**DS – Lab 7**

**Exercise 1 – Service Integration using an ESB (Enterprise Service Bus)**

1. This exercise is based on the following tutorial.

<https://docs.wso2.com/display/EI660/Sending+a+Simple+Message+to+a+Service>

**Download WSO2 Enterprise integrator and install it from the following link. Note the installation directory.**

[https://wso2.com/enterprise-integrator/6.6.0#](https://wso2.com/enterprise-integrator/6.6.0)

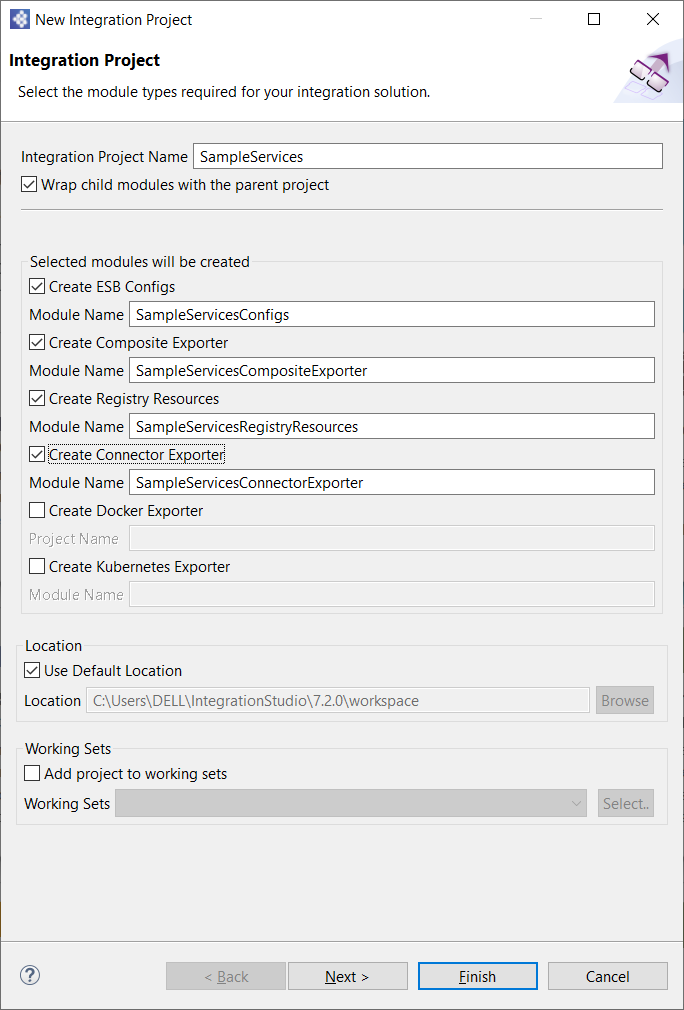
**Download and extract the WSO2 Integration Studio from here.**

<https://wso2.com/integration/integration-studio/>

2. Go to WSO2 Integration Studio directory and double click the file ‘IntegrationStudio.exe’.

3. Select File -> New -> Integration Project

4. Type ‘SampleServices’ as the project name and select the check boxes for **Create Registry Resources**  and **Create connector exporter.**



5. Select ‘Next’ and then click ‘Finish’.

6. Right click SampleServicesConfigs in the project explorer and select New ->Endpoint.

7. Select ‘Create a new Endpoint’.

8. As End point name, enter ‘QueryDoctorEP’.

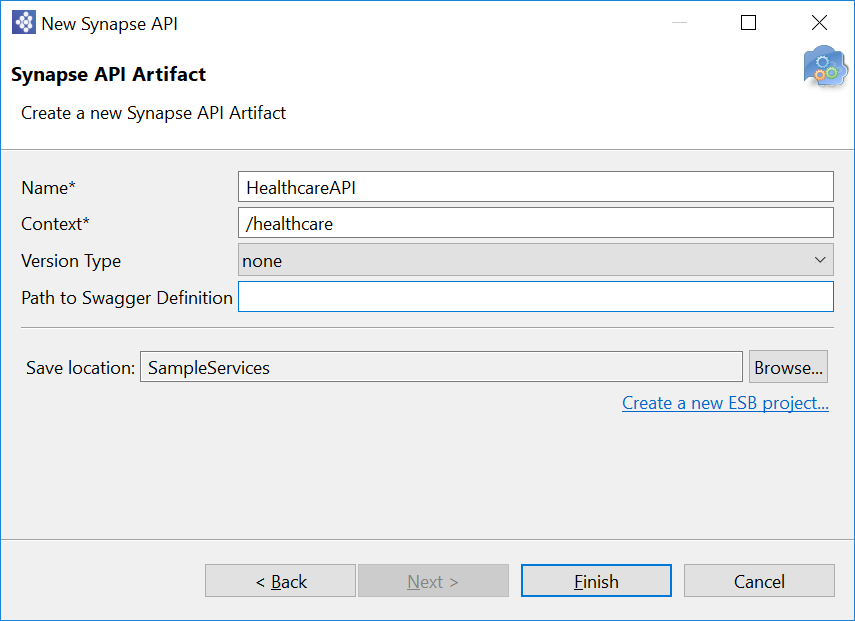
9. As the Address, enter ‘[http://localhost:9090/healthcare](http://localhost:9090/healthcare/%7Buri.var.category)’.

10. Click ‘Finish’. This will configure the backend web service in the integration project.

11. Right click SampleServicesConfigs in the project explorer, and select New -> Rest API.

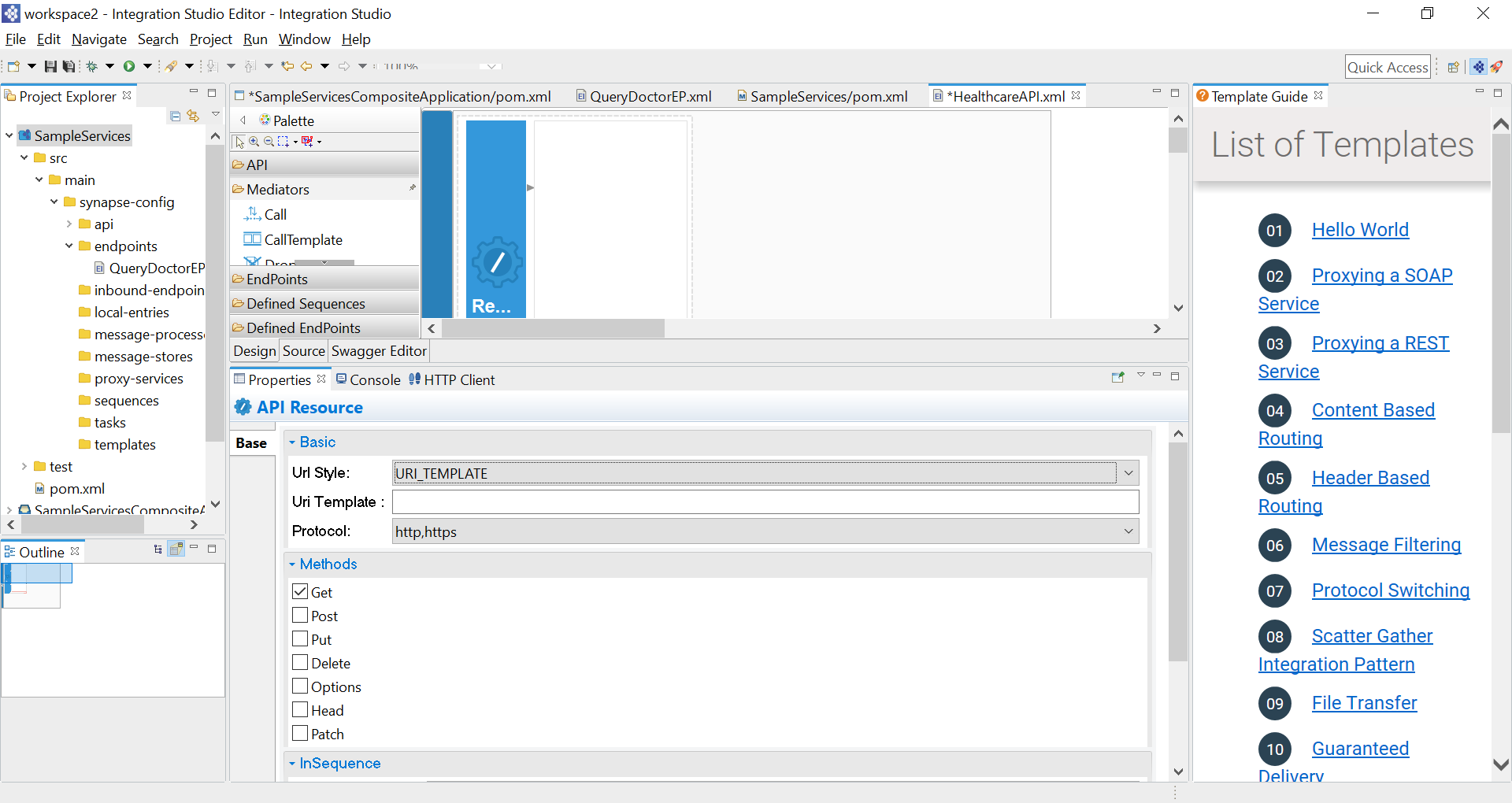
12. Select ‘Create a new API Artifact’.

13. Enter the following information.



14. Click ‘Finish’. You should see a graphical view of the API.

15. Double click the ‘Resource API (API compartment)’ box in the graphic view. This should open a properties view. Enter the following values.



Uri Style – URL\_TEMPLATE

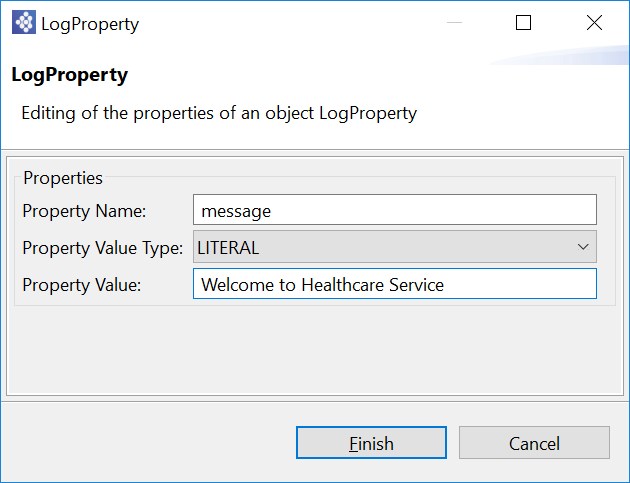
Uri Template – empty

Protocol – HTTP, HTTPs

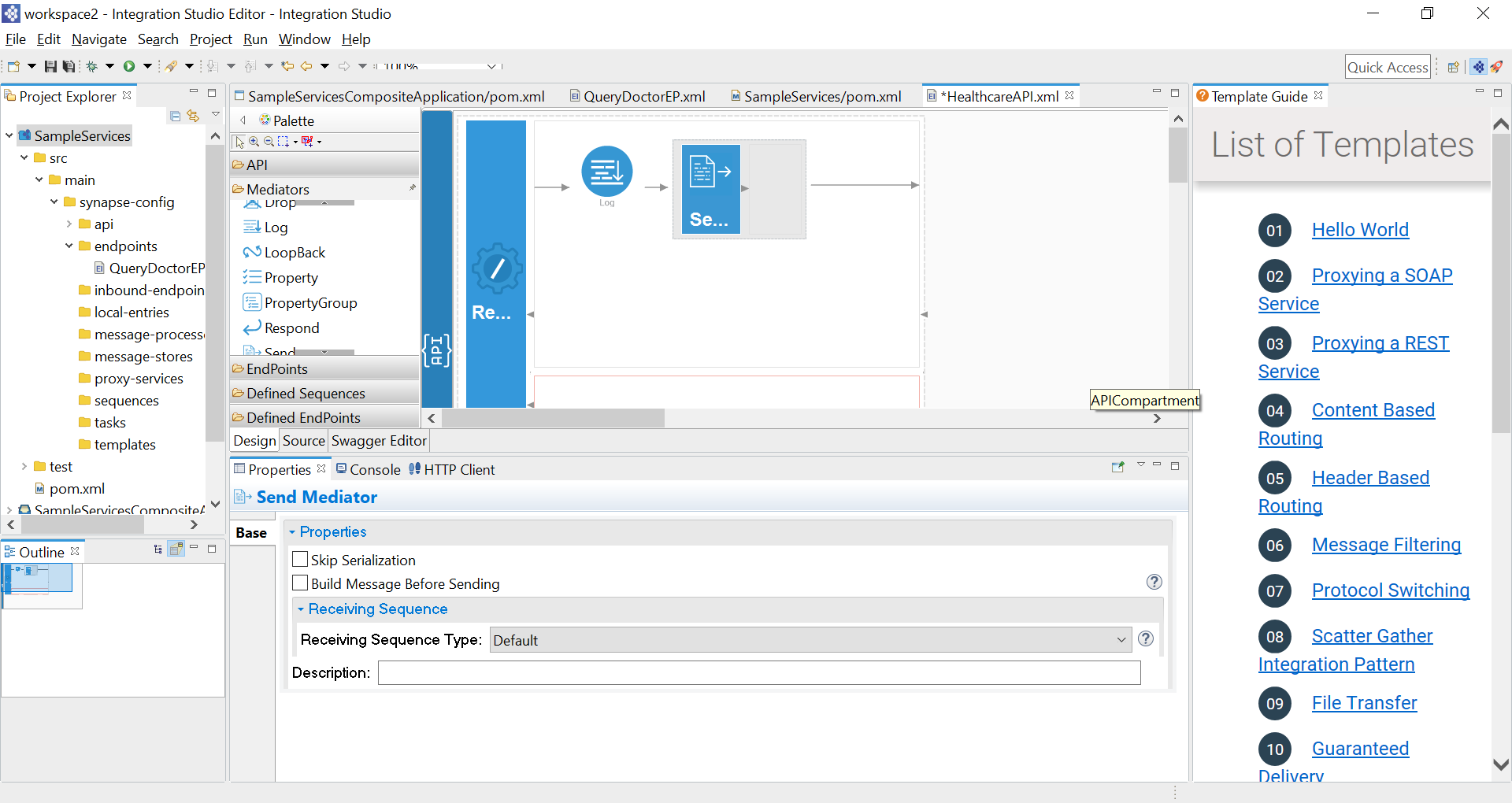
Methods – Get

16. From the ‘Mediators’ palate, drag the log mediator and place it in the box next to the ‘Resource’ box.

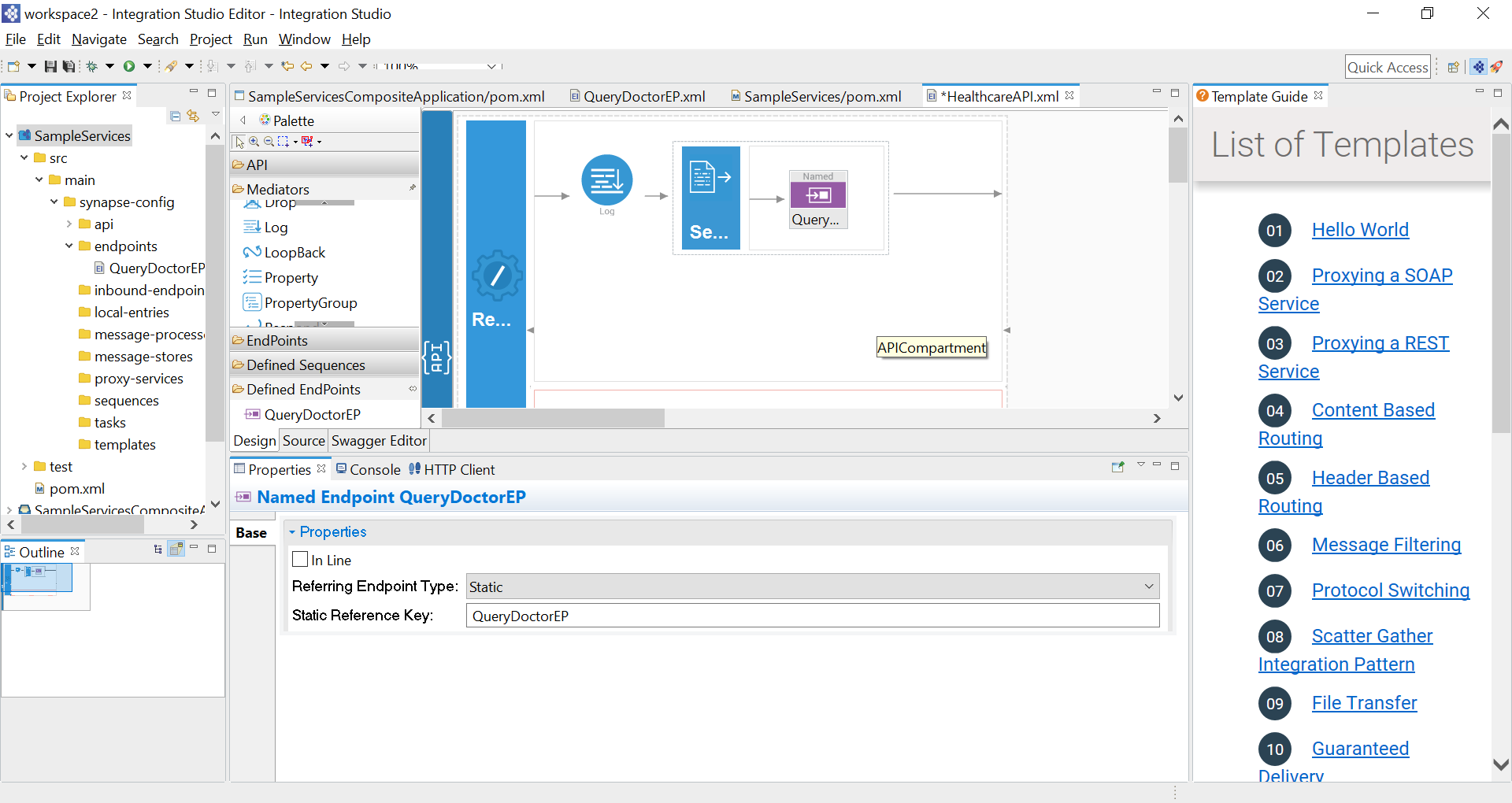
17. Double click the Log mediator and click the + sign in log properties window. Add the following details under Log property.



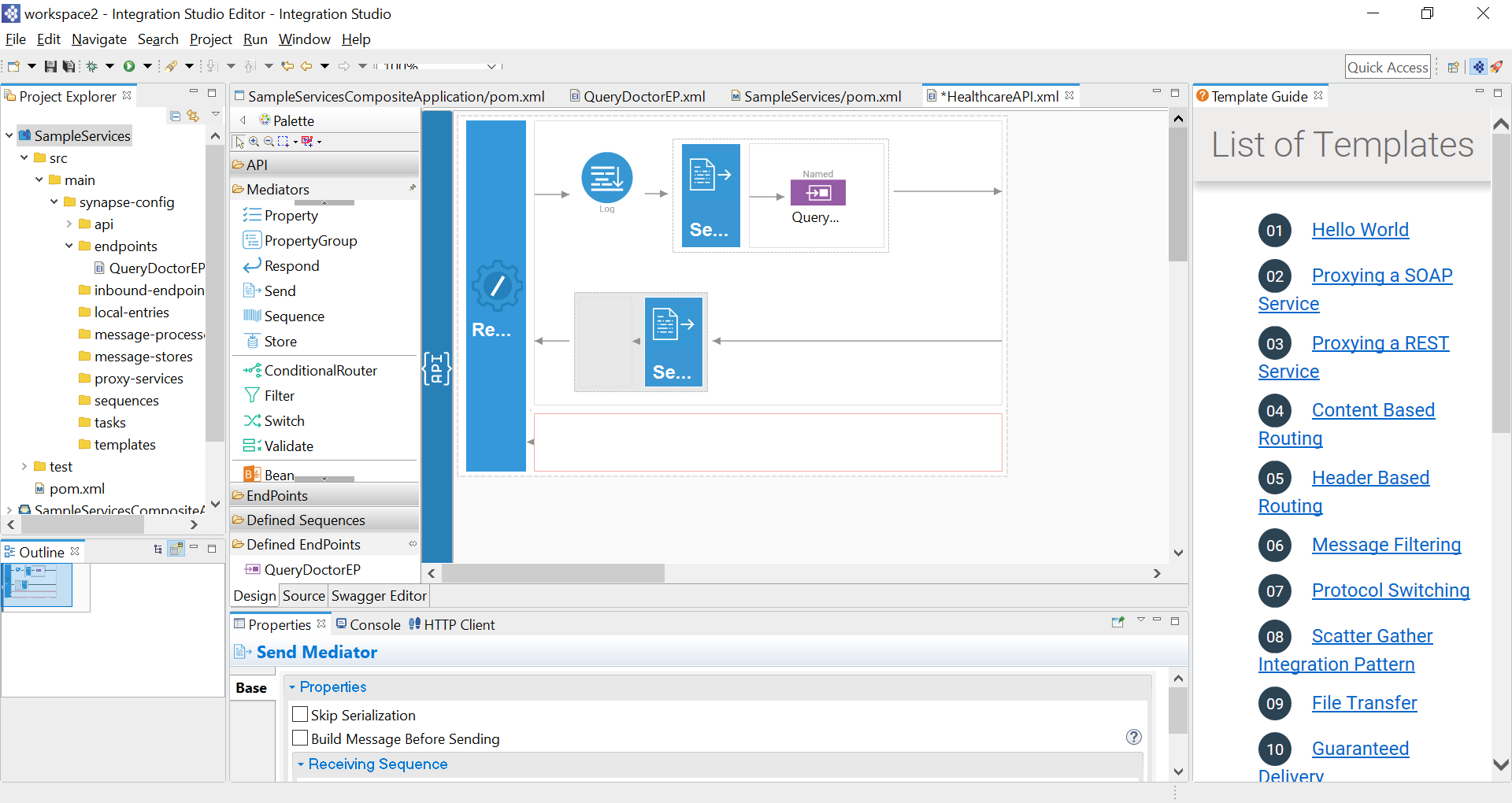
18. From the Mediators palate, drag the ‘Send’ mediator to the right of the log mediator.



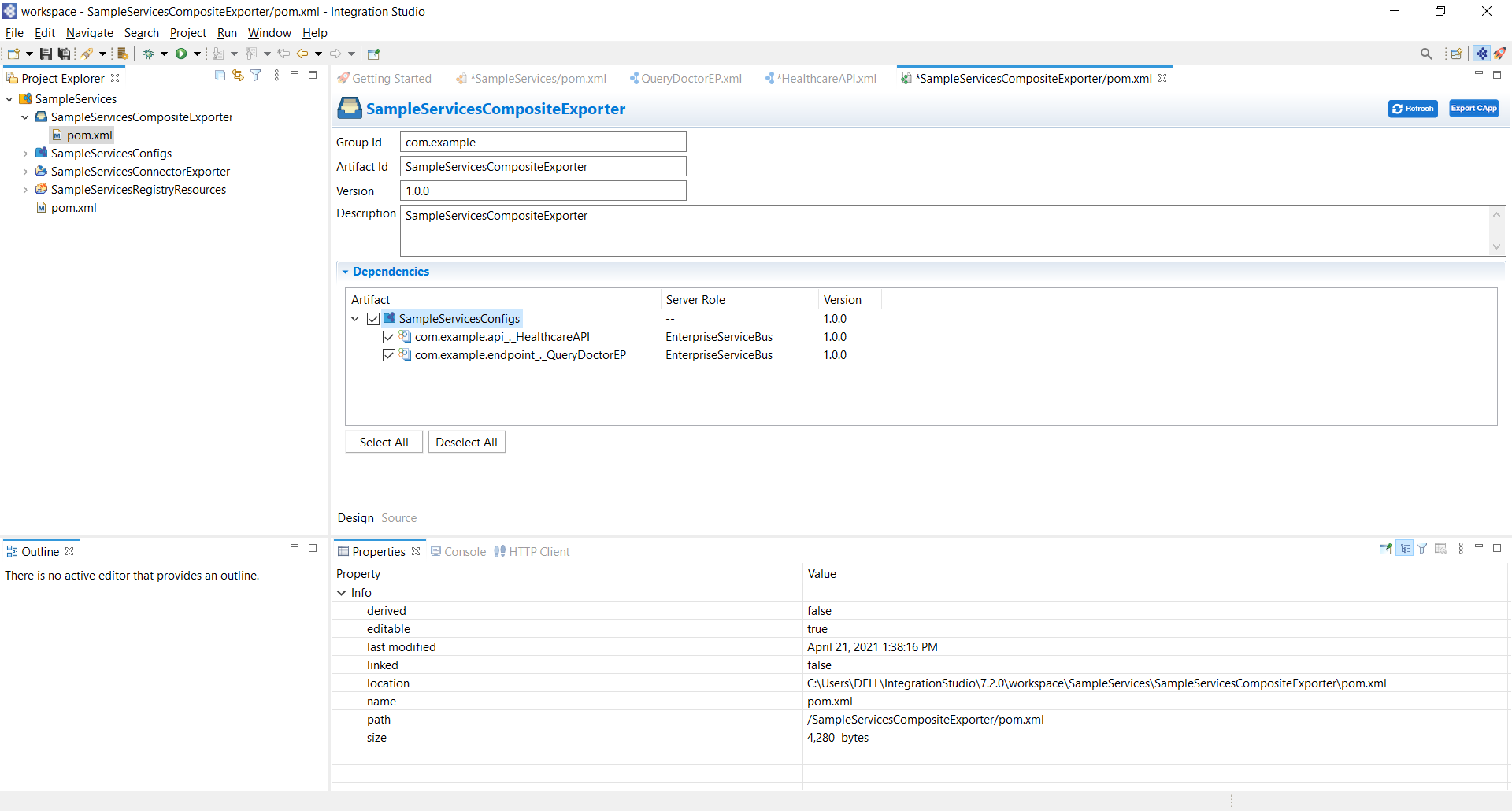
19. From the ‘Defined end points’ palate, drag the QueryDoctorEP endpoint and place it next to the Send mediator.



20. Drag another Send mediator to the out sequence (just below the original send mediator).

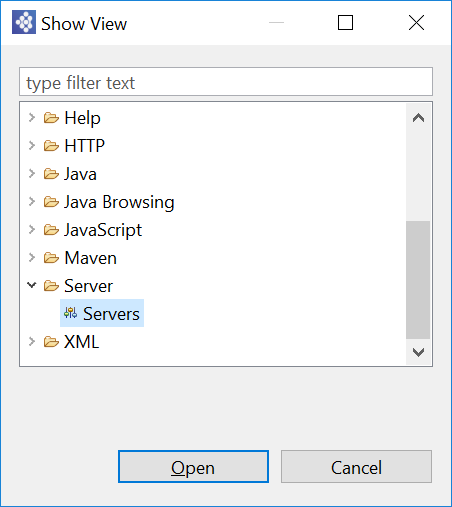


21. Open the pom.xml file under SampleServicesCompositeExporter and select the checkboxes to select all projects.



22. Save the pom.xml file.

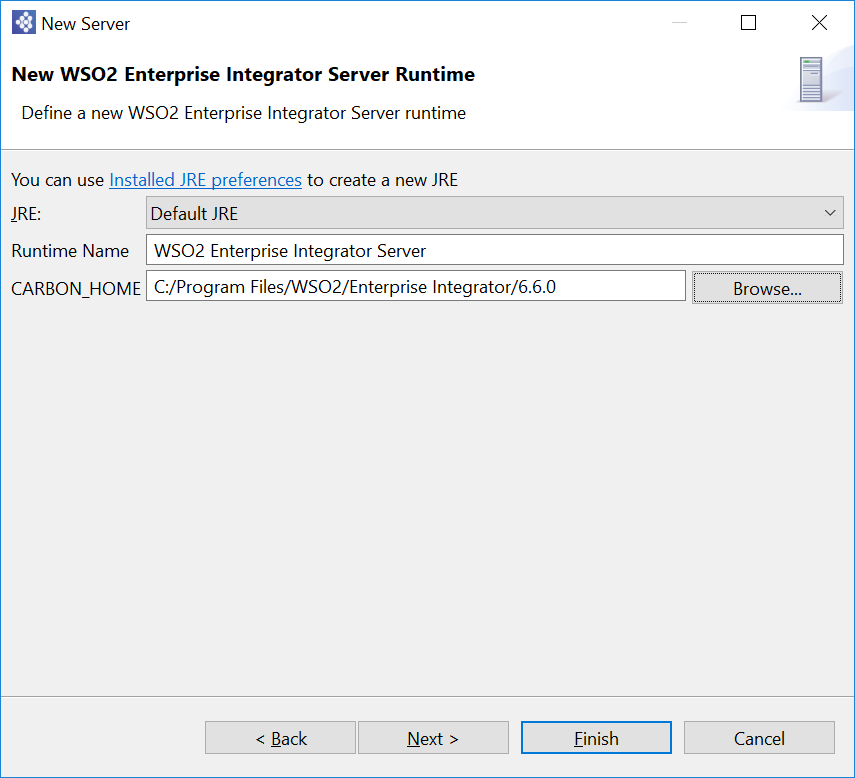
23. Now to deploy the integration project in the WSO2 Enterprise integrator, select Window -> Show View -> Other -> Server->Servers.



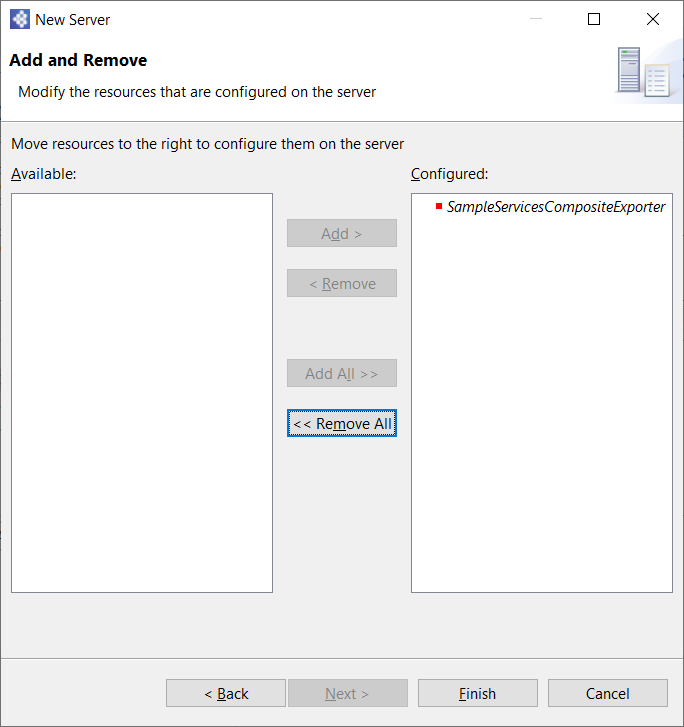
24. In the opened Servers view, Right click and select New -> Server.

25. Select WSO2 Enterprise Integrator 6.6.0.

26. Select Next. Under CARBON\_HOME, select the WSO2 Enterprise Integrator directory.



27. Select Next. Then select the SampleServicesCompositeExporter and add it to the right panel.



28. Click Finish.

29. Select the newly creator server configuration in the Servers window and then click the green colored play button in the server view to run the server.

30. Download the backend service from the following location.

<https://github.com/wso2-docs/WSO2_EI/blob/master/Back-End-Service/Hospital-Service-JDK11-2.0.0.jar>

31. Open the command line. Go to the jar file location and run it with the following command. This should run the service on port 9090 in localhost.

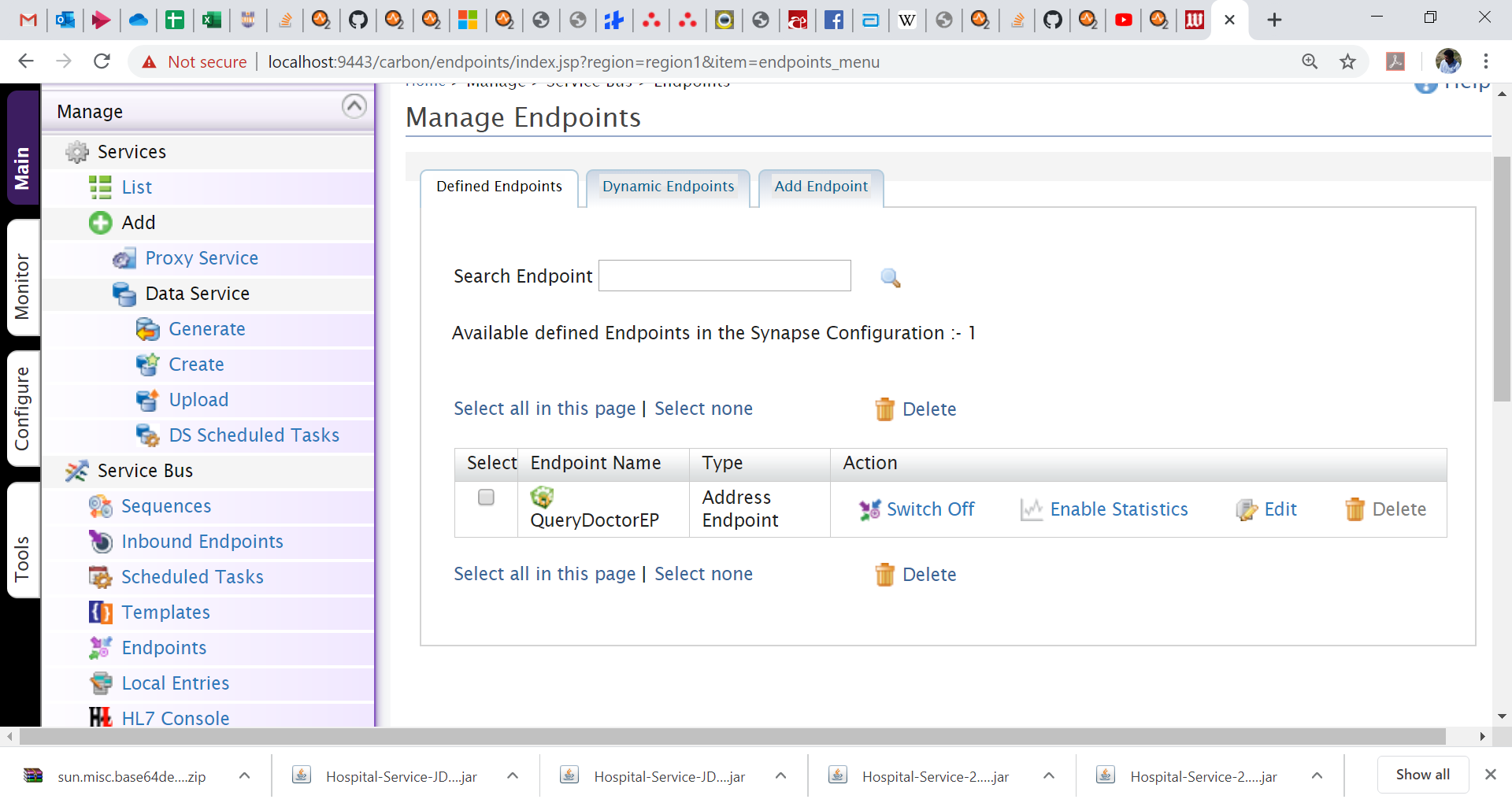
java -jar Hospital-Service-JDK11-2.0.0.jar

32. You should see a browser window open with the Enterprise Integrator management console. If not, go to this url in your browser.

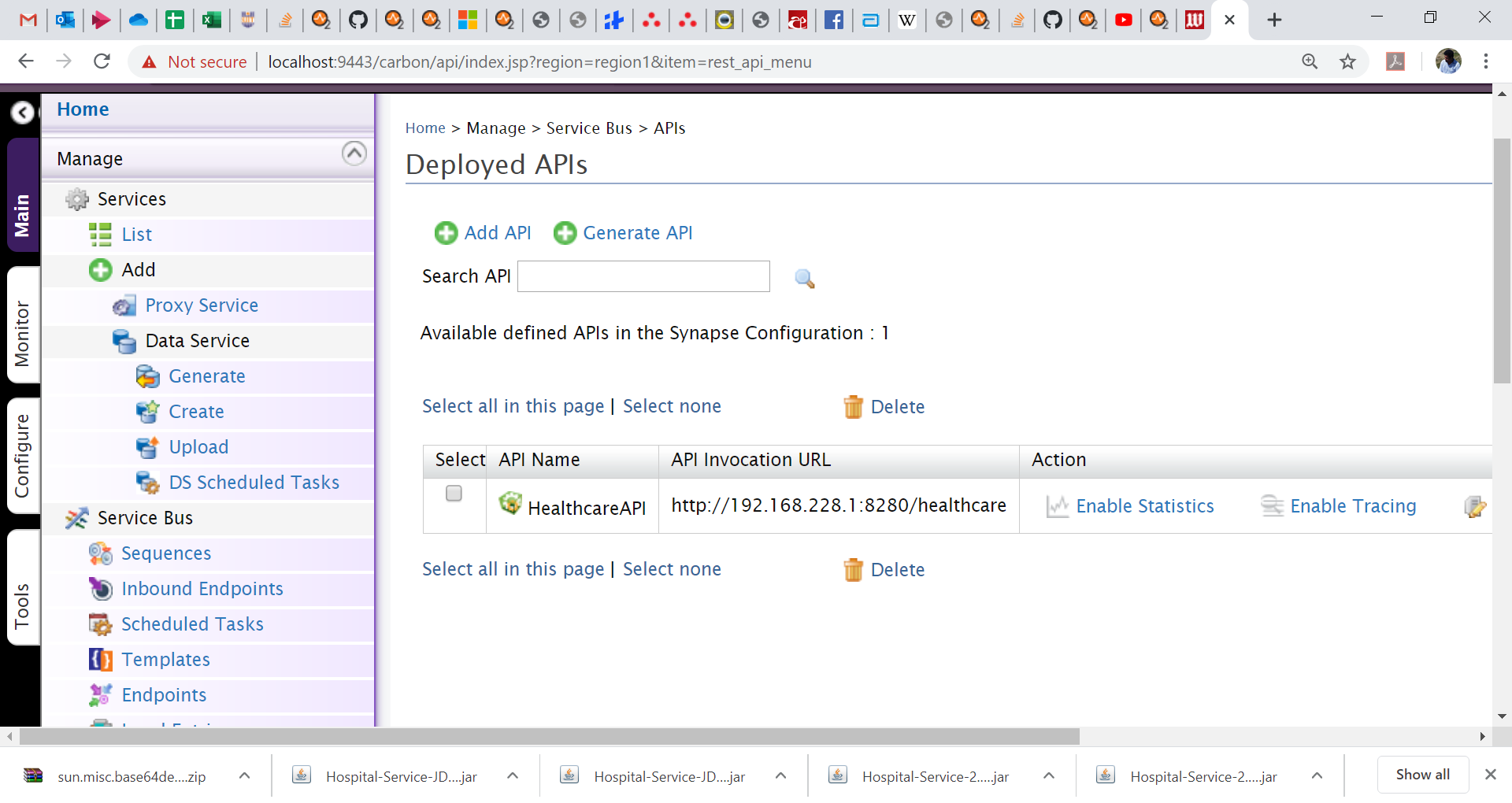
<https://localhost:9443/carbon/admin/login.jsp>

Type admin, admin to login using default credentials.

33. On the left panel, select Endpoints to see the endpoint deployed in the enterprise Integrator.



34. On the left panel, click the APIs to view the API deployed.



35. Copy the API url from here. Paste it in your browser window. Append the url as follows.

<http://192.168.228.1:8280/healthcare/surgery>

Note the actual IP might be different on your machine.

You should see a JSON object on the browser with the response.

You can try other categories such as,

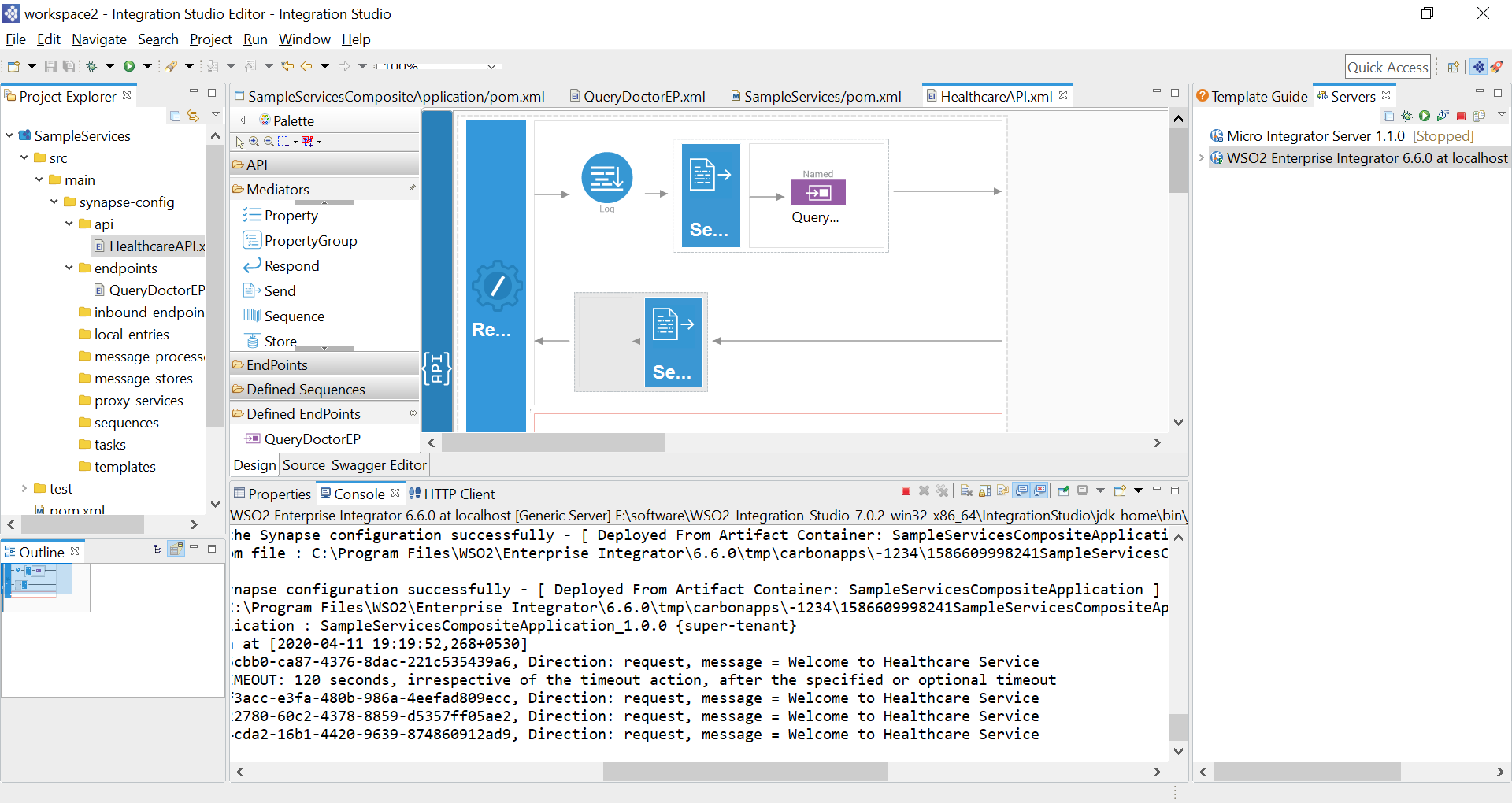
* cardiology
* gynaecology
* ent
* paediatric

to get the doctor info.

Note: Here you are connecting to the API and not the backend service directly. If you want, you can try to send a request to the backend service directly (just to observe), using the following url.

<http://localhost:9090/healthcare/surgery>

36. View the console of the integration studio to view the log message printed from the log mediator. This is printed by intercepting the request to the endpoint.



**Optional:**

**This is an optional exercise (not counted for marking). You may follow the following tutorial to understand how the switch mediator is used in WSO2 Enterprise integrator with multiple services.**

<https://docs.wso2.com/display/EI660/Routing+Requests+Based+on+Message+Content>

**Exercise 2 – Securing a REST Service with HTTPBasicAuth**

1. Extract the attached eclipse project and open it using Eclipse.
2. Run the project in Tomcat web server (refer last week’s lab sheet).
3. After the service is deployed, open the browser and go to the following url. You should see the output.

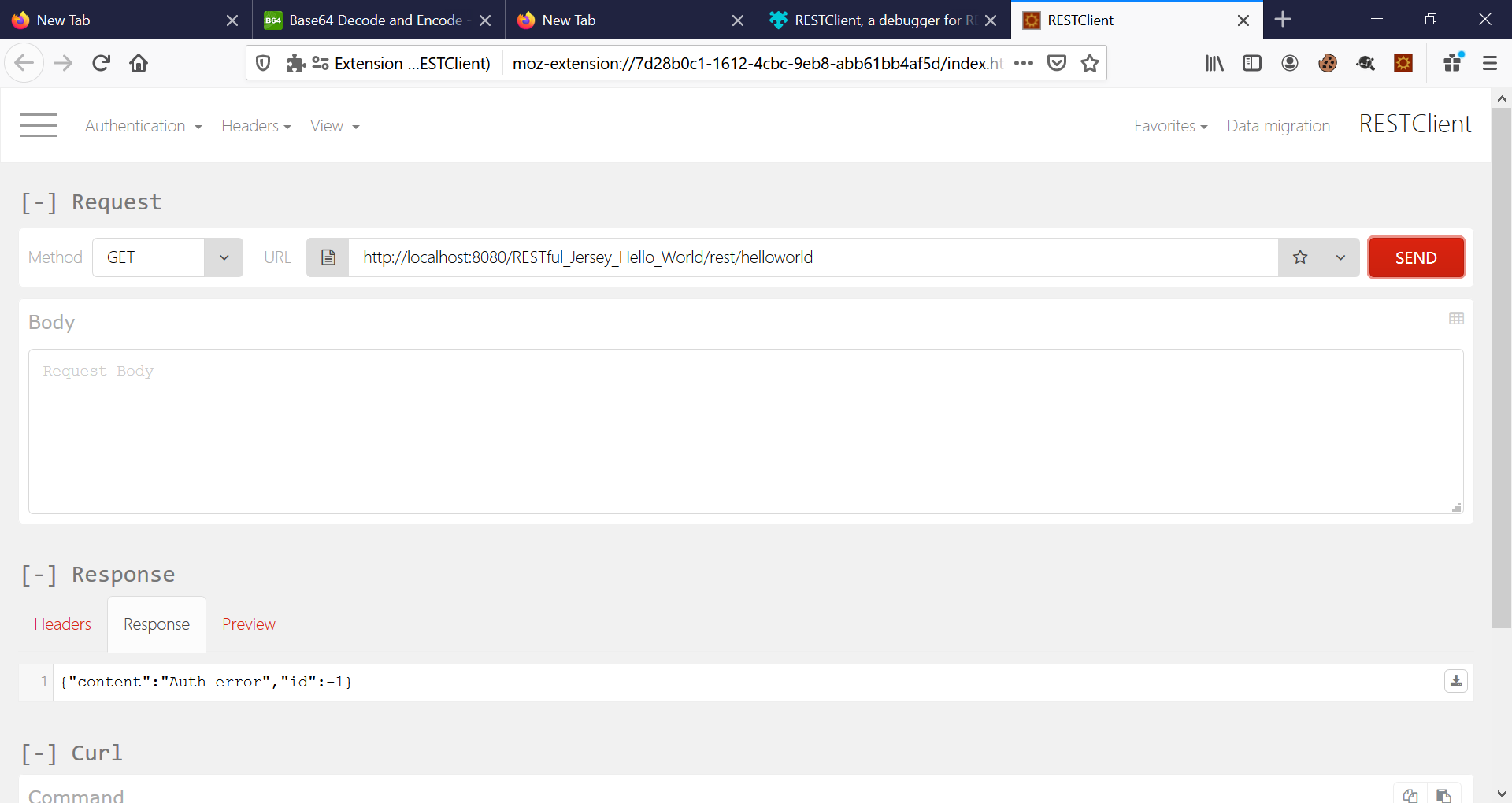
<http://localhost:8080/RESTful_Jersey_Hello_World/>

1. Open Firefox browser and install the RESTClient plugin (if you get an error, try installing an older version of the plugin).

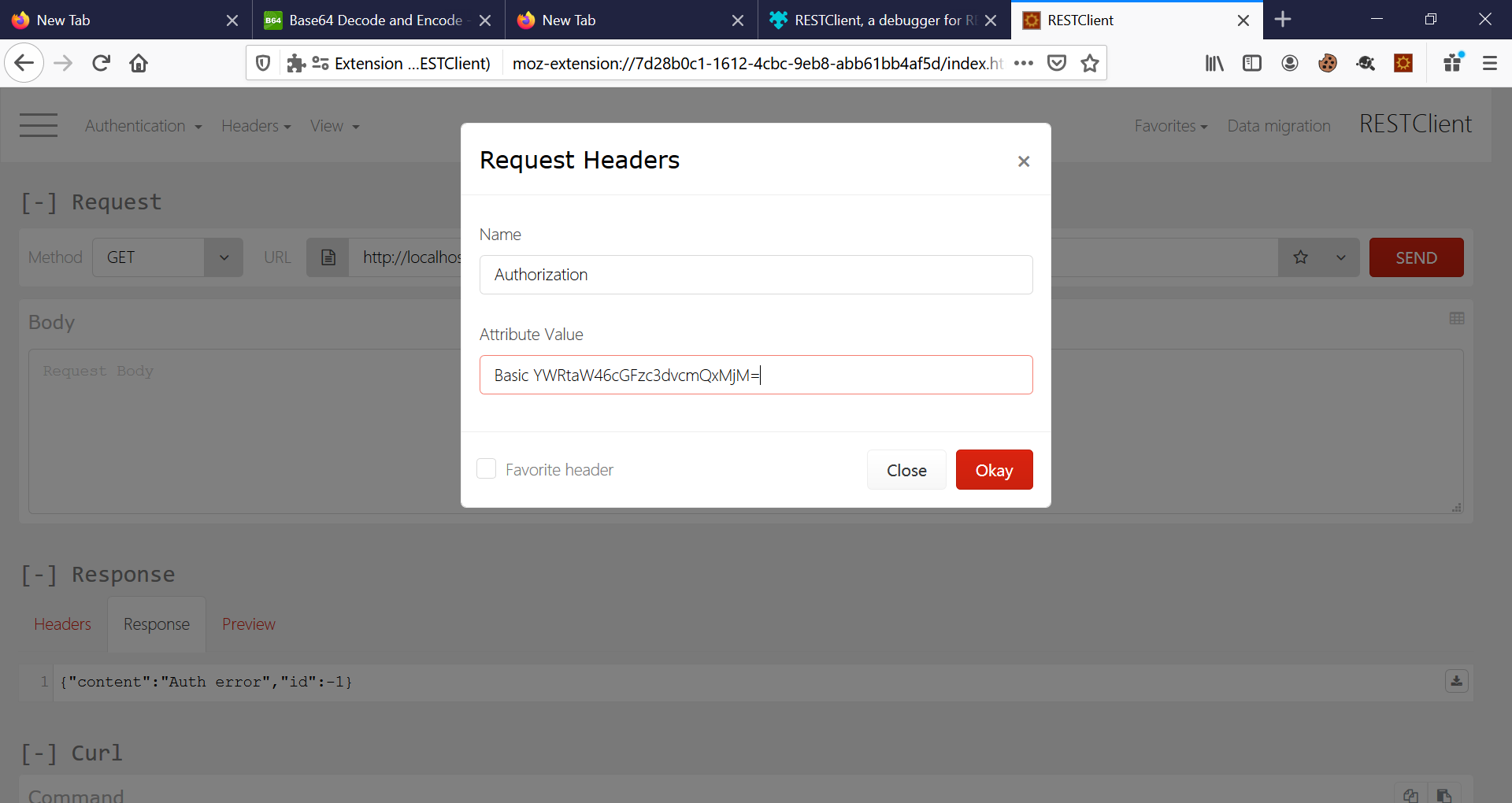
<https://addons.mozilla.org/en-US/firefox/addon/restclient/>

1. You should see the RESTClient button on your browser top right corner. Click it to open it. Paste the above url to see the output (this is the same url in the hello.js file). You should get an error in the response view.

<http://localhost:8080/RESTful_Jersey_Hello_World/rest/helloworld>



1. Go to the following web site and get the base64 encoded value for admin:password123 string.
2. Select Headers-> Custom Header and enter the value ‘Authorization’ and the value as ‘Basic <<encoded string>>’.



1. Now send the request and see whether you get the correct response.
2. You can try giving an incorrect Base64 encoded value in the Authorization header to see whether you get an error.
3. Go through the code in the index.html and hello.js files to see whether you can understand the code.
4. Go through the code in the Greetings.java and HelloWorld.java files to see whether you understand the code.

**Optional: This is an optional exercise and won’t be counted for marking. You can try to configure the tomcat web server to support HTTPS in addition to HTTP to see whether you can do the same authentication via HTTPS. This will be more secure as the user credentials and all messages will be encrypted.**

**Submission:** Upload the Eclipse integration project and the Dynamic web project used in the two exercises as a single zip file to the courseweb link. The zip file name should be your registration no. If the file is too large, upload to Dropbox and share the link in a text file. The text file name should be your reg. no.