Software Architecture

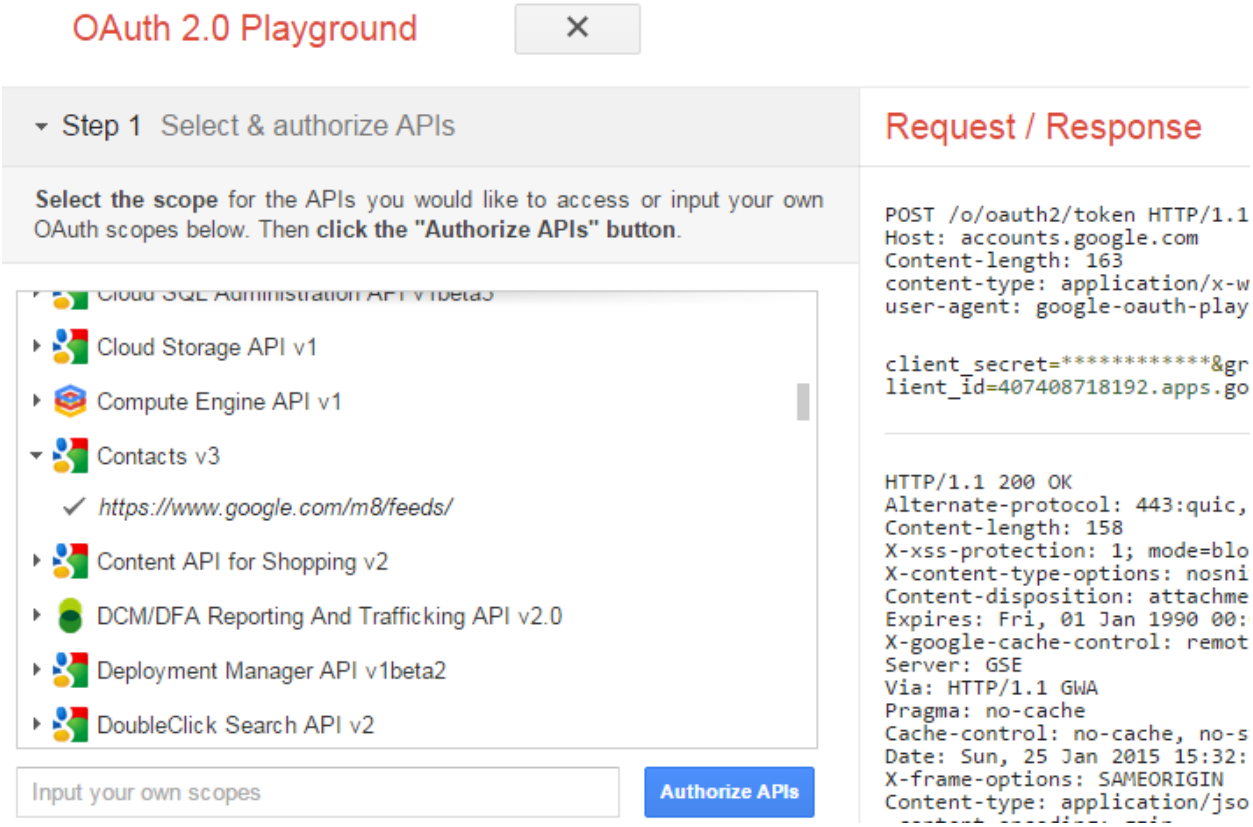
3rd Year – Semester 01

Enterprise Application Integration

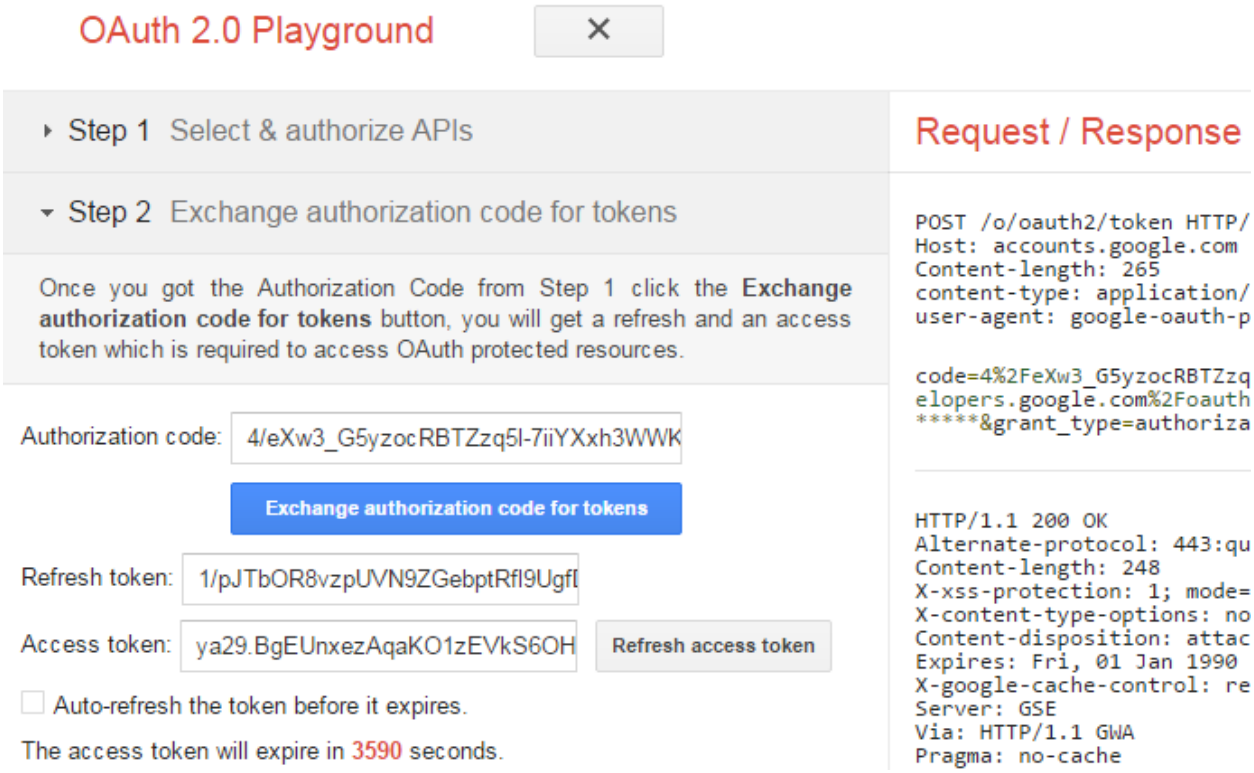
Lecturer - By Udara Samaratunge

Task 01 (Access Application Interfaces)

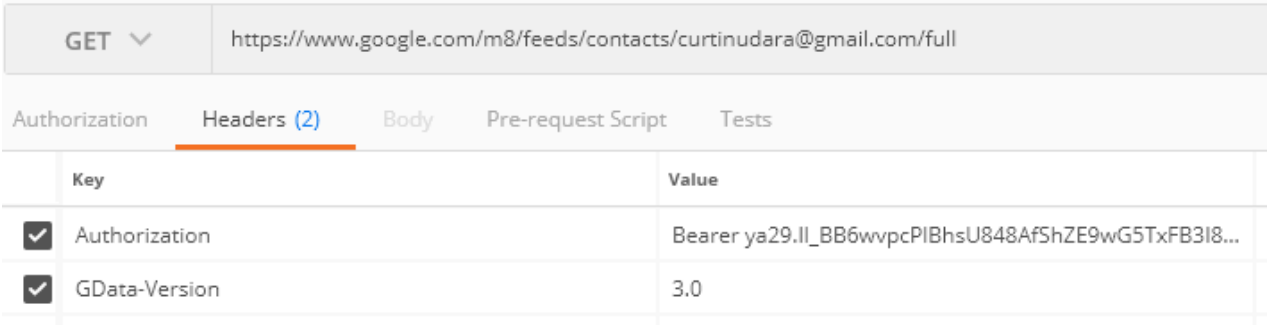
Search **Google Oauth 2.0 playground** and select Application as below. Then select the API and click **Authorize APIs** as below.



- Then in Step 2 section click "**Exchange authorization token**" button as below.
- You can refresh click **Refresh token** when it is going to expire.
- It generate **Access token**. Copy that to access the API.
- Then copy that access toke and use that for Postman Package and send API request as below for Google contact API.



- For that you have to add **Authorization header** as well **GData –Version = 3.0** as well.
- When you add **Authorization header** value should be taken from **Google Oath 2.0 Playground** (Access token) and Append **"Bearer+ " "+{Access token}"** .
- You can use API URL request as below. For **successful response** you should get **200 OK** status value.



GET https://www.google.com/m8/feeds/contacts/curtinudara@gmail.com/full Params Send

Cookies Headers (13) Test Results Status: 200 OK

Pretty Raw Preview XML


```xml
<?xml version='1.0' encoding='UTF-8'?>
<feed xmlns='http://www.w3.org/2005/Atom' xmlns:gd='http://schemas.google.com/g/2005' xmlns:batch='http://schemas.google.com/gdata/batch' x
    :gContact='http://schemas.google.com/contact/2008' xmlns:openSearch='http://a9.com/-/spec/opensearch/1.1/' gd:etag='&quot;FVMWRApBDS1gi
    ..&quot;'>
    <id>curtinudara@gmail.com</id>
    <updated>2020-03-06T11:11:09.069Z</updated>
    <category scheme='http://schemas.google.com/g/2005#kind' term='http://schemas.google.com/contact/2008#contact'/>
    <title>udara curtin's Contacts</title>
    <link href='http://www.google.com/' rel='alternate' type='text/html'/>
    <link href='https://www.google.com/m8/feeds/contacts/curtinudara%40gmail.com/full' rel='http://schemas.google.com/g/2005#feed' type
        ='application/atom+xml'/>
    <link href='https://www.google.com/m8/feeds/contacts/curtinudara%40gmail.com/full' rel='http://schemas.google.com/g/2005#post' type
        ='application/atom+xml'/>
    <link href='https://www.google.com/m8/feeds/contacts/curtinudara%40gmail.com/full/batch' rel='http://schemas.google.com/g/2005#batch' t
        ='application/atom+xml'/>
    <link href='https://www.google.com/m8/feeds/contacts/curtinudara%40gmail.com/full?max-results=25' rel='self' type='application/atom+xm
    <author>
        <name>udara curtin</name>
        <email>curtinudara@gmail.com</email>
    </author>
    <generator version='1.0' uri='http://www.google.com/m8/feeds'>Contacts</generator>
    <openSearch:totalResults>1</openSearch:totalResults>
    <openSearch:startIndex>1</openSearch:startIndex>
    <openSearch:itemsPerPage>25</openSearch:itemsPerPage>
```


Task 02 (Uses Yodiz API)

- Navigate to URL "https://app.yodiz.com/user/signup.vz" and create an account.
- (Save the given **[username],[email address],[password]** for future use.

Base URL = **"https://app.yodiz.com/api/rest/v1/"**

Yodiz API Url = https://www.yodiz.com/api/#introduction

Authenticate Yodiz API

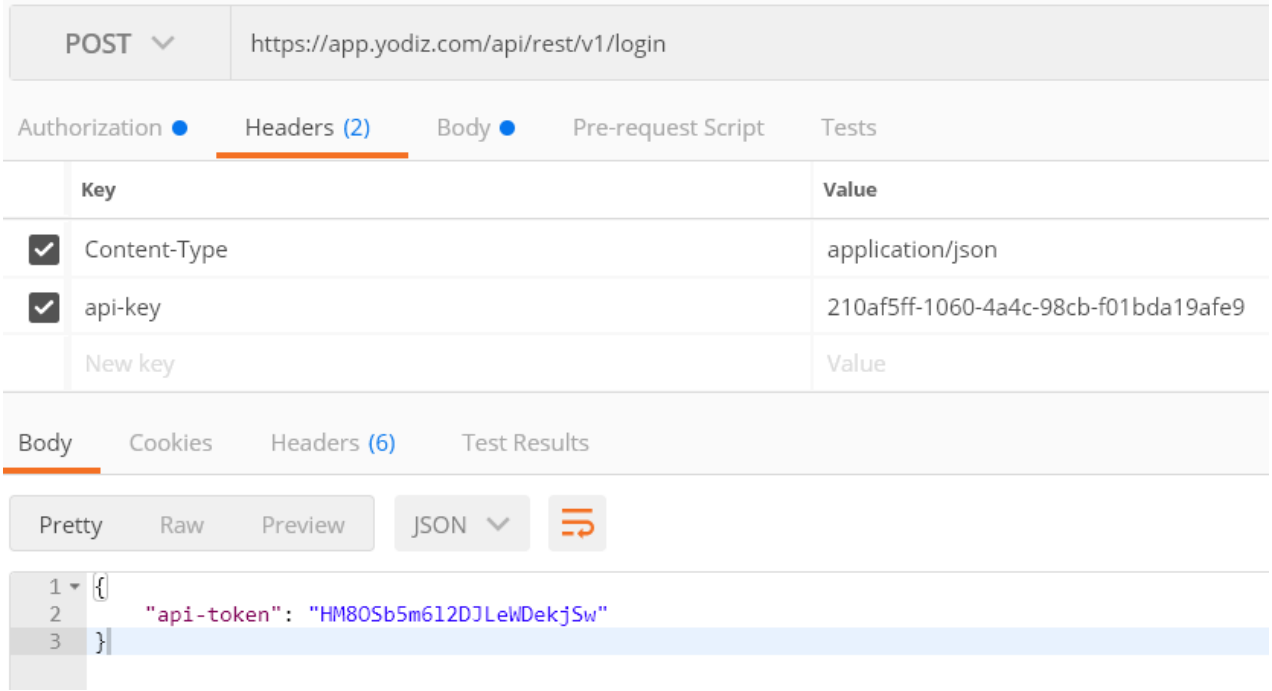
Send Post request = https://app.yodiz.com/api/rest/v1/login

Add the following headers

api-key = 210af5ff-1060-4a4c-98cb-f01bda19afe9

Content-Type = application/json

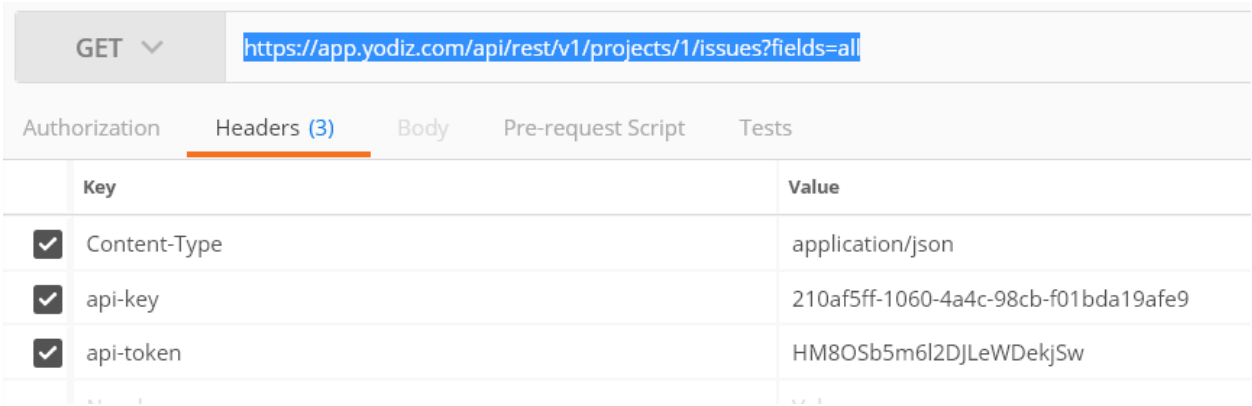
You will get the API token as the response. Then use that api-token as a header to send the rest of other requests



Then use the following Get request to get all project specific issues

GET Request = https://app.yodiz.com/api/rest/v1/projects/1/issues?fields=all

- Use following API keys and tokens as headers to get all issues



Send the request and check the response. You should get **JSON response** as below. If you send correct request you will get **200OK status** as follows.

GET https://app.yodiz.com/api/rest/v1/projects/1/issues?fields=all

Body Cookies Headers (7) Test Results

Pretty Raw Preview JSON


```
[
    [
        {
            "id": 125,
            "title": "Does not work",
            "status": {
                "code": "err-st-30",
                "narrative": "Resolved"
            },
            "severity": {
                "code": "err-svr-40",
                "narrative": "Critical"
            },
            "release": {
                "id": 27
            },
            "sprint": {
                "id": 91
            },
            "effortEstimate": 0,
            "effortRemaining": 0,
            "effortLogged": 0,
            "highPriority": false,
            "dueDate": "2017-01-04T00:00:00.000+0000",
            "mscwPriority": {
                "code": "err-pr-10"
            },
            "tags": [
                {
                    "id": 113,
                    "title": "graphics",
                    "type": 0
                }
            ],
            "createdBy": {
```


Try out providing ticket ID as below.

Send the POST request as follows to create new user story point in the API.

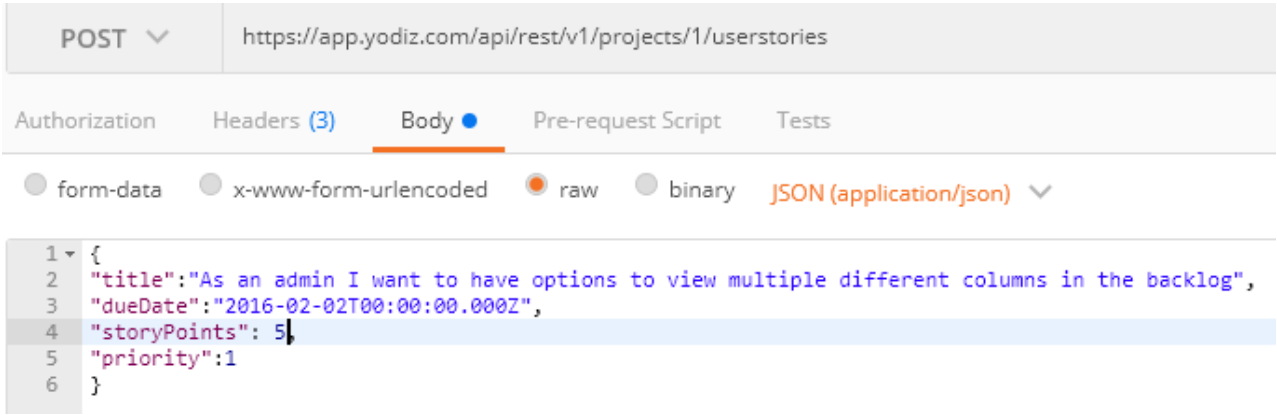
POST https://app.yodiz.com/api/rest/v1/projects/1/userstories

Authorization Headers (3) Body Pre-request Script Tests

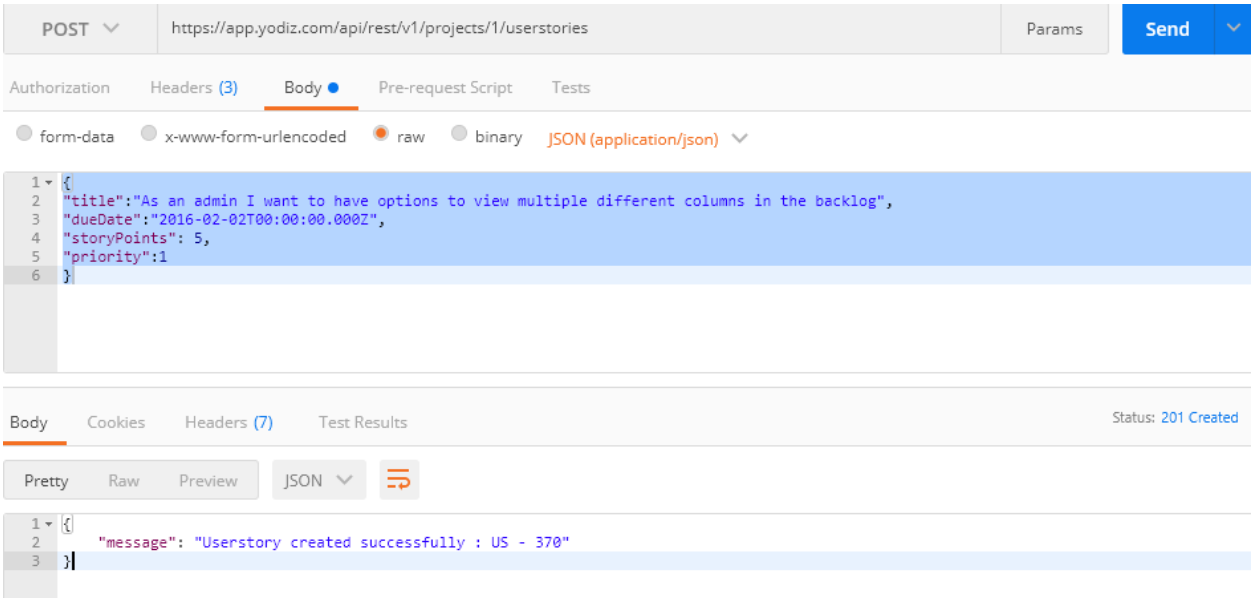
Key	Value
☑ api-token	HM8OSb5m6I2DJLeWDekjSw
☑ api-key	210af5ff-1060-4a4c-98cb-f01bda19afe9
☑ Content-Type	application/json
New key	Value

Then in the POST request body you should add following JSON body. In the Postman body select the type raw and try out the following request.

API URL = https://app.yodiz.com/api/rest/v1/projects/1/userstories



If it is successful you will get **http status code 201** and response message will be User story created successfully.

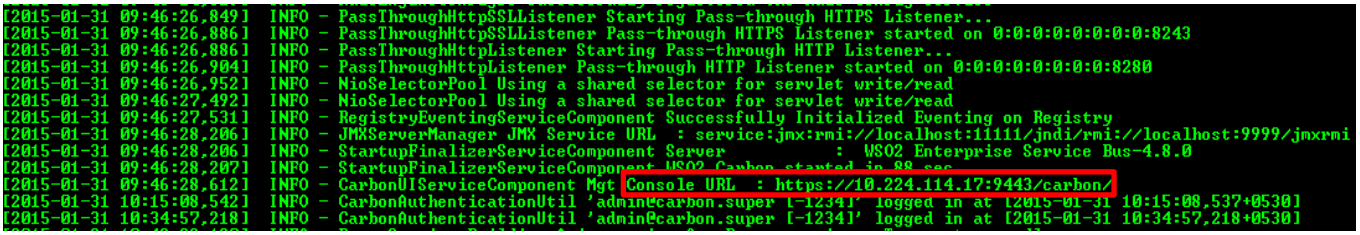


Task 03 (Use WSO2 ESB to test SOAP web service)

In this task you should access **3rd party web service** through **WSO2 ESB** and you are going to test the service through **ESB**. For that you are going to access **Calculator Service** hosted in http://www.dneonline.com/calculator.asmx. This is a SOAP web service you are going to get wsdl of it.

1) Navigate to **\wso2esb-4.8.0\wso2esb-4.8.0\bin** location and **run wso2server.bat** file. If you are in **Unix** environment run **wso2server.sh** file [**./wso2server.sh]**

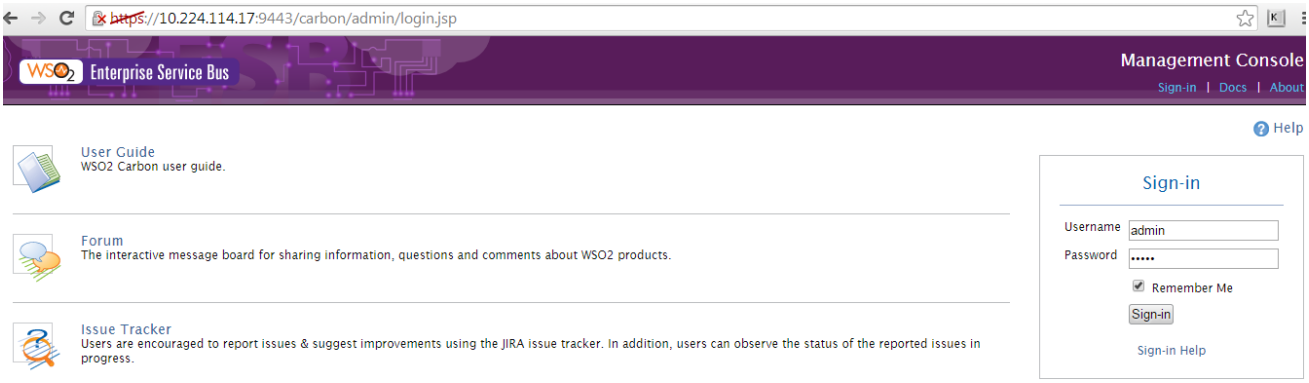
2) Once this is up and running in console it prompts **Management console Url** as below. Copy the **url** and proceed through your browser. [**Accept the security exception** prompts in browser since **https**] Continue up to **login console**.



3) WSO2 ESB management console login credentials are as below. Login to Management console using them

Username = admin

Password = admin

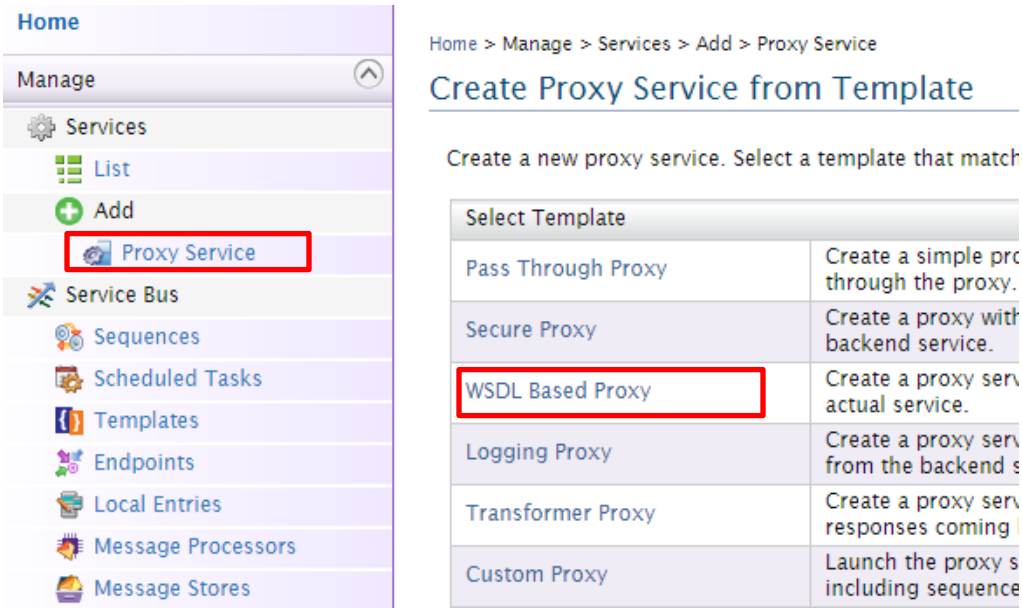


4) Now you are going to access 3rd party web service already hosted in remotely. For that **Navigate to URL** = http://www.dneonline.com/calculator.asmx?WSDL

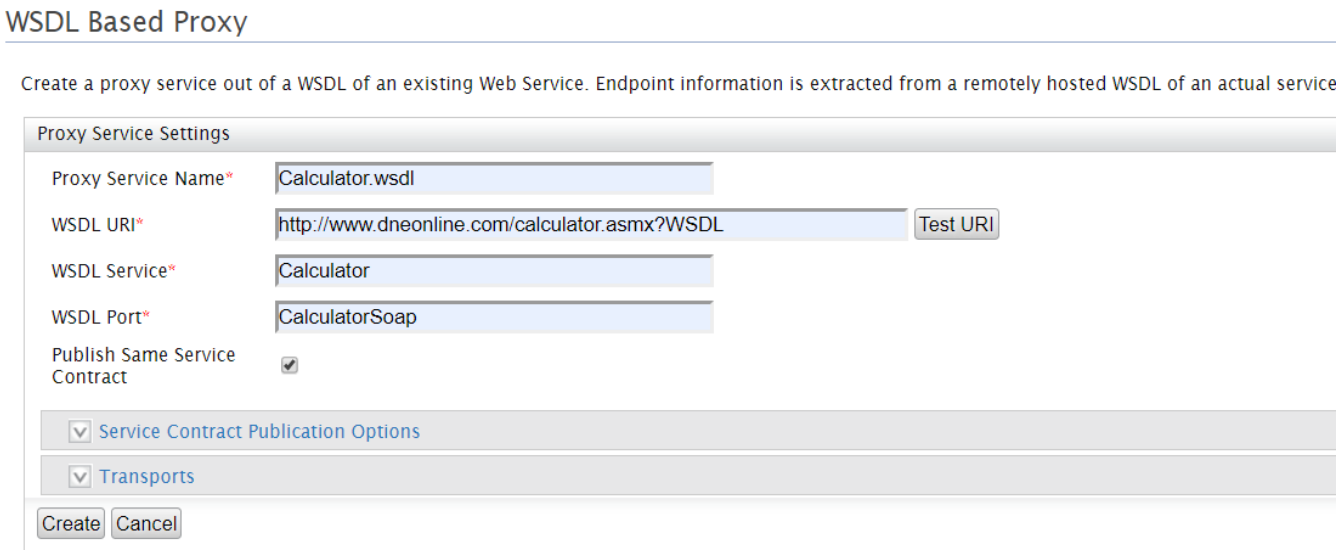
5) Go to view page source and copy whole WSDL content and save it as **Calculator.wsdl** in **wso2esb-4.8.0\wso2esb-4.8.0\repository\samples\resources\proxy** location of your **ESB.** This **wsdl** file would act as a **proxy service** for **ESB** to connect **remote service**.

Create Proxy service (WSDL based proxy)

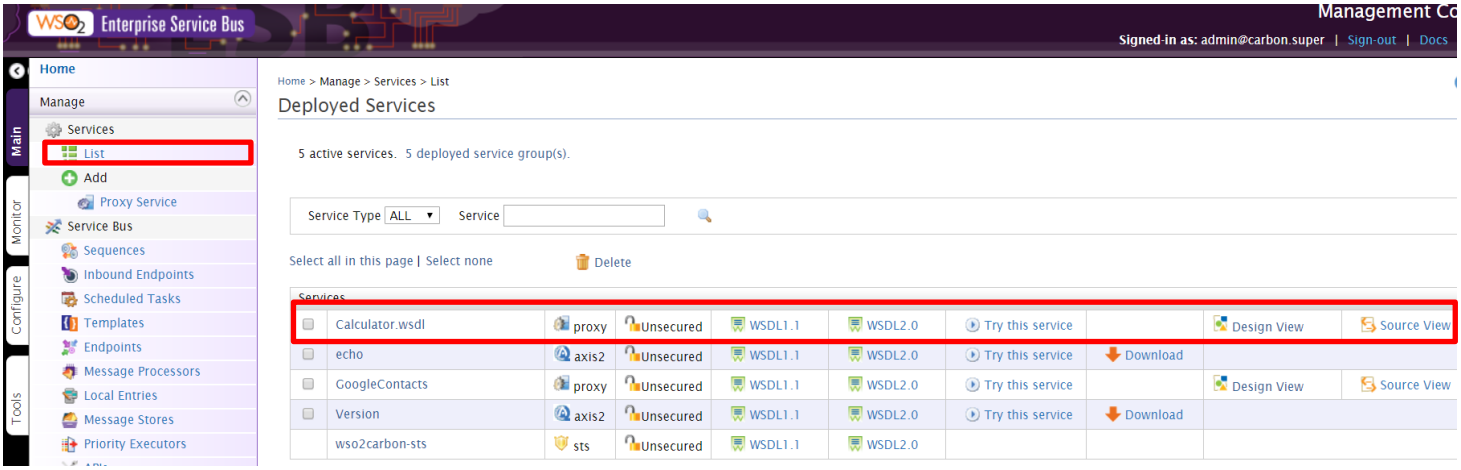
6) In management console click **Proxy service => WSDL based proxy.** Then you have to fill following **Proxy service settings details**. Click Test URI button and check the service availability. Refer below screenshots.



7) Once details are given click **Create Button**.



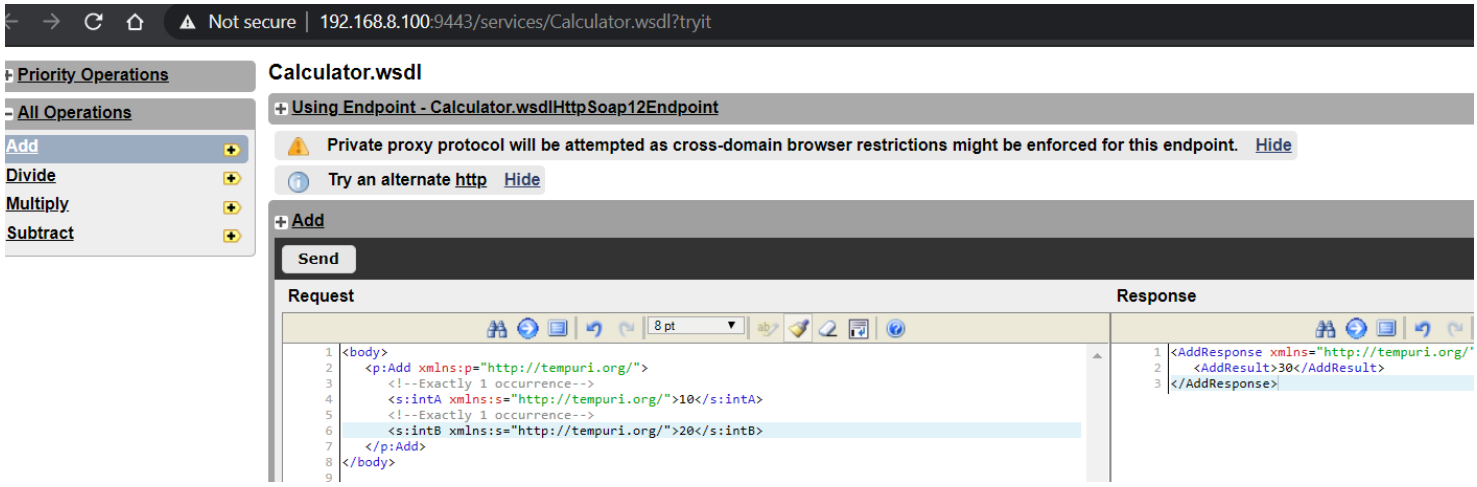
8) Once you created proxy service you should be able to see it under list of proxies. Then click **List** under services tab in main menu.



9) It generates proxy service as follows to your remote service

```xml
<?xml version="1.0" encoding="UTF-8"?>
<proxy xmlns="http://ws.apache.org/ns/synapse"
       name="Calculator.wsdl"
       startOnLoad="true"
       statistics="disable"
       trace="disable"
       transports="https,http">
   <target>
      <outSequence>
         <send/>
      </outSequence>
      <endpoint>
         <wsdl port="CalculatorSoap"
               service="Calculator"
               uri="http://www.dneonline.com/calculator.asmx?WSDL"/>
      </endpoint>
   </target>
   <publishWSDL uri="http://www.dneonline.com/calculator.asmx?WSDL"/>
   <description/>
</proxy>
```

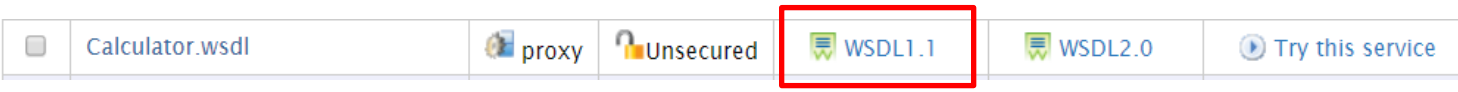
10) In list of proxy services select and click **Try This service.** Now modify SOAP request with adding values and monitor the response. You will get relevant response.



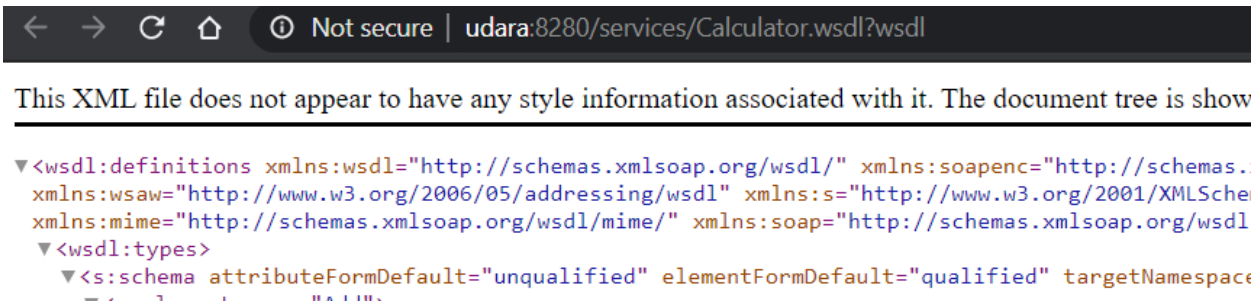
Task 04 (Use SOAP UI to send request for ESB)

1) Open SOAP UI version 4.5.X. Then click **soapui.bat** file (in **Windows**) or run **./soapui.sh** for **UNIX environment**. File located = **soapui-4.5.1-win32-standalone-bin\soapui-4.5.1\bin**

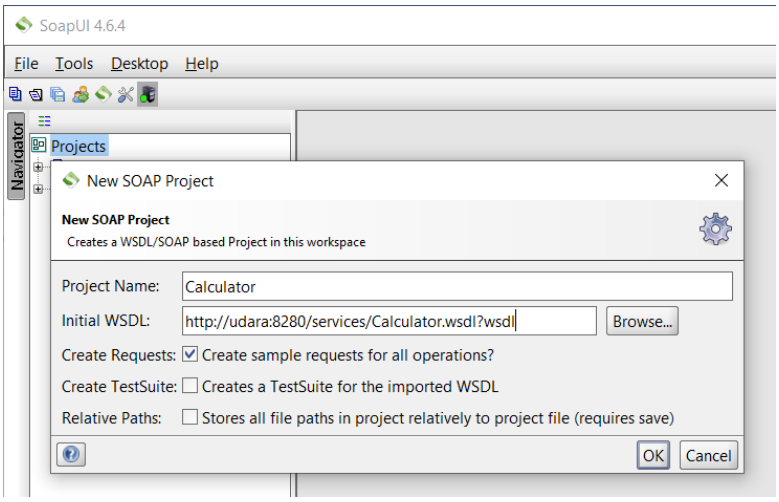
2) Now in **WSO2 ESB management console** select the relevant proxy service and click **WSDL.1.1** as below.



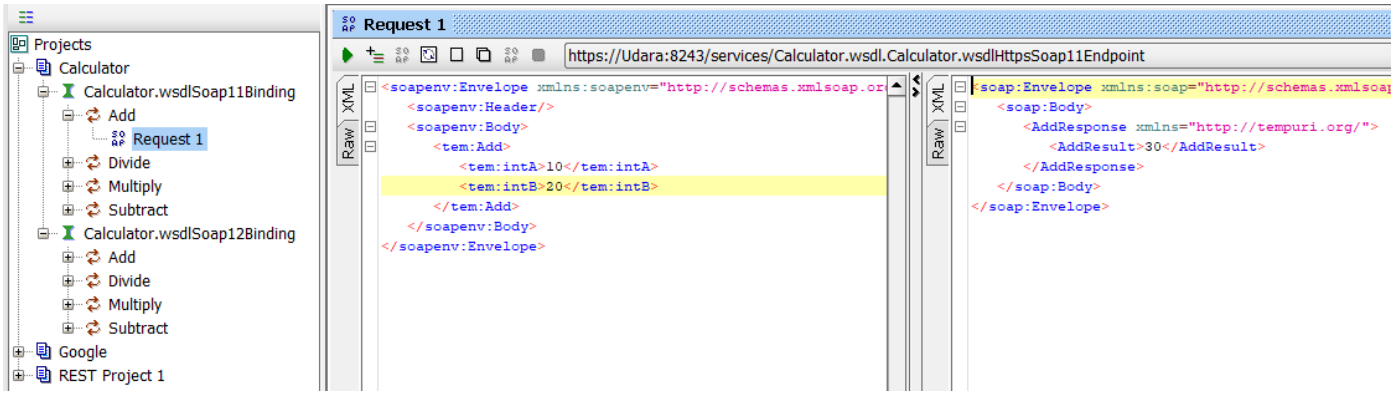
3) Then copy the **WSDL URL** in browser. This is the service for you are going to access to connect Proxy service. Now **SOAP UI** sends the request to **ESB** and **ESB** sends the request to End point.

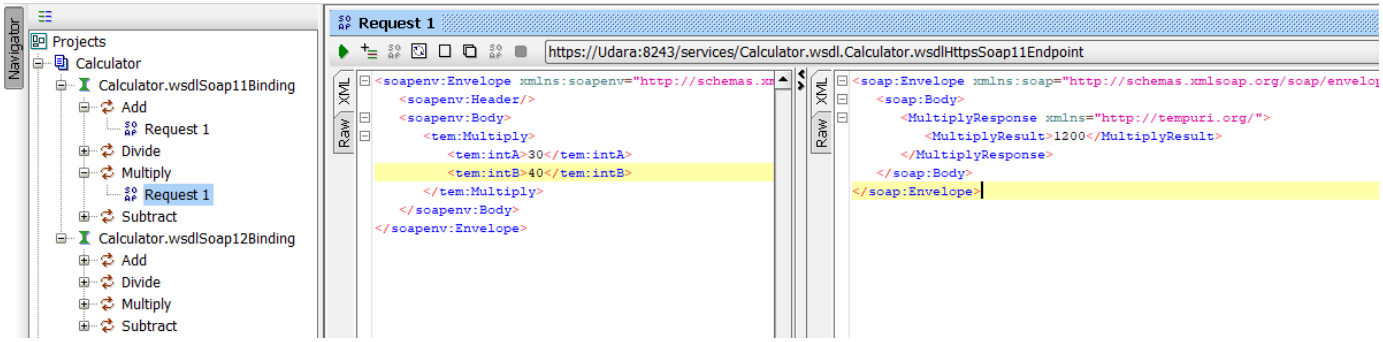


4) Copy the WSDL url by selecting WSDL1.1 in management console of ESB. Then in SOP UI, select **File => New SOAP UI project** and paste URL as below.



5) It automatically get project name as Temperature and create SOAP project as below. Then in the project **select request1** and **double click**. **SOAP request** automatically get generated. Modify the request with suitable values. You will get below response.





The End