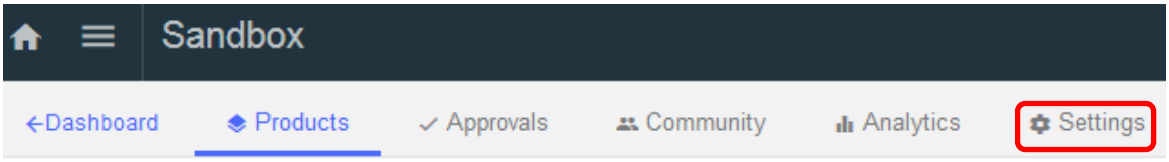


8. If successful, you will see returned data in the body of the response:

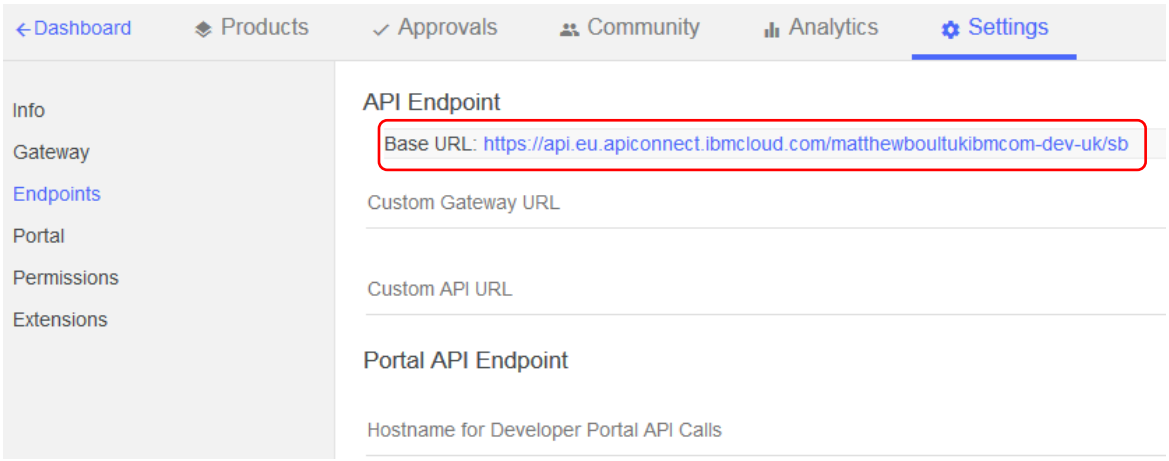
6. Determine URL of published REST API

1. Navigate to the Dashboard and click on the Sandbox catalog icon.

2. Click on the Sandbox icon and then **Settings**.



3. Select the **Endpoints** option from the menu at the left and note the API Endpoint Base URL.



4. The published URL will be of the form.

   <API Endpoint Base URL>/HR_Services/resources/employees/{employeeNumber}?client_id=<client ID>&client_secret=<client secret>

   Construct a URL in this form, using the API Endpoint, client ID and client secret (saved in the last section), and a test employee number.

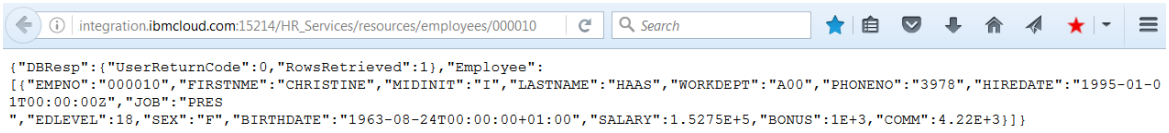
   An example is:

   https://api.eu.apiconnect.ibmcloud.com/iibtester99workshop-dev/sb/HR_Services/resources/employees/000010?client_id=756566f7-4fee-45a9-afda-9c79ab8dbecc&client_secret=R5bB2aE8vC5iG2oJ7kU7vS3kN6eI2qJ3yJ4qM6cS7wH1xG8mJ1

   To help you do this, we have provided the following template file for you to edit:

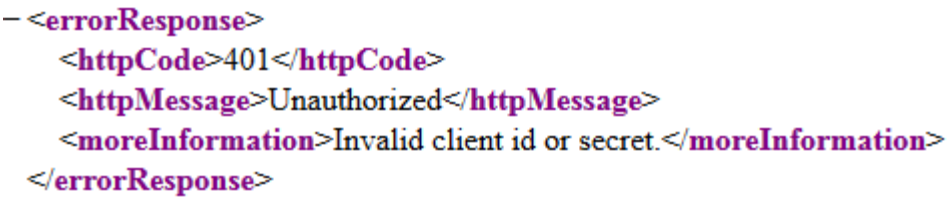
   C:\student10\IIB_APIC\resources\Published_API_with_Credentials_template.txt

5. Try this from a browser and you should see a response showing the relevant employee's details:



6. Copy the URL for future reference. **Keep the browser tab open – you will need it again.**

7. In another browser tab, paste the URL again, but change the client secret slightly. You should see an error message, like this:

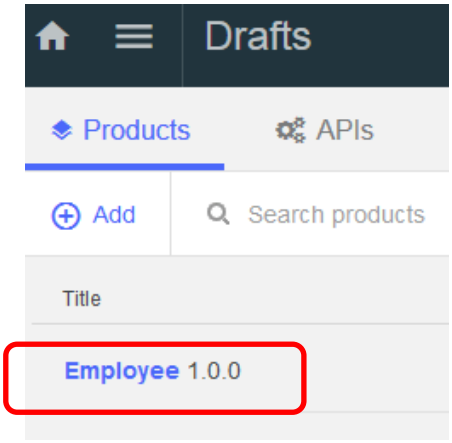


7. Apply a rate limit to the product plan

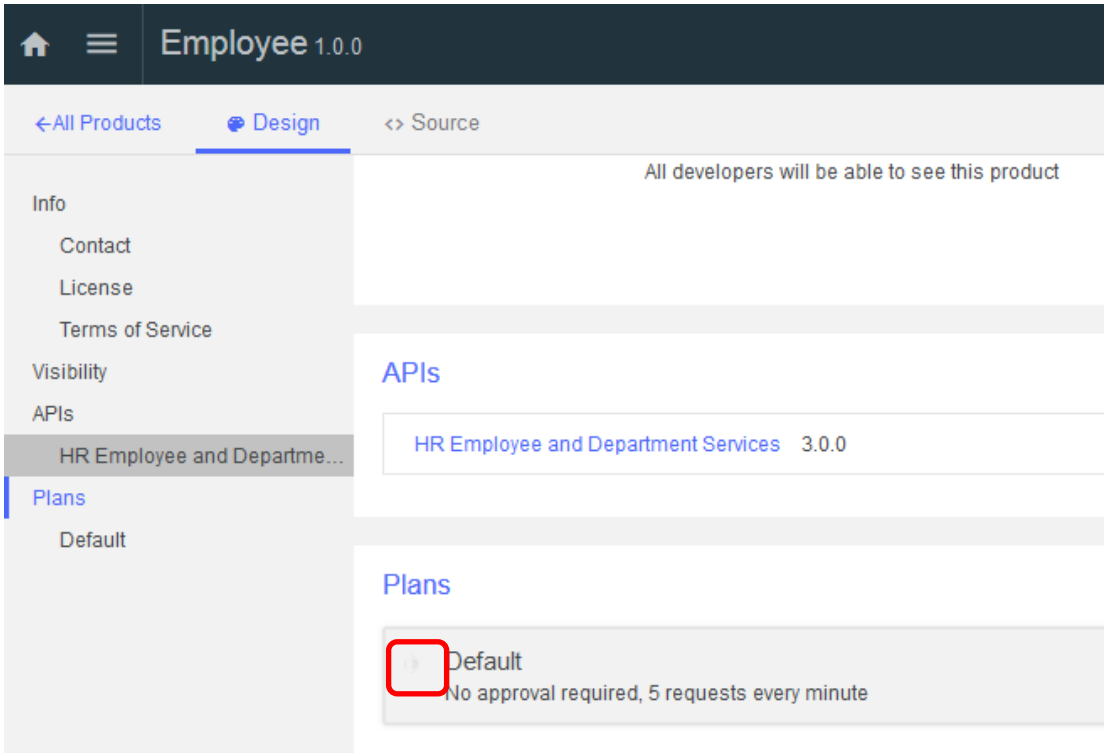
So far, access has been restricted by use of Client ID and Client Secret. If these details are incorrect, then the requested query will not work, as we have seen earlier.

Now we will apply a limit to the number of valid transactions that can be accepted in a given period of time. This is the **rate limit**.

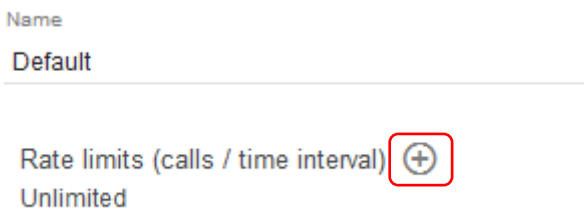
1. Go to the product view under the menu ( ) then **Drafts** and click on the product.



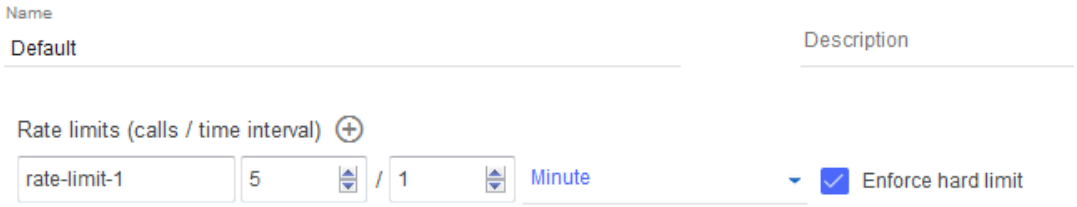
2. Navigate to the Default plan in the Plan section and click the twisty as shown.



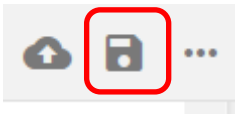
3. Add a rate limit by clicking the plus symbol as shown.



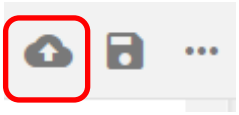
4. Now, set a rate limit of say 5 calls per minute and enforce a hard limit.



5. Save the plan settings in the product by clicking the save symbol at the top right:



6. Then restage the plan by clicking as shown.



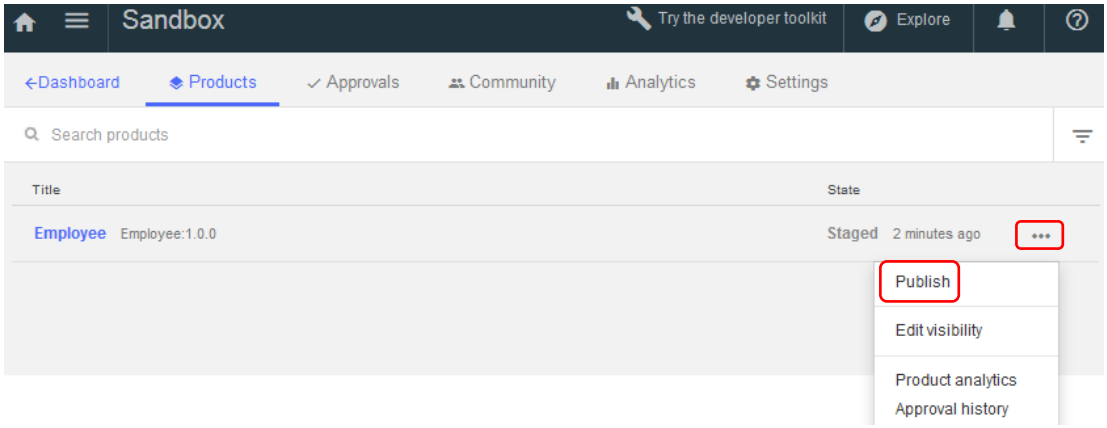
7. Select the Sandbox catalog from the option provided.

   This will result in a message to say that the product has been staged. This can be dismissed.

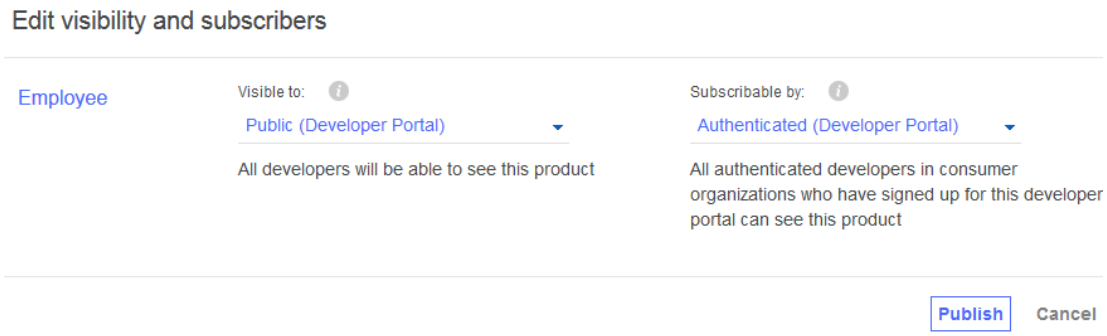
8. The product has been staged and now needs to be published again. Navigate to the

   Sandbox catalog by clicking on the menu icon () then **Dashboard** and then click on the Sandbox icon.

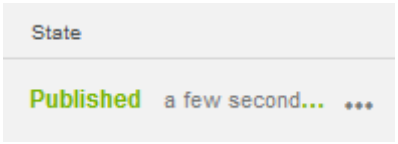
9. The product view will be shown. Click on the dotted menu symbol at the right and then select **Publish**.



10. At the next window, accept the defaults and click **Publish**.



11. The plan state should now be changed:



12. Test the published REST API URL from a browser window again repeatedly. If you exceed the rate limit you have just specified, you should see a message like this:

```
−<errorResponse>
    <httpCode>429</httpCode>
    <httpMessage>Too Many Requests</httpMessage>
    <moreInformation>Rate Limit exceeded</moreInformation>
</errorResponse>
```

8. Call published REST API from SOAP UI

In this section, you will make requests to the published REST API from SOAPUI. (SOAPUI can be used to test REST APIs as well as SOAP APIs).

8.1 Request from SOAPUI

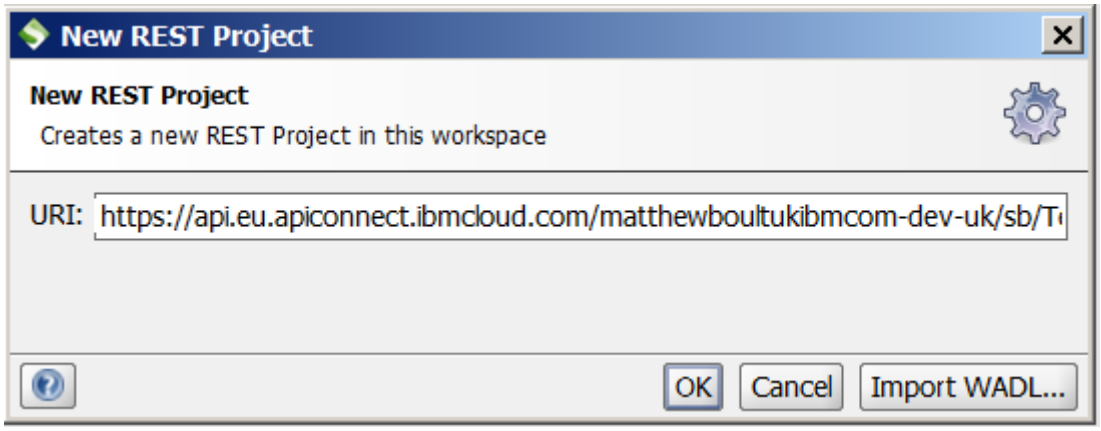
1. Locate and start SOAPUI from the Start Menu.

2. Click on **File**, then **New REST Project**.

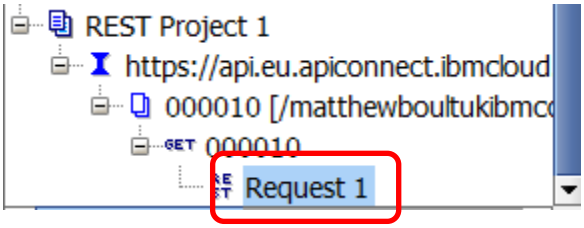
3. The **New REST Project** dialogue will open. You must specify an example of the URL of the operation in the published REST API. Use the URL you used when you tested the published REST API in a browser, with this format:

```
https://<API Endpoint Base URL>/HR_Services/resources/employees/
            {employeeNumber}?client_id=<client ID>
                &client_secret=<client secret>
```

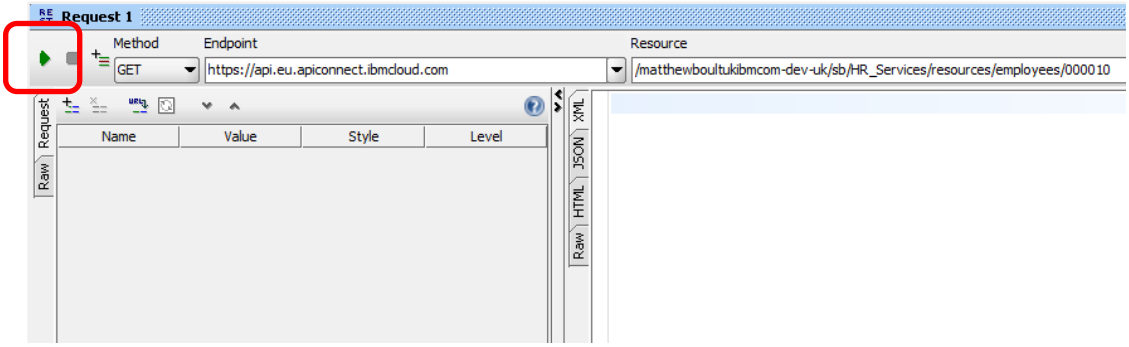
   Click **OK**.



4. A new REST API project will be created and within it, a new request:

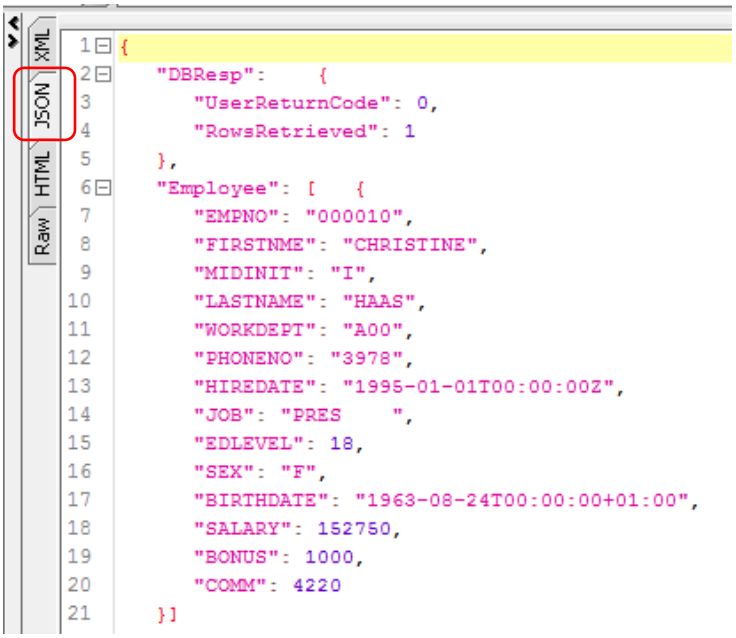


5. The new request will appear as follows. Click on the play button (green arrow) on the request.



6. The results will be displayed on the right.

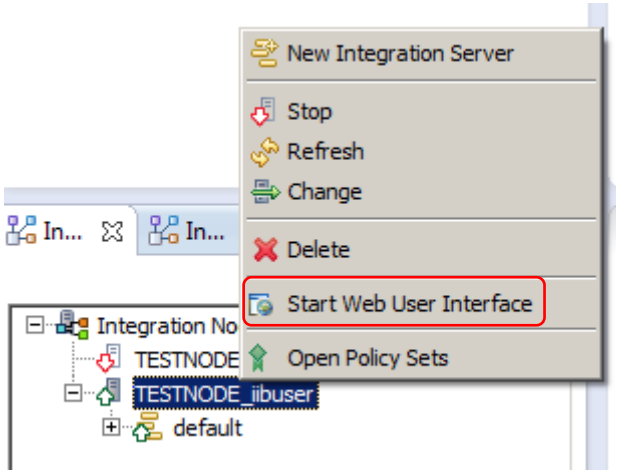
   Click the JSON tab to make sure the data is shown in JSON format.



8.2 View throughput for published REST API in IIB Web UI

The web user interface for an integration node can be used to view throughput information for a REST API.

1. In the Integration Toolkit, navigate to the Integration Nodes view. Right click **TESTNODE_iibuser**, and click on **Start Web User Interface**:

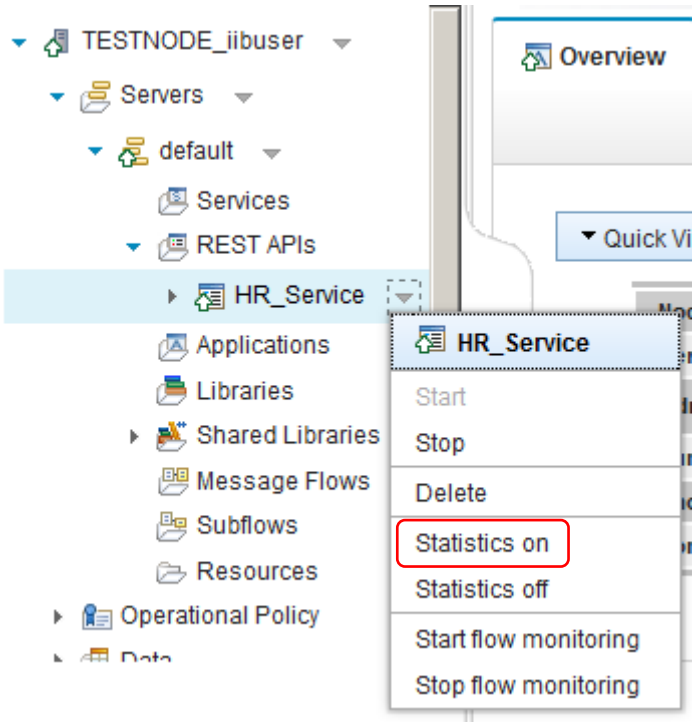


A web browser will automatically open showing the web user interface.

2. In the web user interface, expand **TESTNODE_iibuser** to find the REST API that you have deployed.

Click on the drop down arrow to the right of the **HR_Service** label. (Do not click on the **HR_Service** API!)

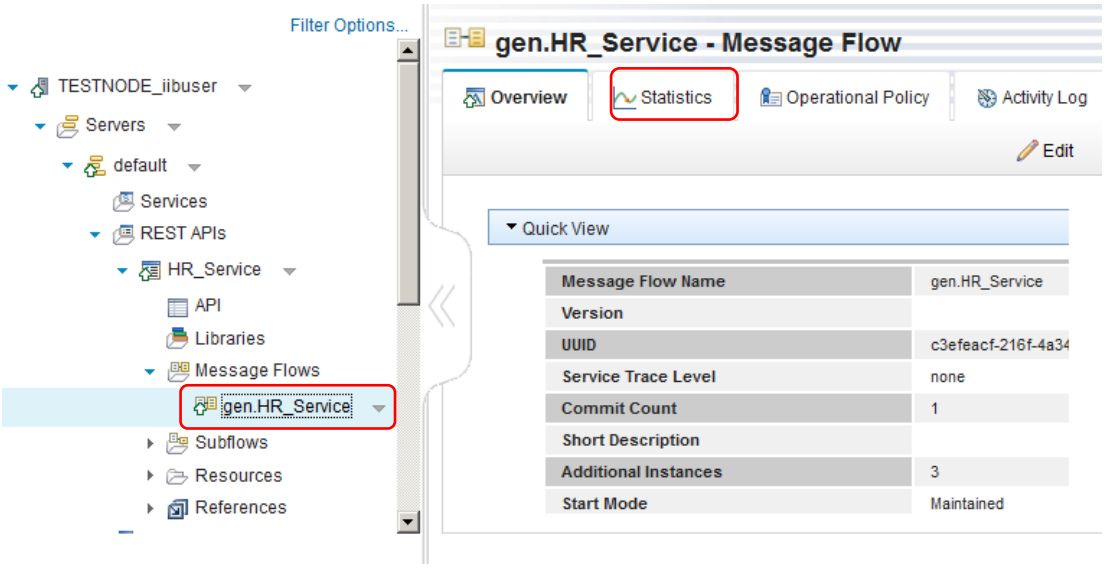
Click on **Statistics on** to enable statistics collection for this REST API:



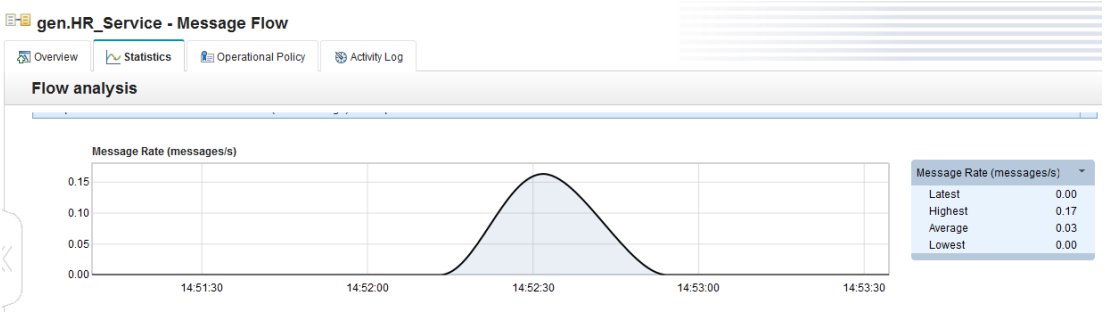
3. A message will appear after a few seconds to confirm that flow statistics have been successfully turned on. Statistics will now be collected for all requests made to the published REST API.

   In order to view these statistics, navigate to message flow **gen.HR_Service**.

   Click on the message flow, and then click on the **Statistics** tab at the right.



4. The statistics view is updated every 20 seconds. Use SOAPUI to manually submit several requests to the published REST API and you should see the statistics view in the web user interface updated accordingly:



9. Load Test published REST API

A load test will be run to show how IBM API Management can be used to control throughput (at the application level) so that a backend system such as a database is not adversely affected by traffic spikes.

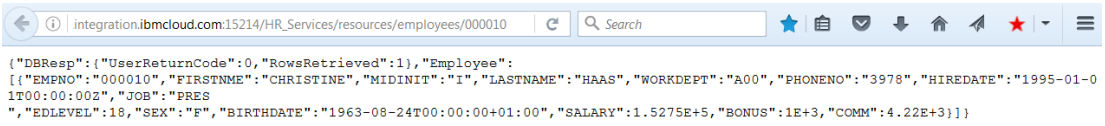
Note: throughput is controlled by the IBM DataPower Gateway in the API Connect environment on Bluemix,

In order to do this, the rate limit of the API Connect Default plan will be increased.

9.1 Change use plan rate limit

1. Repeat section 7, **Apply a Rate Limit to the product plan**, but this time change the rate limit to 5 requests per second.

2. Test the published URL in a browser once more, beyond the original rate limit (e.g. 5 calls per minute) to confirm that the limit has indeed been raised.

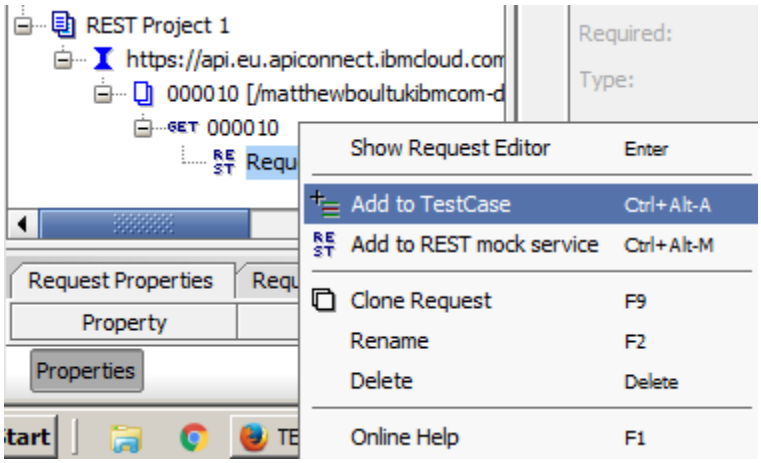


{"DBResp":{"UserReturnCode":0,"RowsRetrieved":1},"Employee":
[{"EMPNO":"000010","FIRSTNME":"CHRISTINE","MIDINIT":"I","LASTNAME":"HAAS","WORKDEPT":"A00","PHONENO":"3978","HIREDATE":"1995-01-0
1T00:00:00Z","JOB":"PRES
","EDLEVEL":18,"SEX":"F","BIRTHDATE":"1963-08-24T00:00:00+01:00","SALARY":1.5275E+5,"BONUS":1E+3,"COMM":4.22E+3}]}

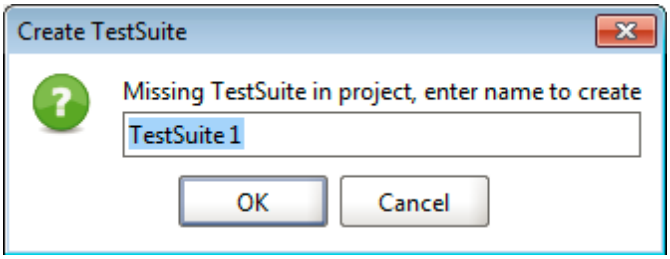
9.2 Create and run load test

You will now create a Load Test in SOAPUI to demonstrate the new rate limit.

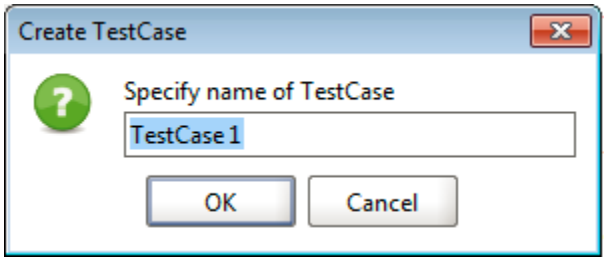
1. Switch back to SoapUI and right click **Request 1** in the project tree, and click on **Add to TestCase**:



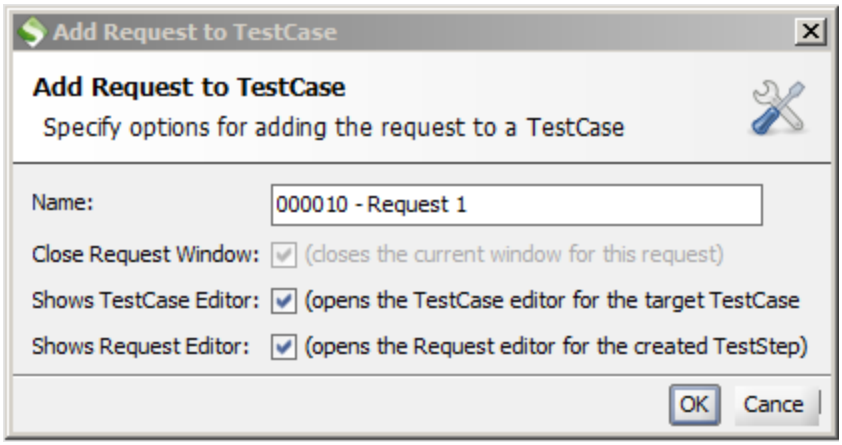
2. The **Create TestSuite** dialogue opens. Click **OK**.



3. The **Create TestCase** dialog opens. Click **OK**.

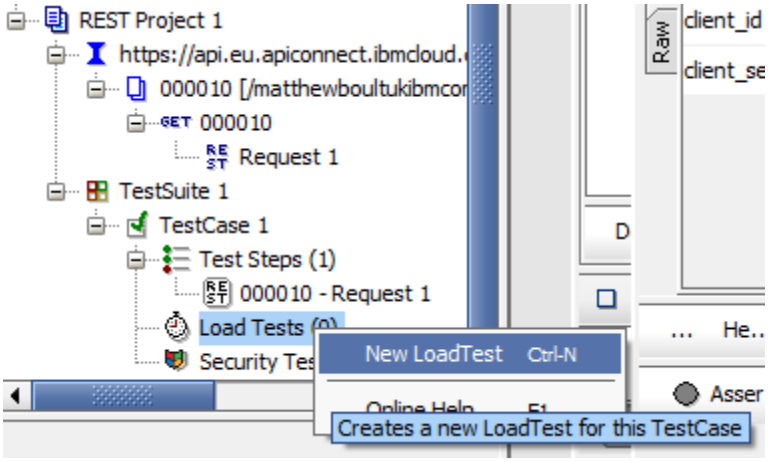


4. The **Add Request to TestCase** dialog opens. Click OK.

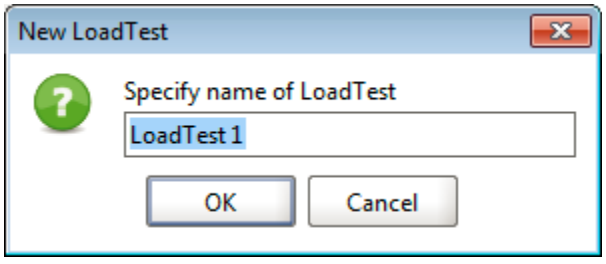


5. The new TestSuite and associated artefacts now appear in the project tree.

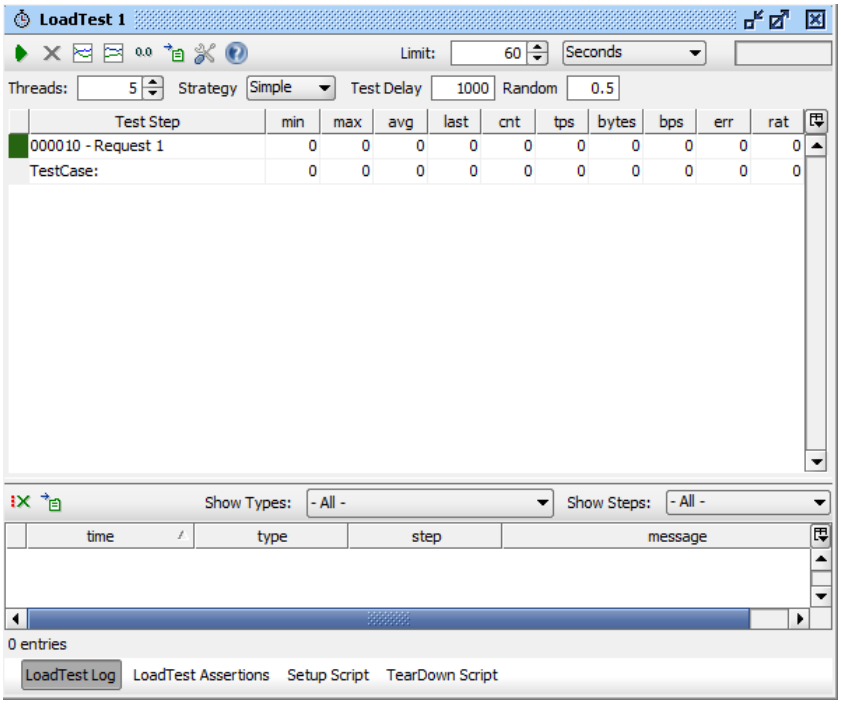
   Right click **Load Tests** in the project tree, and click on **New LoadTest**.



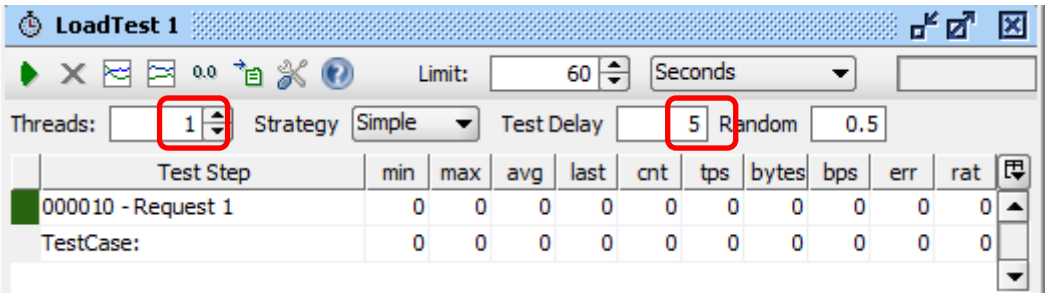
6. The New LoadTest dialogue appears. Click **OK**.



7. A new load test appears:

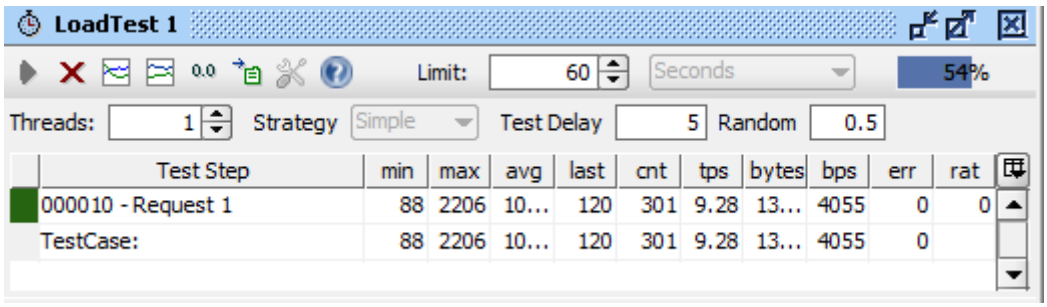


8. Change the value of **Threads** to 1 and the value of **Test Delay** to 5 (ms). In tests, these generated a throughput of around 10 requests per second (before rate limiting) for the duration of the test, which is sufficient for our purposes.



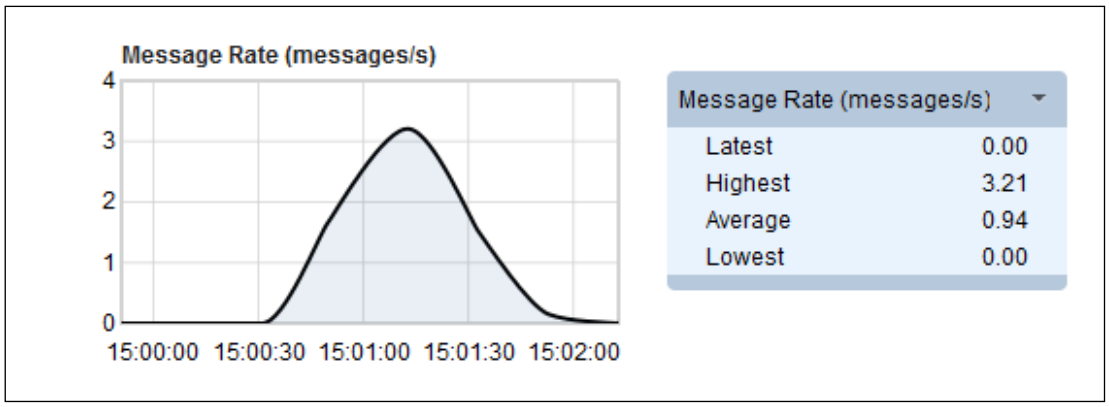
9. Click on the play button on the load test. It will run for 60 seconds by default unless stopped using the stop button.

   You will see the results of the load test update in SoapUI.



10. The statistics view in the web UI will show that the message rate has been limited by the IBM DataPower Gateway, as per the limit set by the API Connect product plan for the deployed REST API.

    Note the rate in IIB may not reach the full limit controlled by APIC, if other factors such as memory and processor provide additonal constraints.



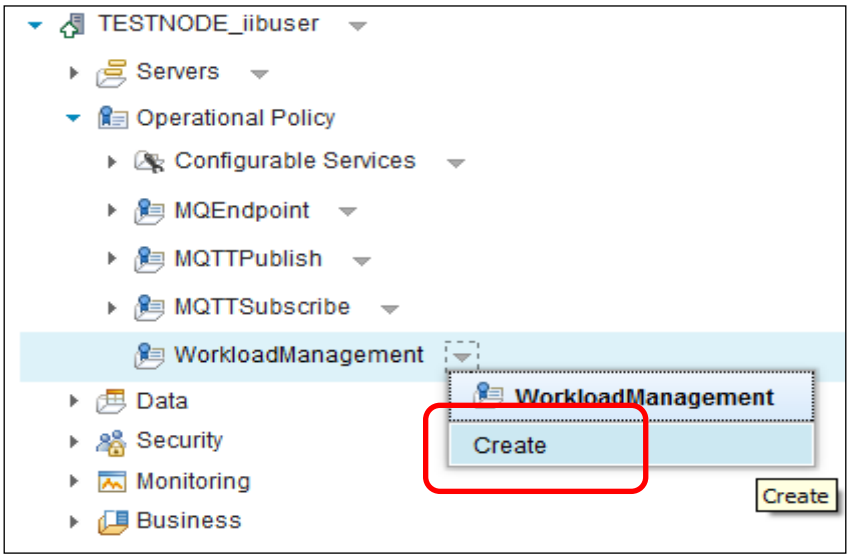
10. Load Test Deployed API using IIB WLM

The throughput in the load test in the last section was throttled by rate limits enforced by the IBM DataPower Gateway, at the application level before traffic reached the local network.

To control throughput for the deployed API overall, we will use a separate workload management (WLM) policy within IIB, i.e. at the centre.

1. Open the web user interface for the integration node. Find **WorkloadManagement** in the tree on the left under **TESTNODE_iibuser** -> **Operational Policy**.

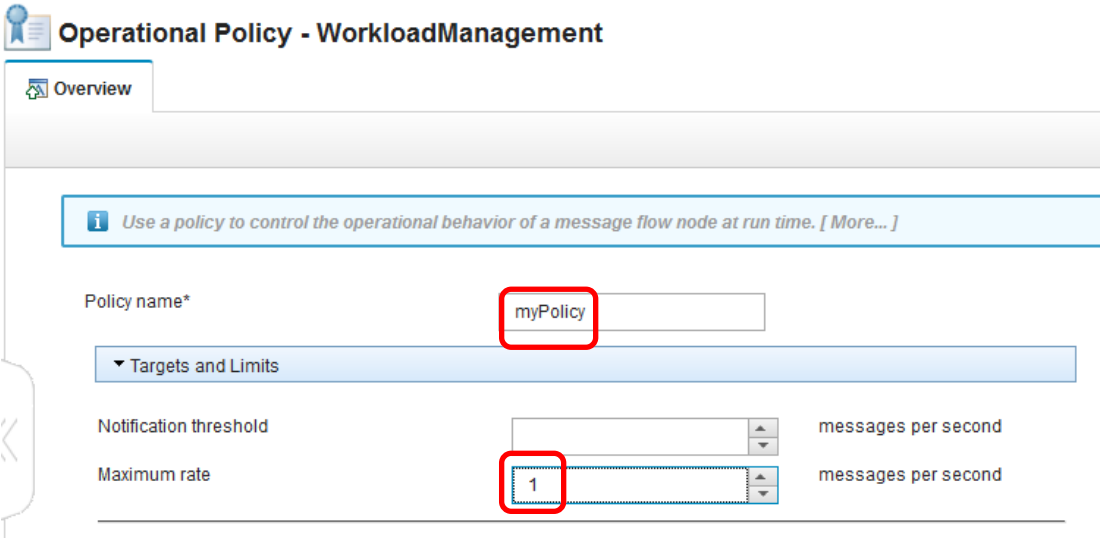
   On the drop-down arrow, select Create.



2. A new WorkloadManagement policy is created and displayed.

   Give the policy a name (for example, **myPolicy**), and set the Maximum rate to 1 message per second and click the **Save** button.

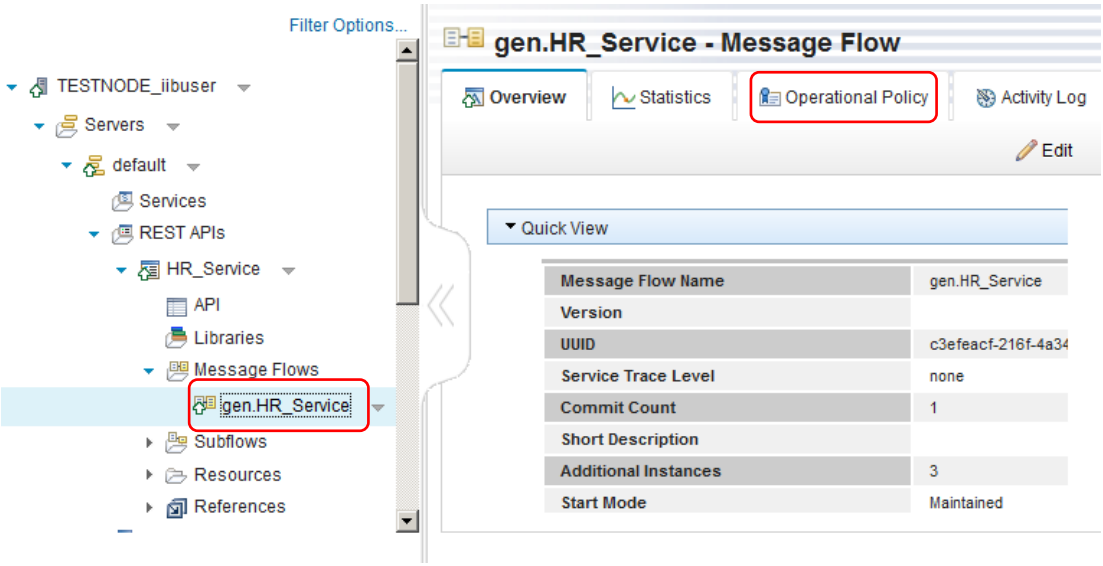
   Note: in practice, the limit of the backend server (in this case, IIB), could be higher than the limit of the front-end (APIC). For the purposes of this lab, the IIB limit will be lower than the APIC limit.



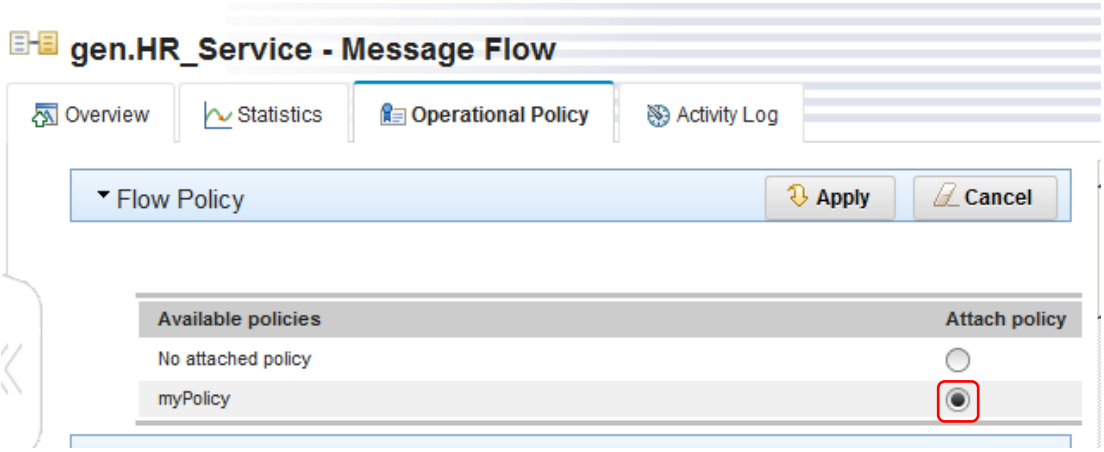
   The WLM policy must be attached to a message flow in order to dynamically configure that message flow to restrict the rate of requests to the REST API.

   A WLM policy can be attached to multiple message flows. If the WLM policy is updated whilst attached to message flows, all of those message flows are updated when you save the WLM policy.

3. To apply the WLM policy to the deployed message flow, navigate to message flow **gen.HR_Service** and click on the Operational Policy tab:

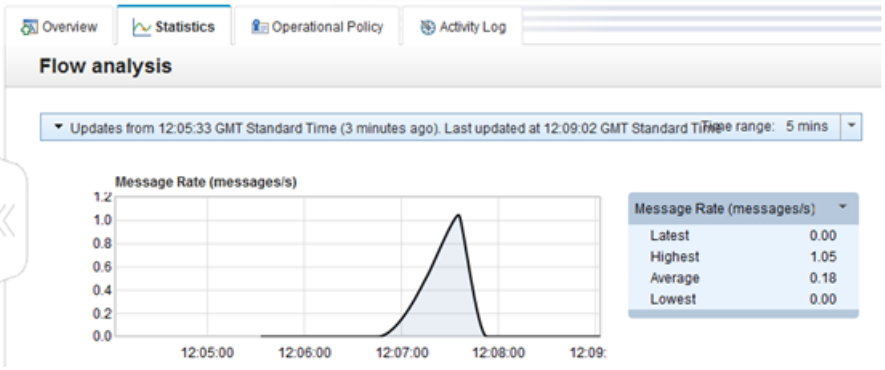


4. Find the WLM policy you created, and select the radio button. Once the WLM policy is selected, click on **Apply** to apply the WLM policy to the message flow. The changes take effect immediately.



5. Run a new load test, repeating the steps in section 9.2, Create and run load test.

6. The statistics view in the web UI will show that the message rate has been limited as it was before. In this case, though, the throughput has been limited further, within IIB by WLM.



END OF LAB GUIDE