

Creating a JAX-WS Webservice Using IBM RAD and WebSphere 6.1

Creating a JAX –WS Web Service Using IBM RAD and WebSphere 6.1

I will discuss today creating a jax-ws web service using IBM RAD and websphere 6.1 .The theoretical parts I will cover in some other article. Today I am going to just show you all that how a jax-ws web service can be created using IBM RAD and websphere 6.1 and believe me if you are in learning phase then you will feel very happy to create web service of your own 😊. I will write in steps with the screen shots attached so that it will be easy for you.

Step 1:

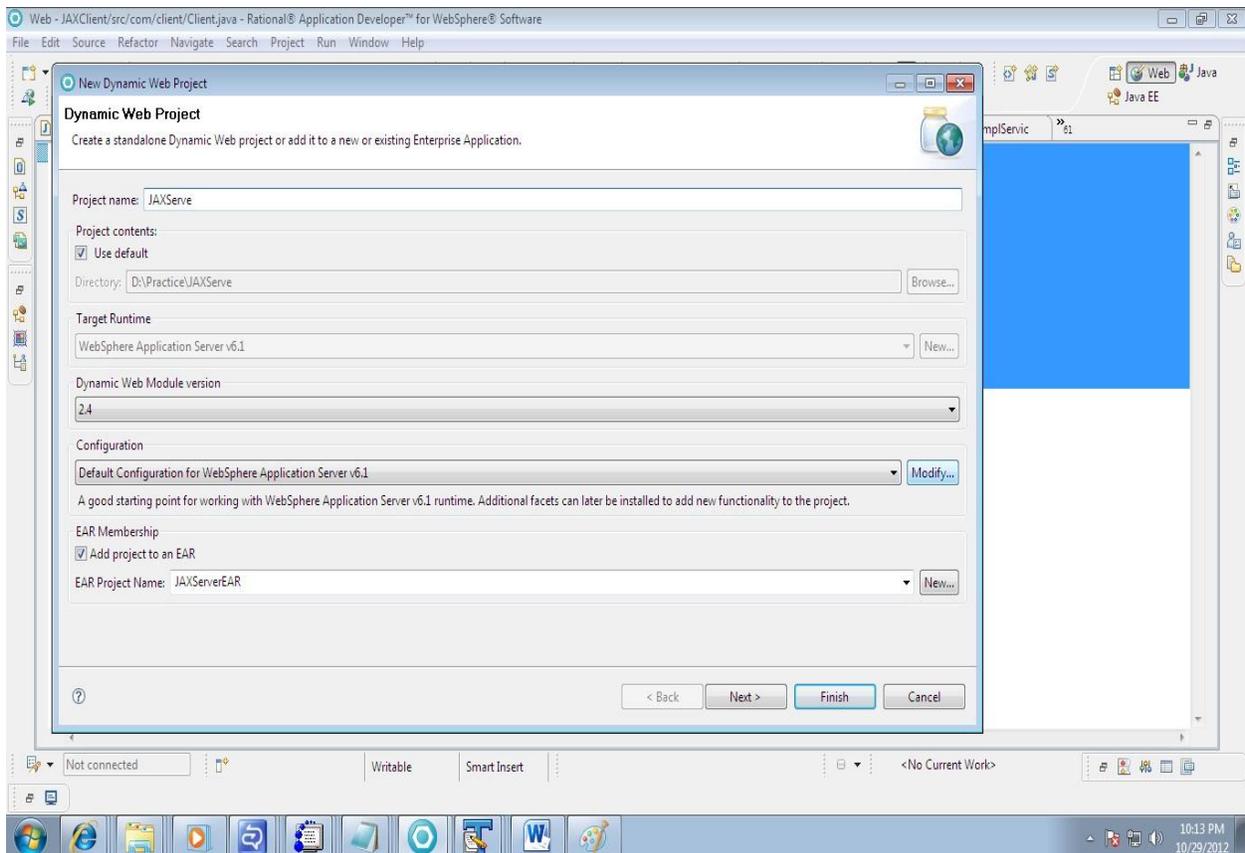
Create a new IBM RAD Workspace.

Step 2:

Create a new Dynamic Java Project named "**JAXServer**". We need to enable Webservice Feature pack of WAS 6.1. Follow the below snapshot.

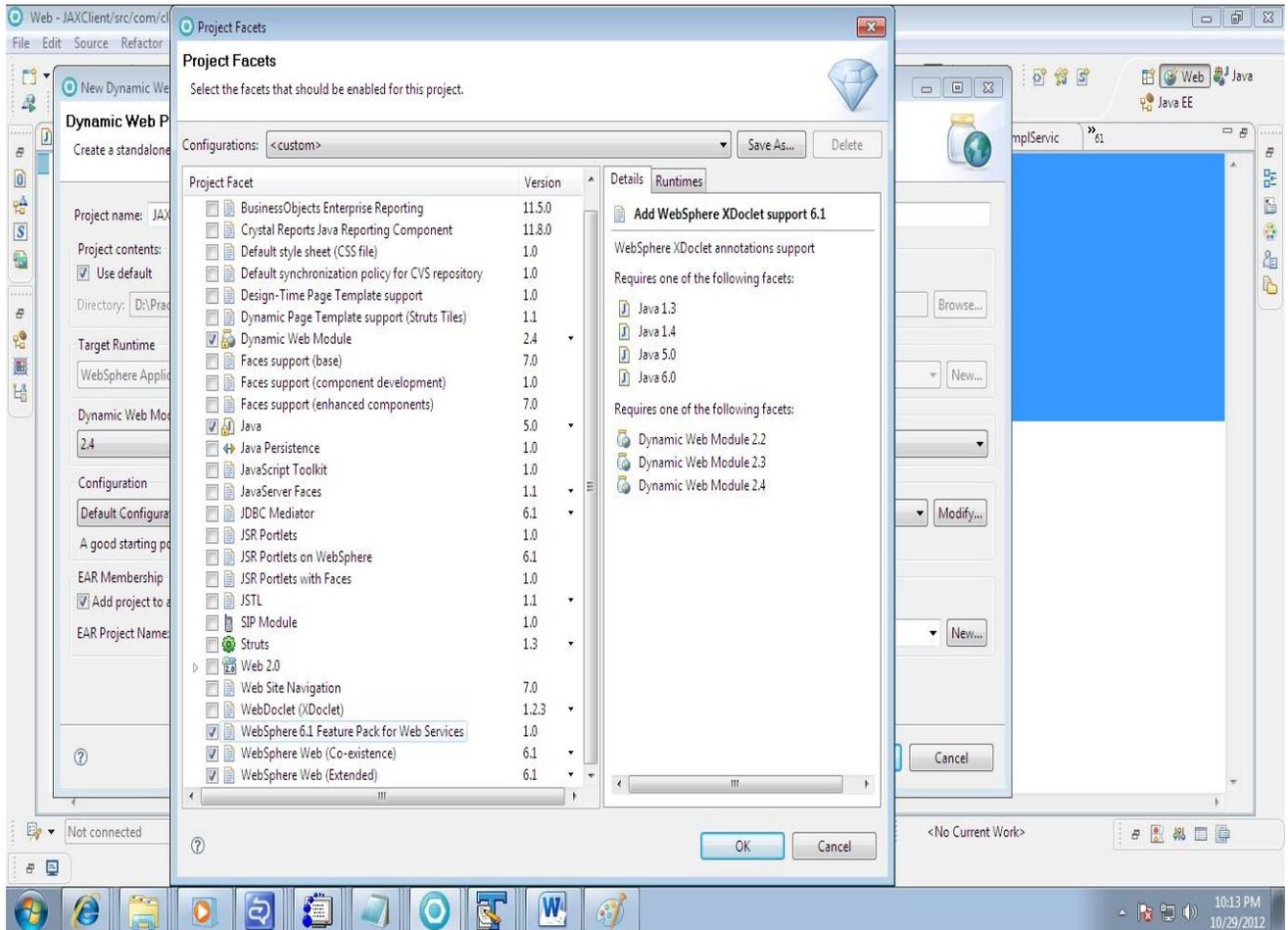
Enter the Project name as JAXServer.

In Configuration Click on Modify Button



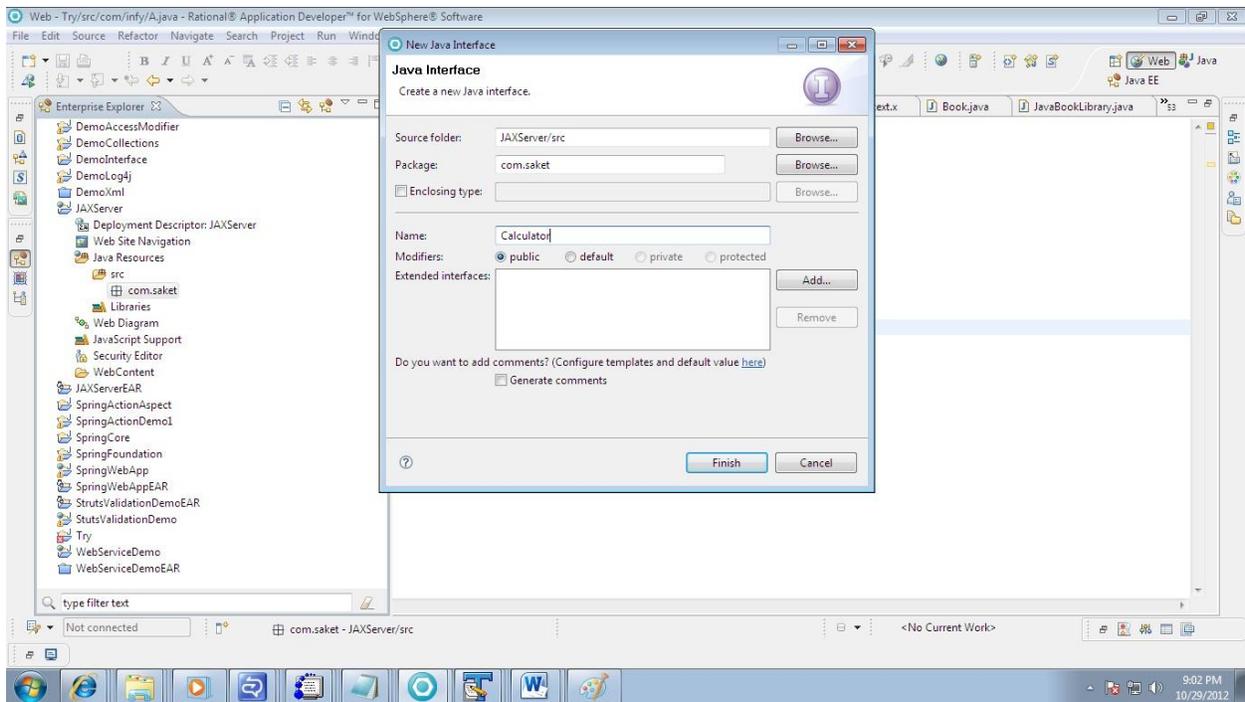
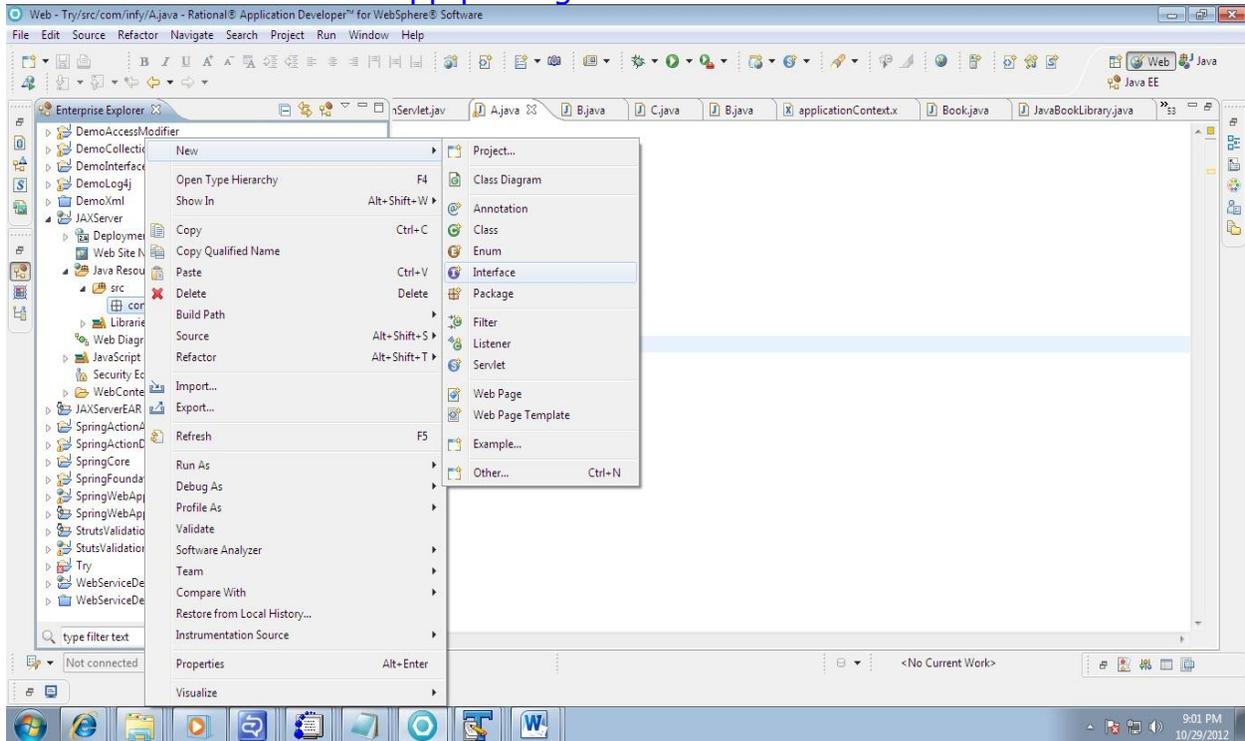
Creating a JAX-WS Webservice Using IBM RAD and WebSphere 6.1

Select WebSphere 6.1 Feature Pack for WebServices and then press Ok. And then Click on Next and Finish.



Creating a JAX-WS WebService Using IBM RAD and WebSphere 6.1

Step 3: Create a new endpoint interface for defining our endpoint class. Right click in the **src** folder and add new **interface** and name it as **"Calculator"** in the **com.pp** package



Creating a JAX-WS WebService Using IBM RAD and WebSphere 6.1

Step 4:

Now we will define an interface for our calculator program. Write the below code in the Calculator interface.

```
package com.pp;

import javax.jws.WebMethod;
import javax.jws.WebService;

@WebService
public interface Calculator {

    @WebMethod
    public int add(int var1, int var2);
}
```

Step 5:

Now we should define an implementing class for our interface. We should now create a class in the same package and name it as "**CalculatorImpl**". This class will implement the **Calculator** interface and override the **add** method. The code that must be written inside **CalculatorImpl** class is given below.

```
package com.pp;

import javax.jws.WebService;

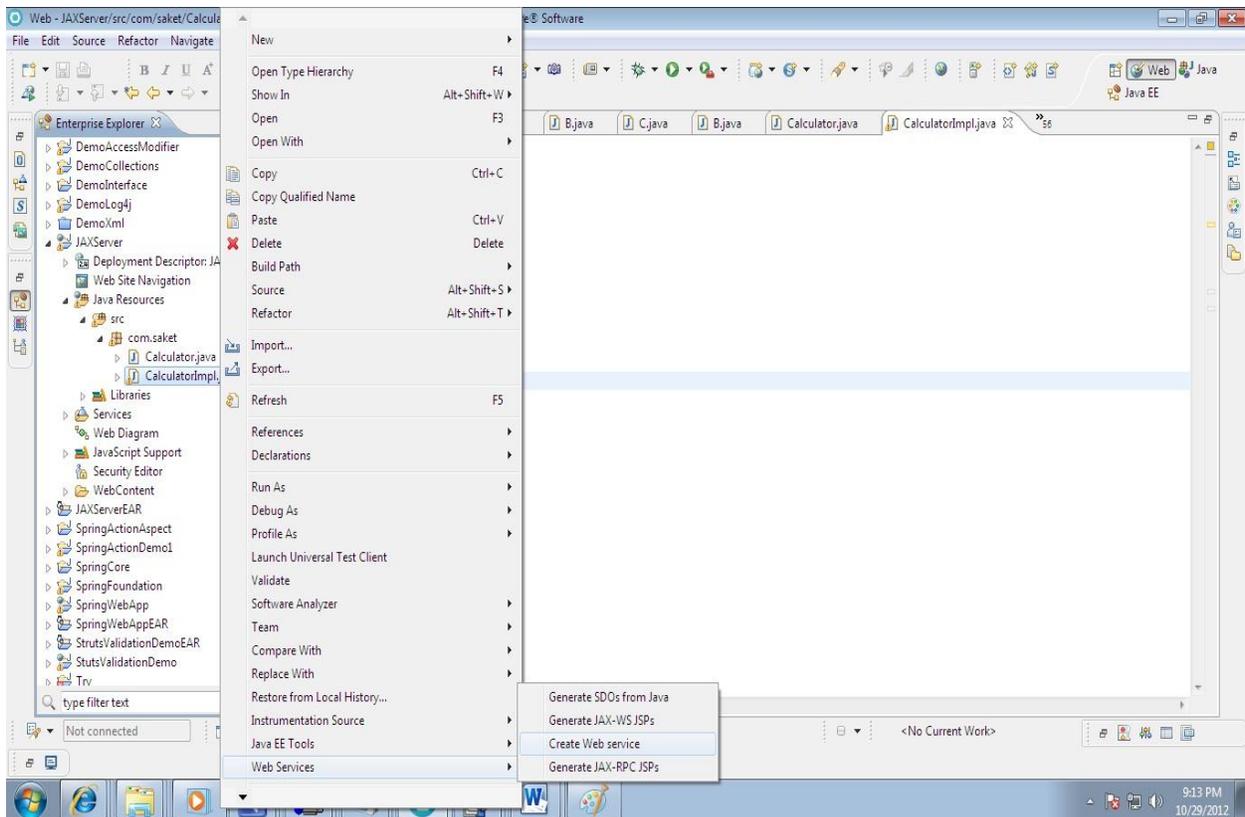
@WebService(endpointInterface=" com.pp.Calculator")
public class CalculatorImpl implements Calculator{

    public int add(int var1, int var2) {
        return var1+var2;
    }
}
```

Creating a JAX-WS WebService Using IBM RAD and WebSphere 6.1

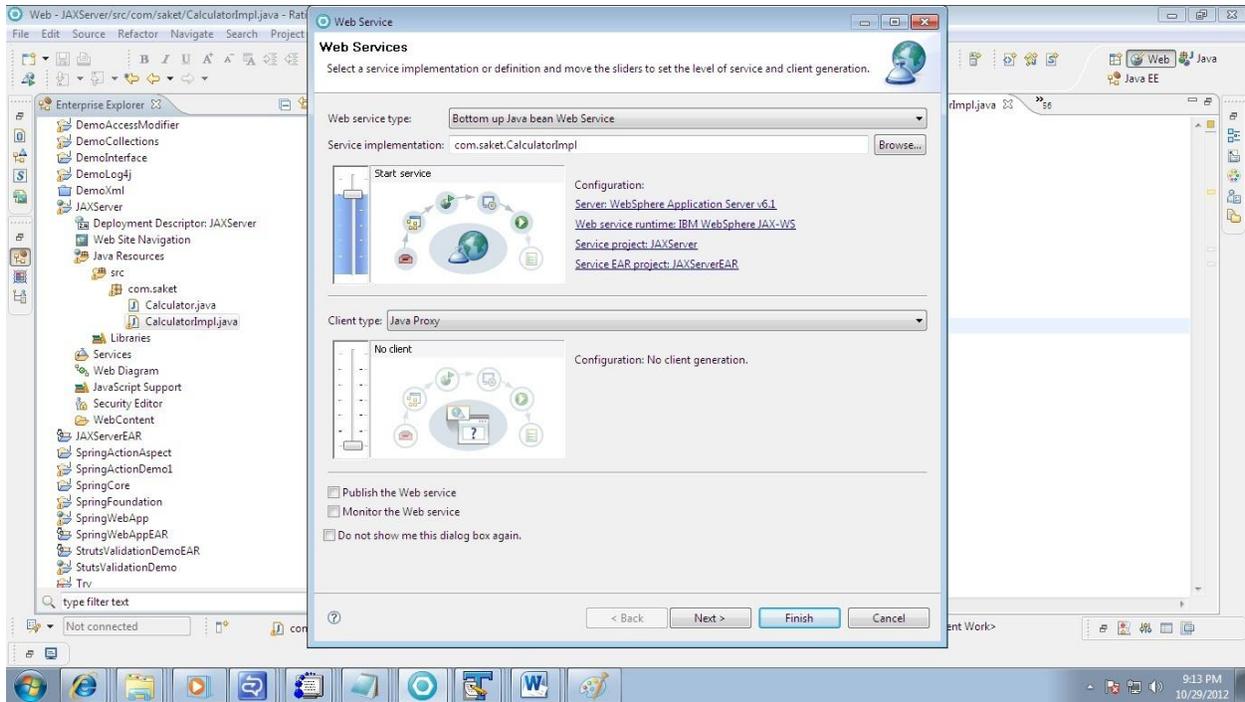
Now everything is explained via snapshot.

Step 6: Go to the Impl class and right click->WebServices->CreateWebService



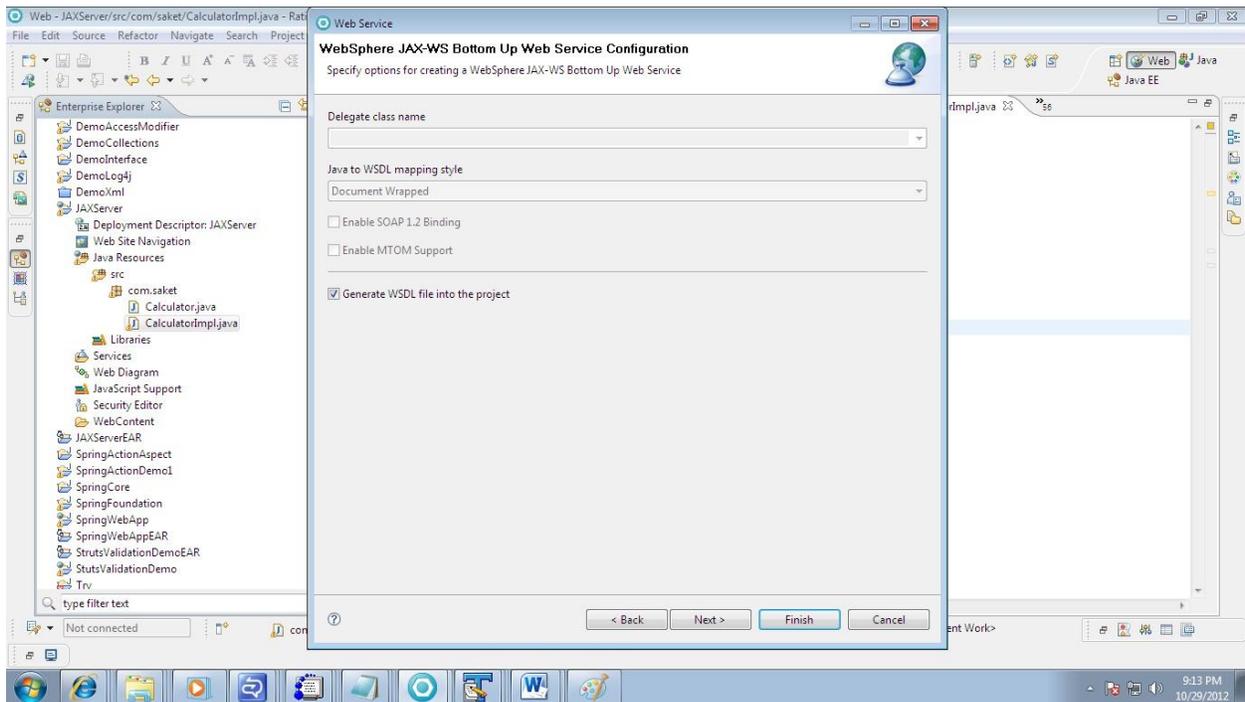
Creating a JAX-WS Webservice Using IBM RAD and WebSphere 6.1

Service Implementation class should show the name of Impl Class. Here CalculatorImpl



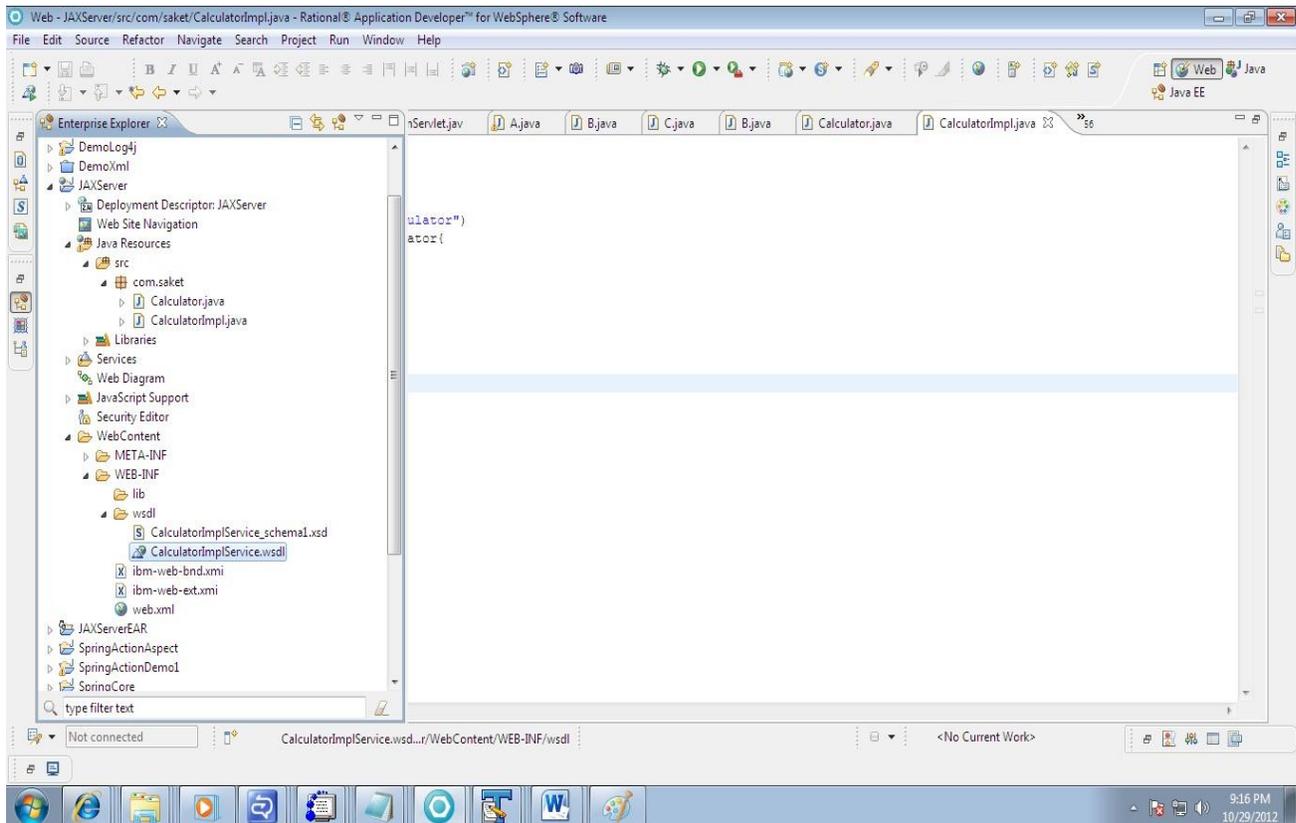
Creating a JAX-WS Webservice Using IBM RAD and WebSphere 6.1

Check the Option Generate WSDL file into Project



Creating a JAX-WS WebService Using IBM RAD and WebSphere 6.1

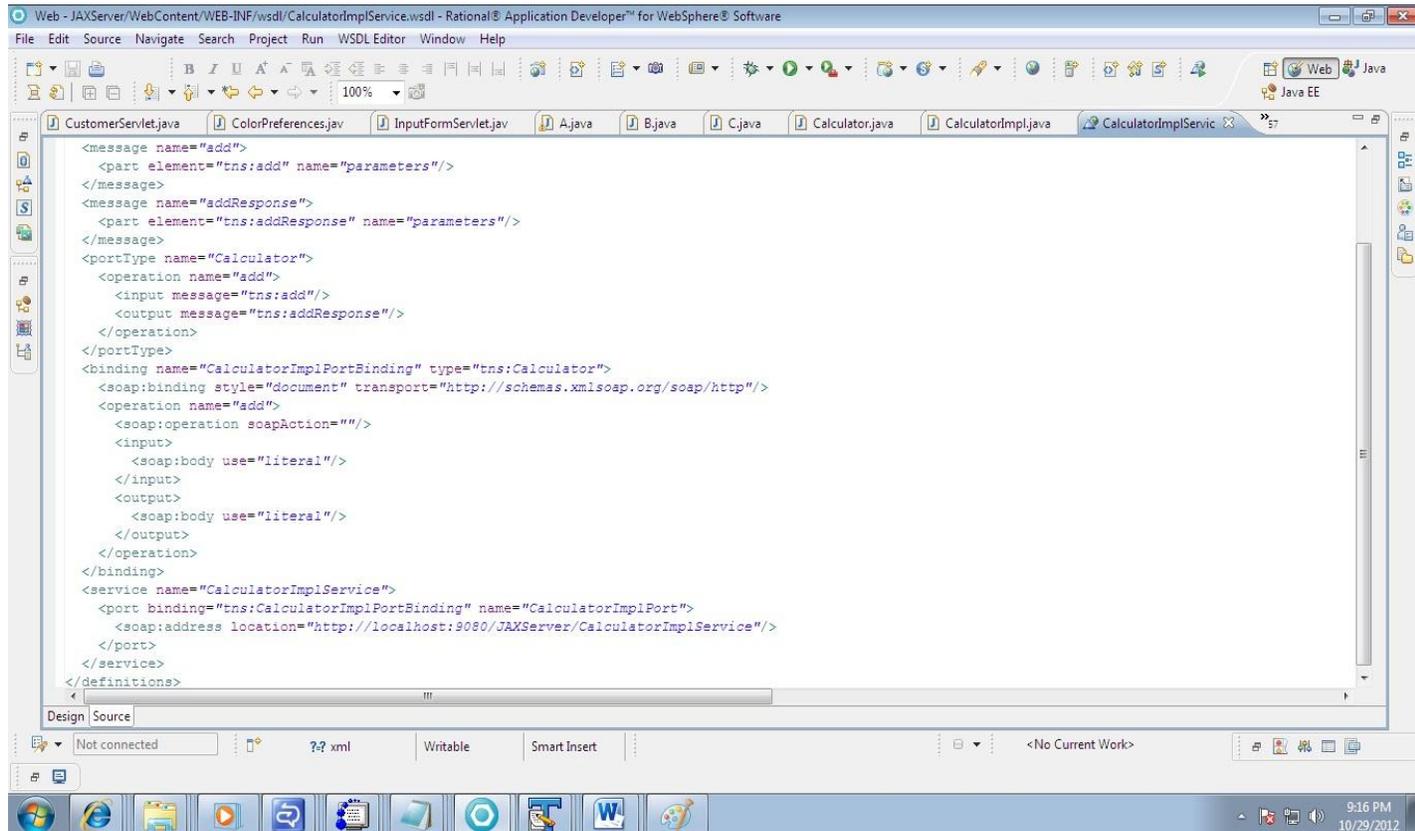
Step 7: When you click finish WsdL will be auto generated and you webService will be deployed. Expand the Project JAXServer and look for wsdl in wsdl folder.



Creating a JAX-WS WebService Using IBM RAD and WebSphere 6.1

Step 8:

Now validate the wsdl and Test using WebService Explorer



Step 9:

Now let us check the wsdl document that would have got generated. Type the below URL in the internet explorer.

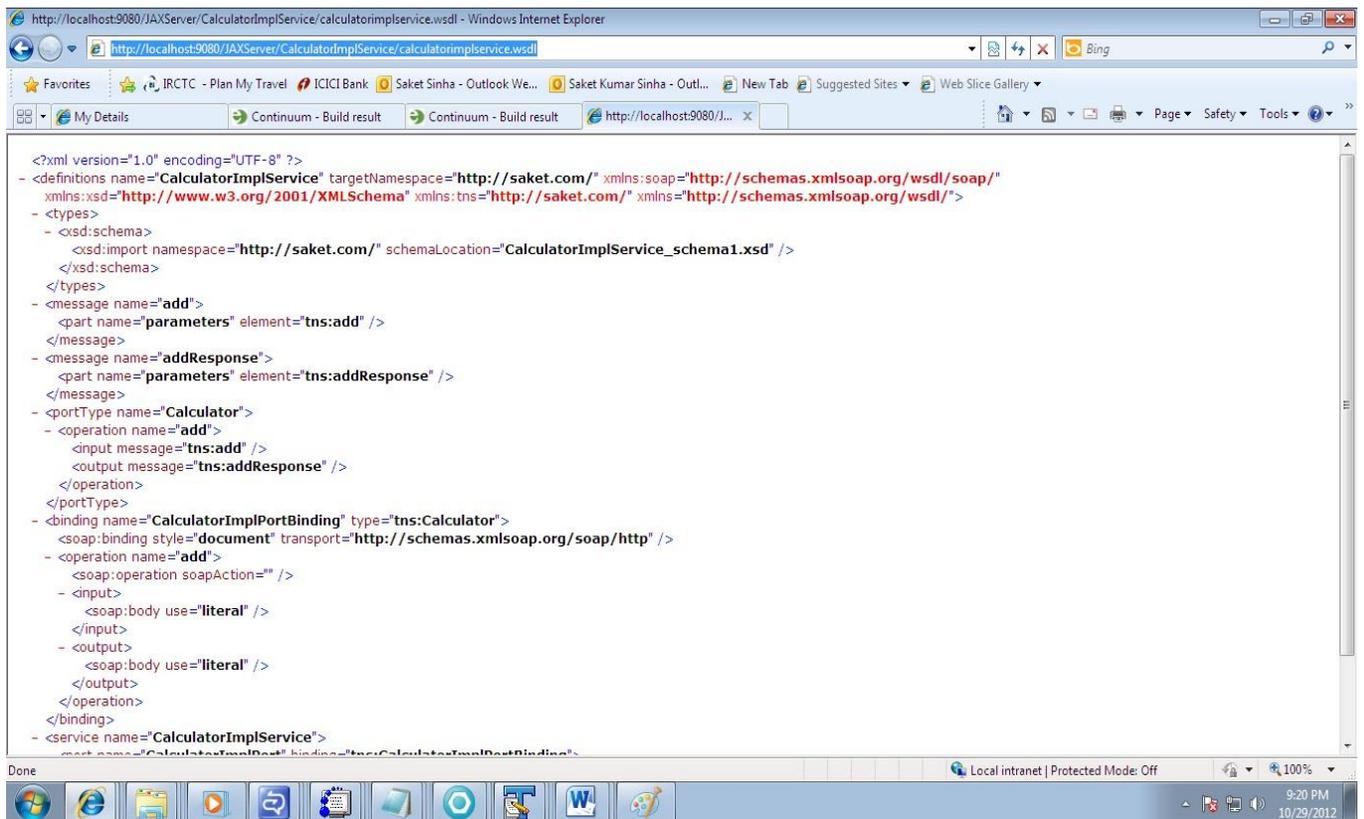
This address is there in soap:address location in WSDL

<http://localhost:9080/JAXServer/CalculatorImplService?wsdl>

Creating a JAX-WS Webservice Using IBM RAD and WebSphere 6.1

Step 10:

You would get the wsdl file that would have been generated for our Calculator endpoint.

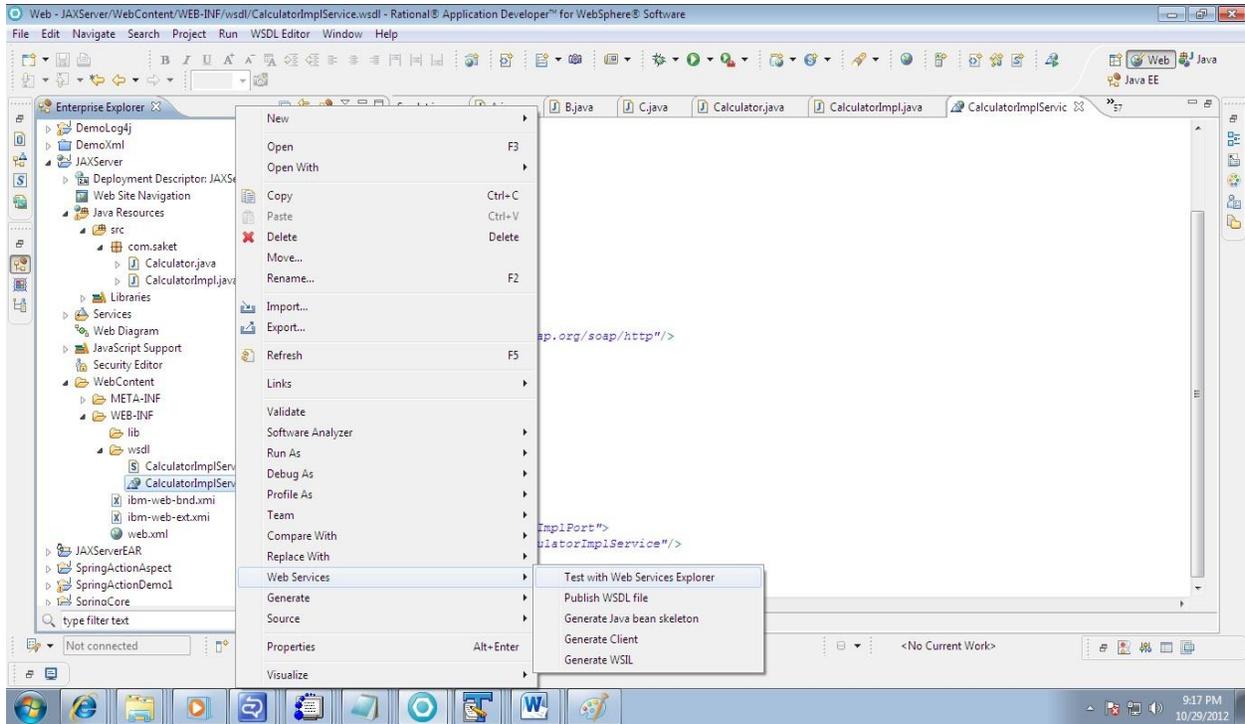


```
<?xml version="1.0" encoding="UTF-8" ?>
- <definitions name="CalculatorImplService" targetNamespace="http://saket.com/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://saket.com/" xmlns="http://schemas.xmlsoap.org/wsdl/">
- <types>
- <xsd:schema>
  <xsd:import namespace="http://saket.com/" schemaLocation="CalculatorImplService_schema1.xsd" />
  </xsd:schema>
</types>
- <message name="add">
  <part name="parameters" element="tns:add" />
</message>
- <message name="addResponse">
  <part name="parameters" element="tns:addResponse" />
</message>
- <portType name="Calculator">
  <operation name="add">
    <input message="tns:add" />
    <output message="tns:addResponse" />
  </operation>
</portType>
- <binding name="CalculatorImplPortBinding" type="tns:Calculator">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
  <operation name="add">
    <soap:operation soapAction="" />
    <input>
      <soap:body use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>
</binding>
- <service name="CalculatorImplService">
  <port name="CalculatorImplPort" binding="tns:CalculatorImplPortBinding" />
</service>
```

Creating a JAX-WS Webservice Using IBM RAD and WebSphere 6.1

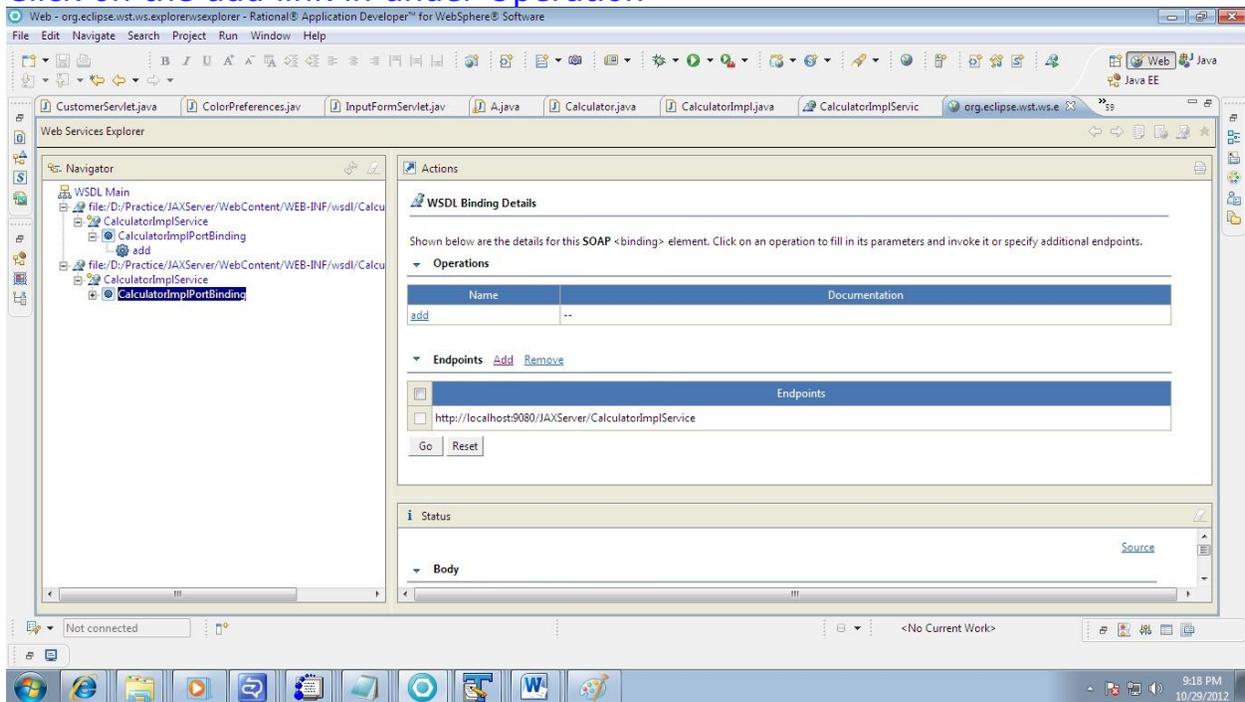
Step 11:

So far we have created a service and published it. Next thing is that we must test with WebService Explorer. Right click on wsdl->WebServices->Test with WebServices Explorer



Creating a JAX-WS Webservice Using IBM RAD and WebSphere 6.1

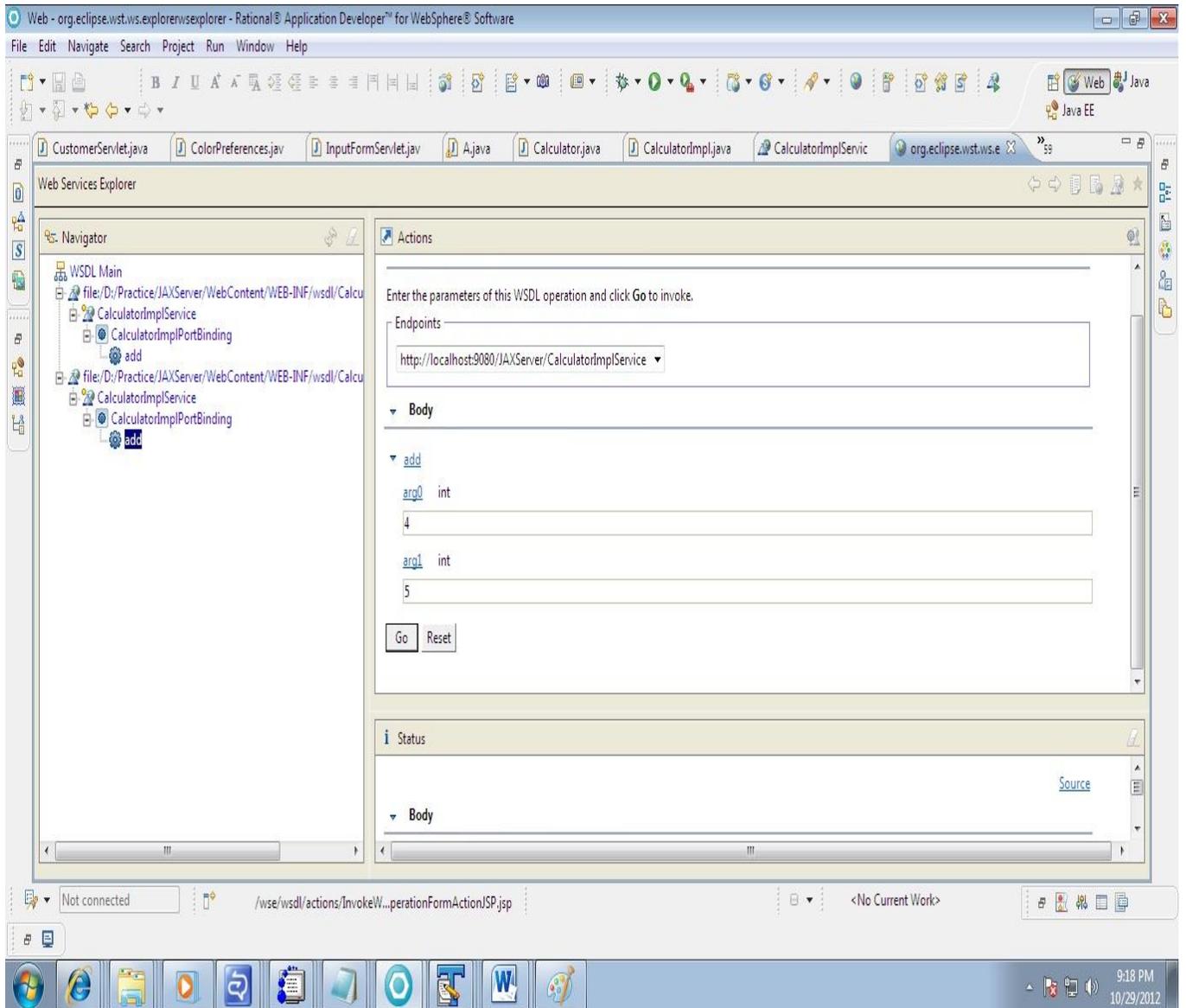
Click on the add link in under Operation



Enter the value of arg0 and arg1

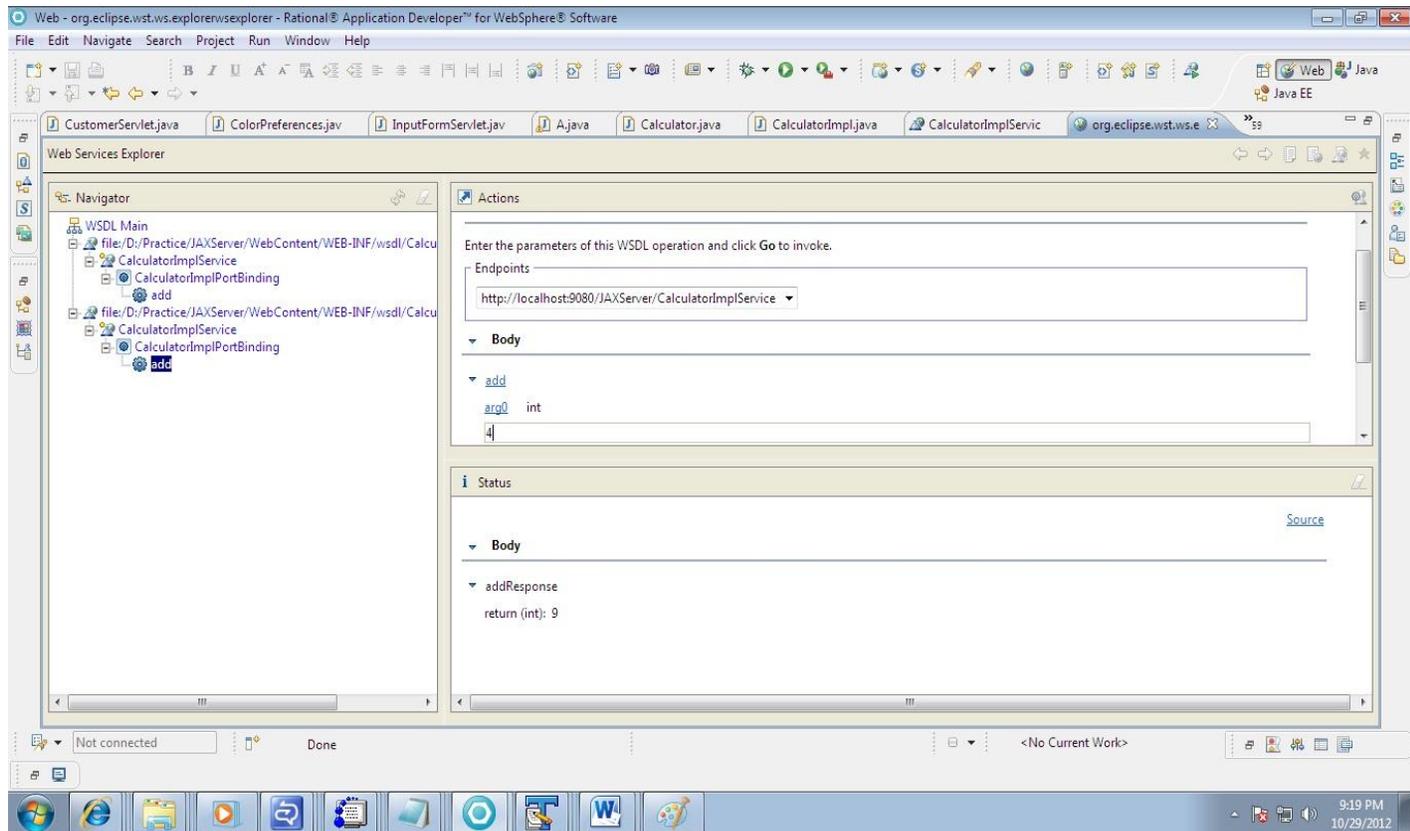
For Ex : 4 and 5 I have entered

Creating a JAX-WS Webservice Using IBM RAD and WebSphere 6.1



Creating a JAX-WS Webservice Using IBM RAD and WebSphere 6.1

You should see the result as 9



Now you Web Service is Up and running.

Step 12: Client Creation

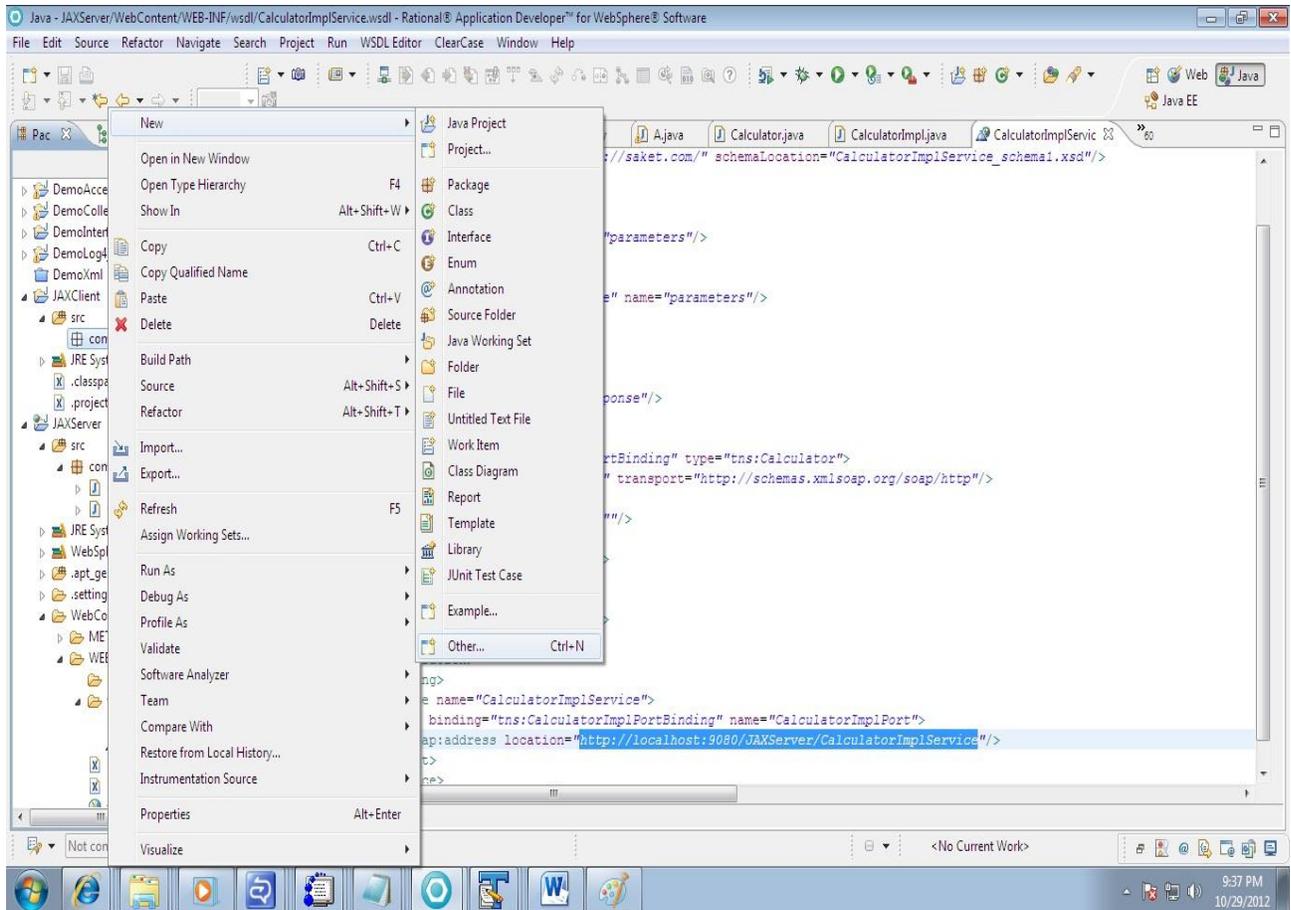
Create a new Dynamic Web project titled "**JAXClient**".

Step 13:

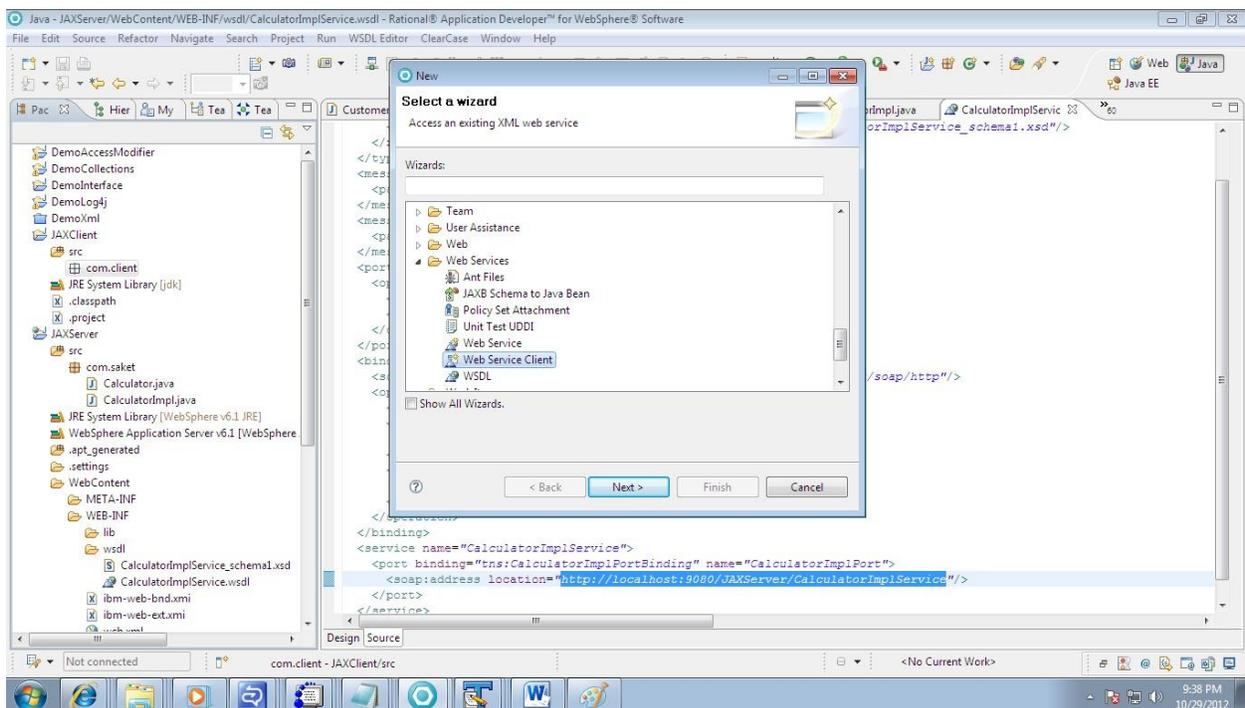
Now we need to create the various stubs required for accessing the service. The stubs can be auto generated. Create a package com.client and follow the diagram.

Right click on package com.client->New->Other

Creating a JAX-WS WebService Using IBM RAD and WebSphere 6.1



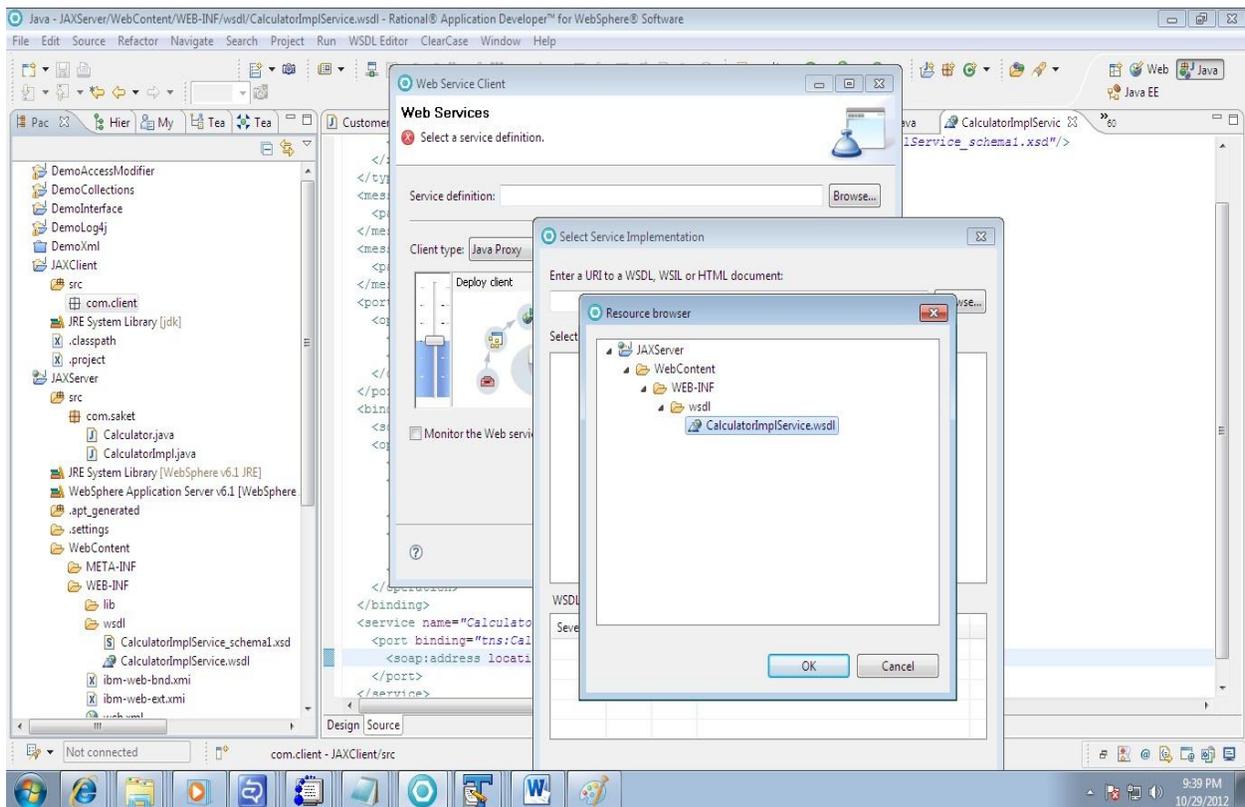
Under Web Services Select Web Service Client. Click on Next



Creating a JAX-WS WebService Using IBM RAD and WebSphere 6.1

Click on Browse Button of Service Definition and select the WSDL file of JAXServer Project.

Press Ok.

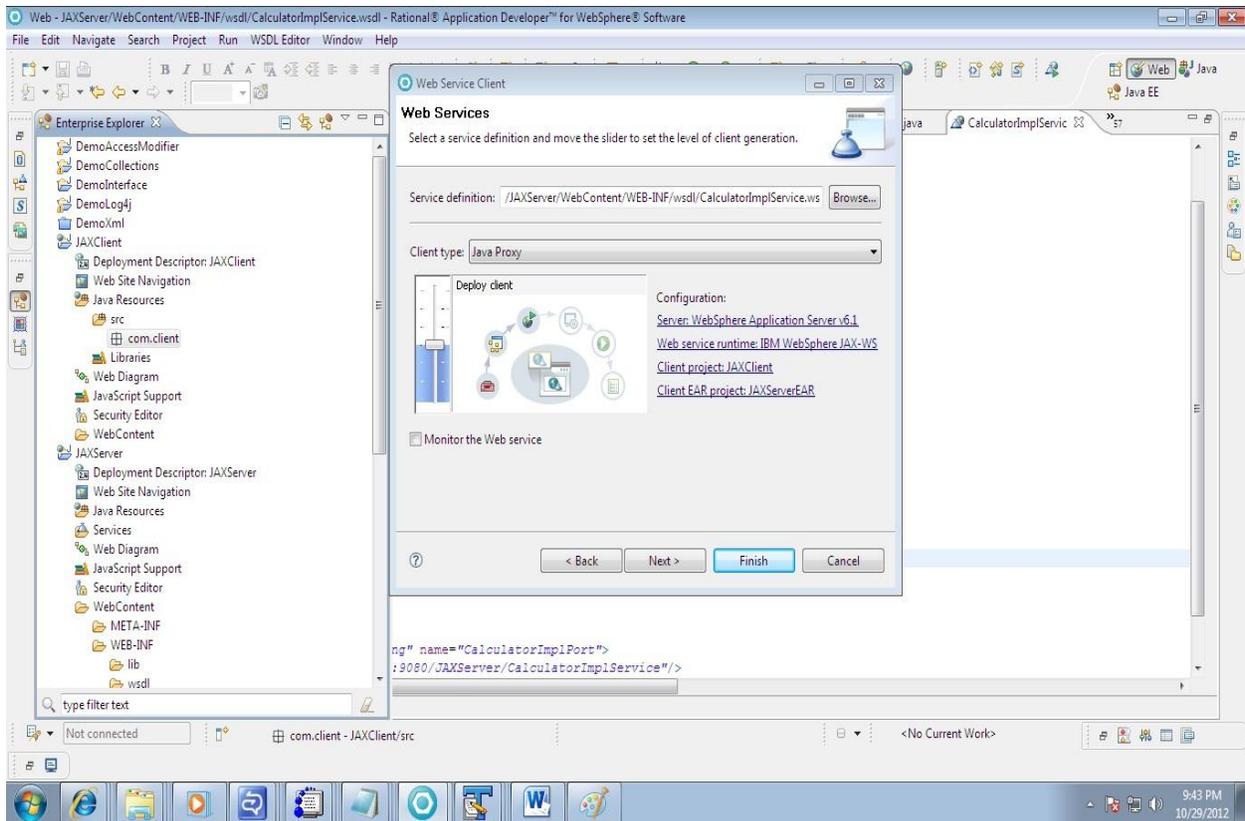


When you have pressed OK. You will see the below window. Verify The path and Configuration mentioned is correct or not.

Server : WebSphere Application Server 6.1

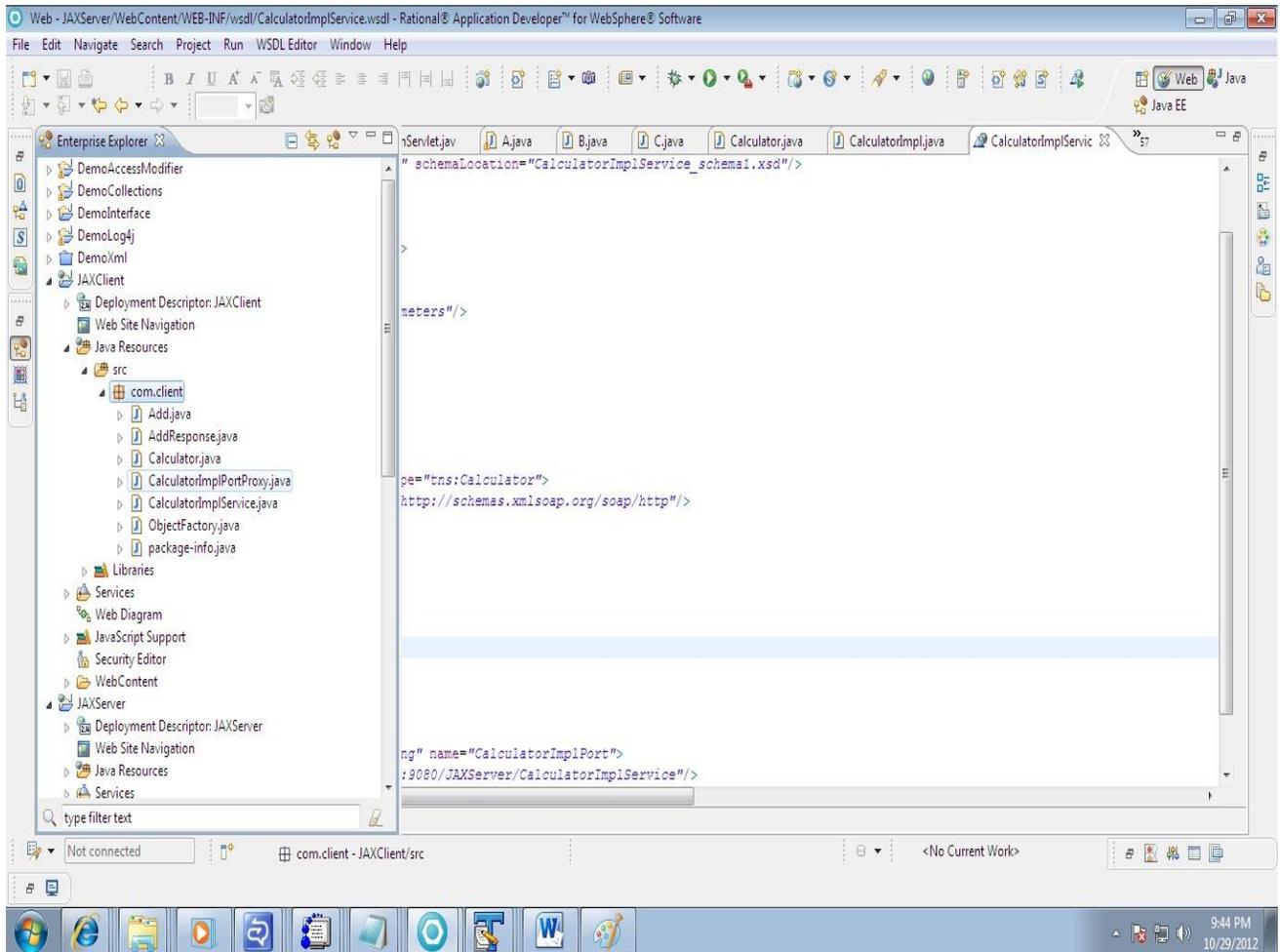
WebService Runtime: IBM webSphere JAX-WS

Creating a JAX-WS Webservice Using IBM RAD and WebSphere 6.1



Creating a JAX-WS Webservice Using IBM RAD and WebSphere 6.1

Click on Finish. All the required files will be auto generated. See the below snapshot



Creating a JAX-WS Webservice Using IBM RAD and WebSphere 6.1

Step 14:

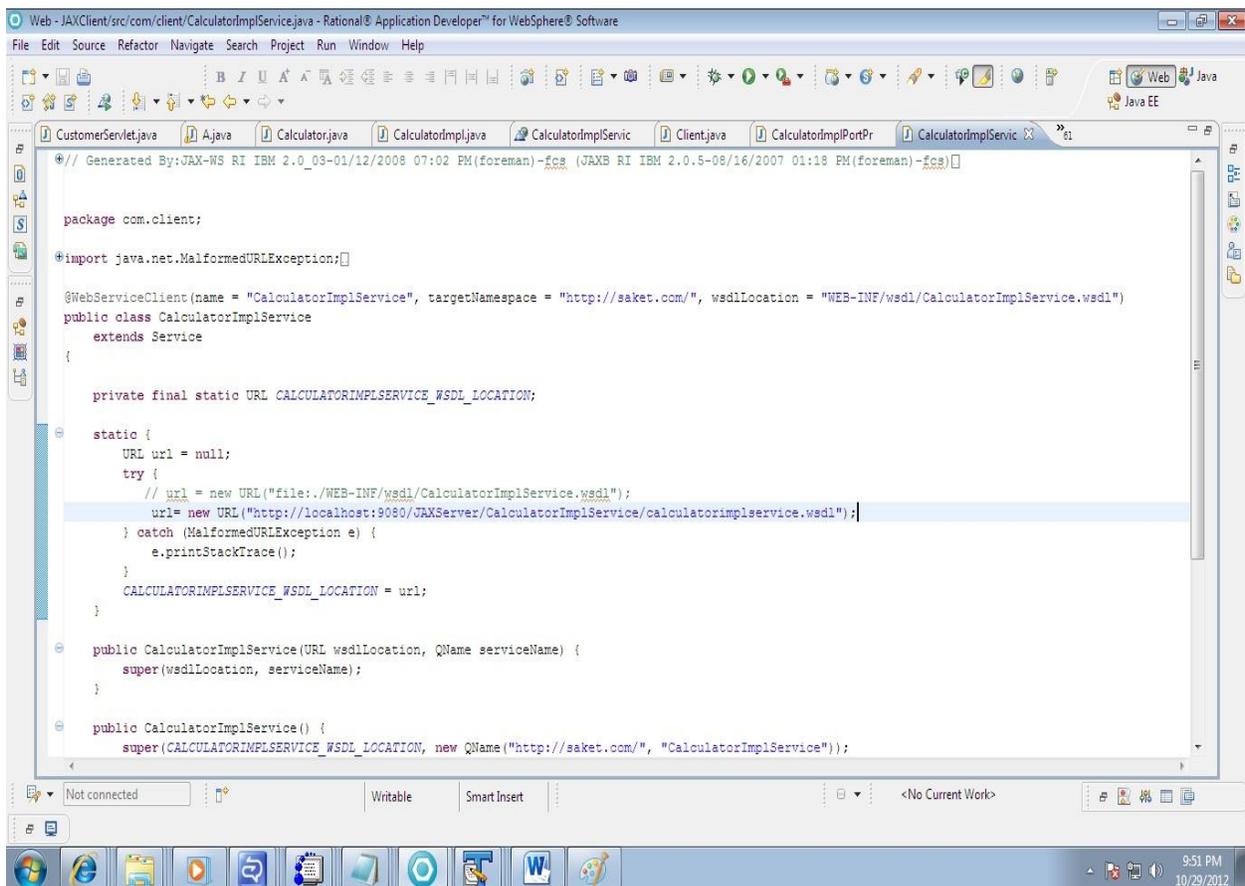
Go to the class CalculatorImplService in com.client package

The Client project uses WSDL . You need to change it the url.

See the below snapshot.

I have changed the URL to

<http://localhost:9080/JAXServer/CalculatorImplService/calculatorimplservice.wsdl>



```
Web - JAXClient/src/com/client/CalculatorImplService.java - Rational® Application Developer™ for WebSphere® Software
File Edit Source Refactor Navigate Search Project Run Window Help

package com.client;

import java.net.MalformedURLException;

@WebServiceClient(name = "CalculatorImplService", targetNamespace = "http://saket.com/", wsdlLocation = "WEB-INF/wsdl/CalculatorImplService.wsdl")
public class CalculatorImplService
    extends Service
{
    private final static URL CALCULATORIMPLSERVICE_WSDL_LOCATION;

    static {
        URL url = null;
        try {
            // url = new URL("file:./WEB-INF/wsdl/CalculatorImplService.wsdl");
            url = new URL("http://localhost:9080/JAXServer/CalculatorImplService/calculatorimplservice.wsdl");
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }
        CALCULATORIMPLSERVICE_WSDL_LOCATION = url;
    }

    public CalculatorImplService(URL wsdlLocation, QName serviceName) {
        super(wsdlLocation, serviceName);
    }

    public CalculatorImplService() {
        super(CALCULATORIMPLSERVICE_WSDL_LOCATION, new QName("http://saket.com/", "CalculatorImplService"));
    }
}
```

Creating a JAX-WS Webservice Using IBM RAD and WebSphere 6.1

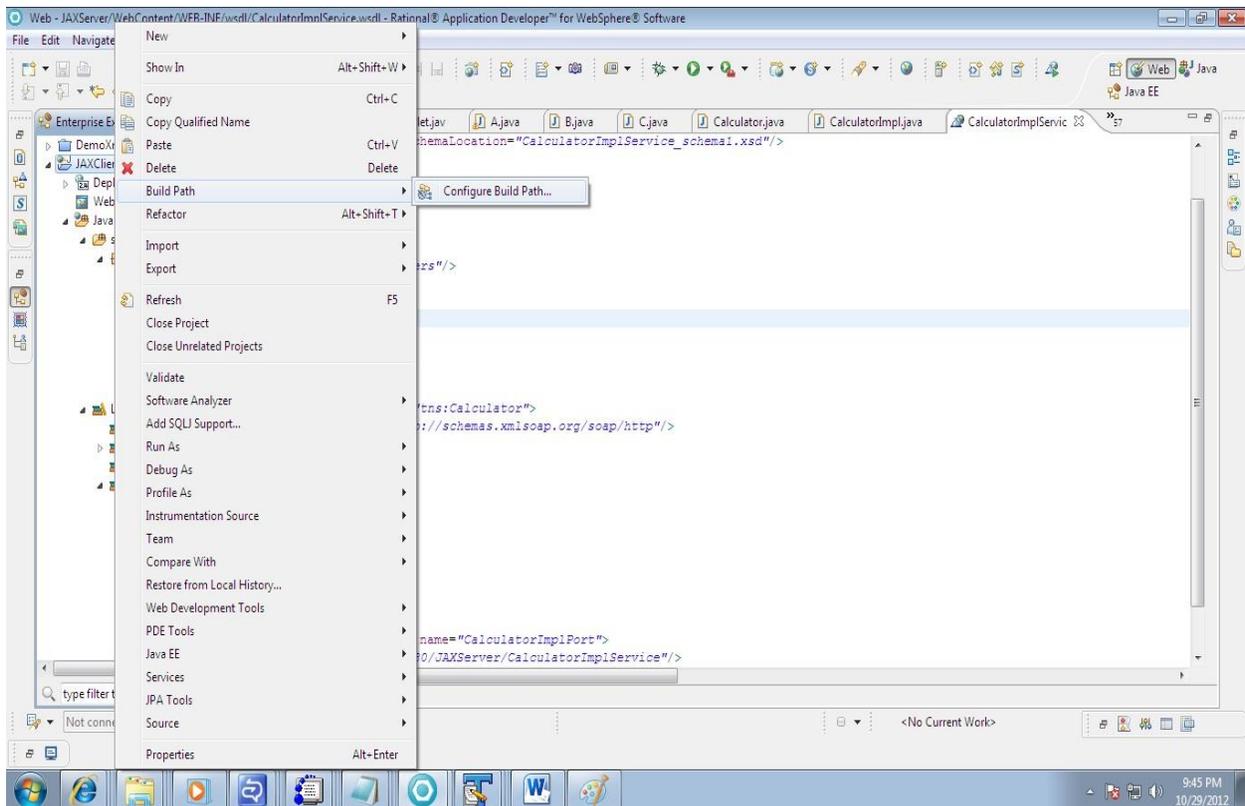
Step 15:

Now Remove the WebSphere Library from the Project.

We Need only the WebSphere thin client jar for Webservice Client.

Follow the below steps

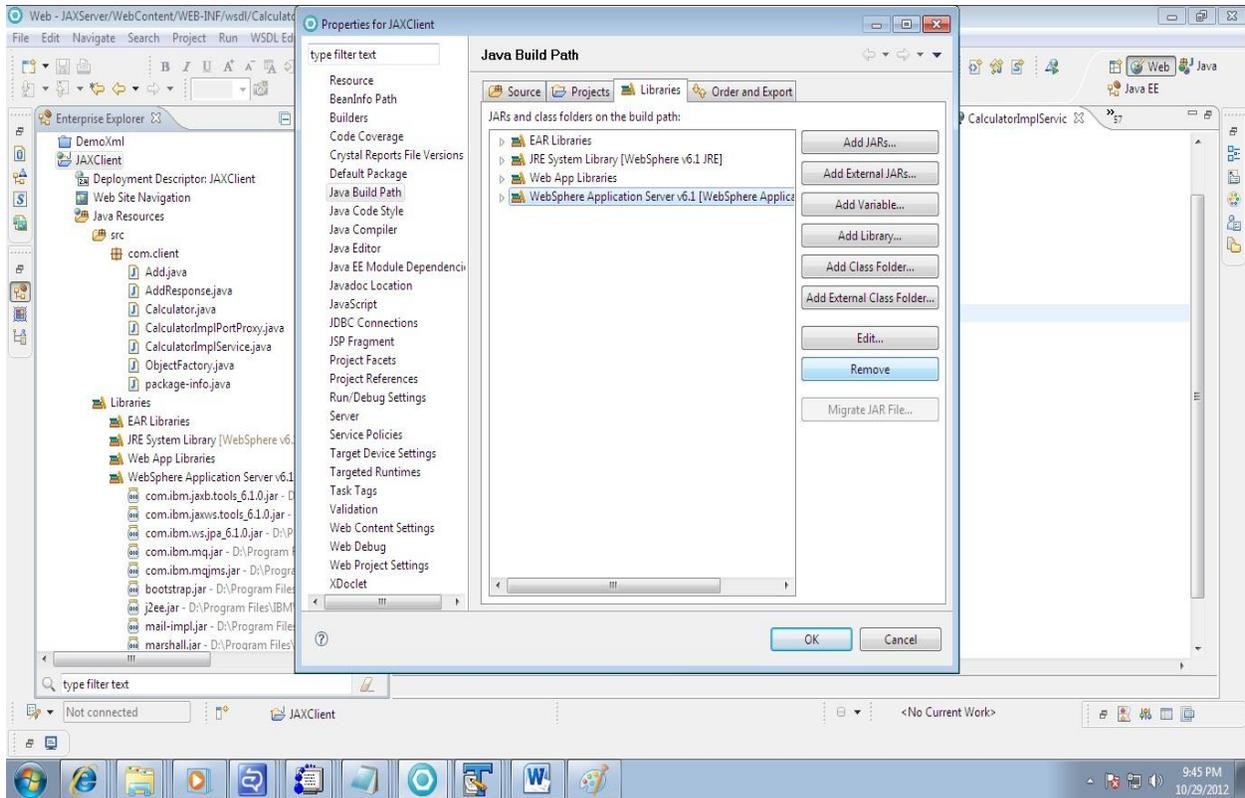
Right Click on JAXClient->BuildPath->Configure Build Path



Under Libraries Tab -> Click on WebSphere Application Server 6.1. Click On Remove.

And Press OK.

Creating a JAX-WS WebService Using IBM RAD and WebSphere 6.1



When You do this all the class in client package will be marked with red sign. So we need to add thin client jar.

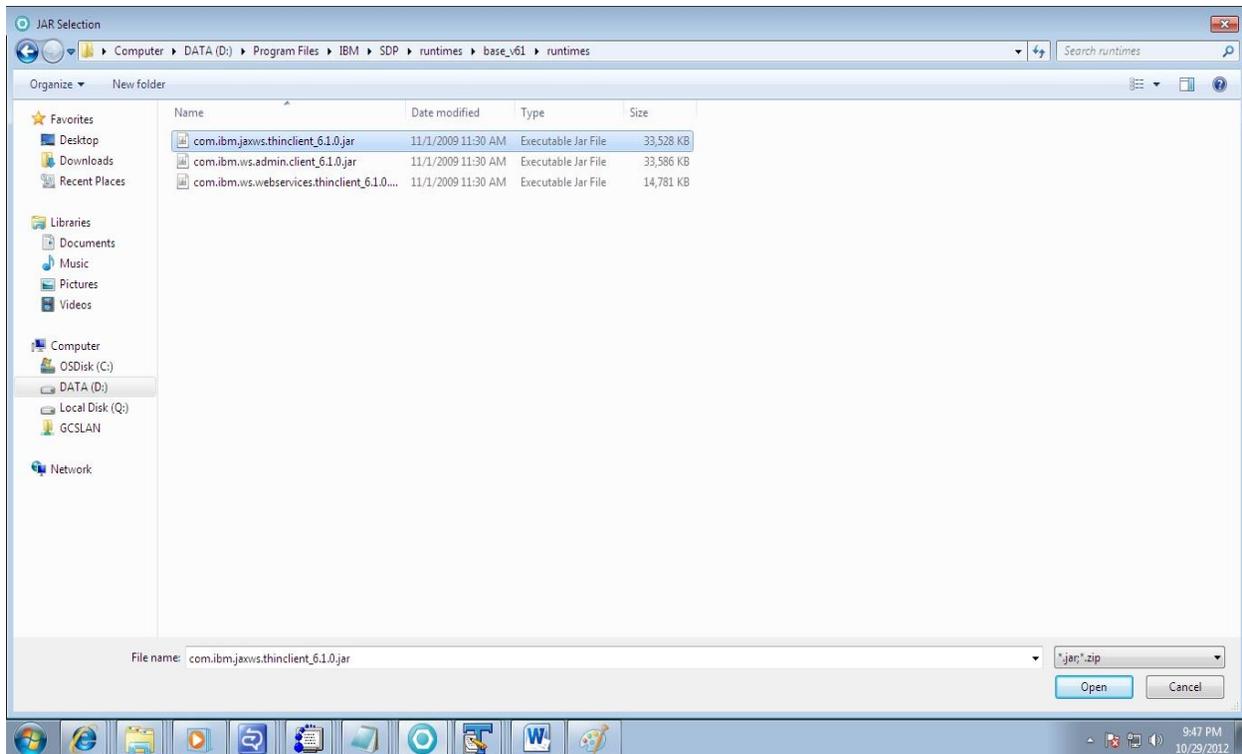
For me the location of thin client jaxws jar is

D:\Program Files\IBM\SDP\runtimes\base_v61\runtimes

So accordingly you need to add this

com.ibm.jaxws.thinclient_6.1.0.jar to the JAXClient by clicking on Add External Jars under Configure Build Path Option.

Creating a JAX-WS WebService Using IBM RAD and WebSphere 6.1



Step 16:

we need to create a client program that invokes the server through the stub.

Step 17:

Create a new Java class "**Client**" in the com.client package.

Step 18:

Write the below code inside the Client class to invoke the stubs.

```
package com.client;

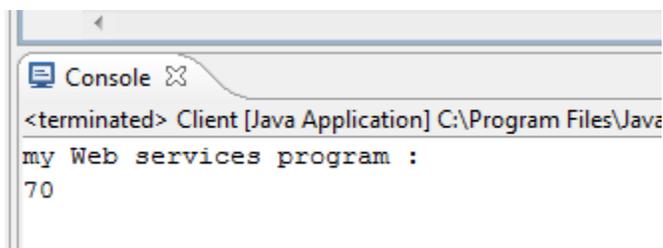
public class Client {
    public static void main(String[] args) {
        CalculatorImplService service = new CalculatorImplService();
        Calculator calculator = service.getCalculatorImplPort();
        int var1 = 20;
    }
}
```

Creating a JAX-WS Webservice Using IBM RAD and WebSphere 6.1

```
int var2 = 50;
int value = calculator.add(var1, var2);
System.out.println("my Web services program : ");
System.out.println(value);
}
}
```

Step 19:

Execute the Client.java as java Application. You would get the following output in the console



Step 20:

We have created a web service and deployed it successfully.

In the next article I am going to explain the theoretical details as well as how to create a secure web service and how any client can invoke it using certificates i.e public key.

Author: [Prashant Prakash](#)

FaceBook Page: <https://www.facebook.com/aliencoders>

Twitter: <http://twitter.com/aliencoders>

Online Content: <http://www.aliencoders.com/content/creating-jax-ws-web-service-using-ibm-rad-and-websphere-61>

You are free to share the pdf, modify/edit/update the contents, provided that author's approval is given. Please give credit to the site and author before sharing it anywhere else.