**Creation of Web service and Web Client**

Web Service is similar to a function. How a function can be used in any application, similarly web service can be utilized to make web application.

**Software Required**

Installation of Net Beans (Netbeans 8.0.2)in Ubuntu 16

This version will be available as script file.

You need to give execute permission

chmod +x netbe.....sh

To run script

./nerbe....sh

While installing, choose GlassFish Server.

**Creation of Hello Web Service**

1. To create web application by doing the following:

File -> New Project -> Java Web -> Web Application

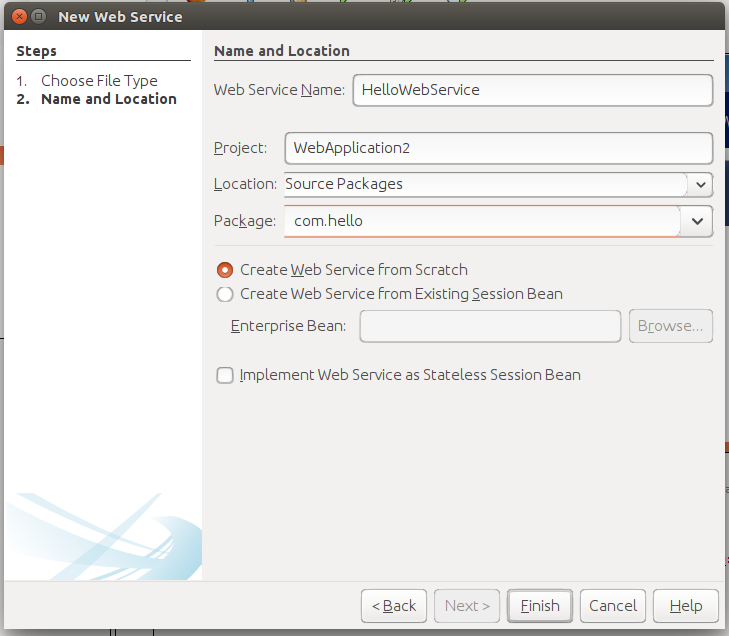
Server -> Choose GlassFish Server

Press Finish button

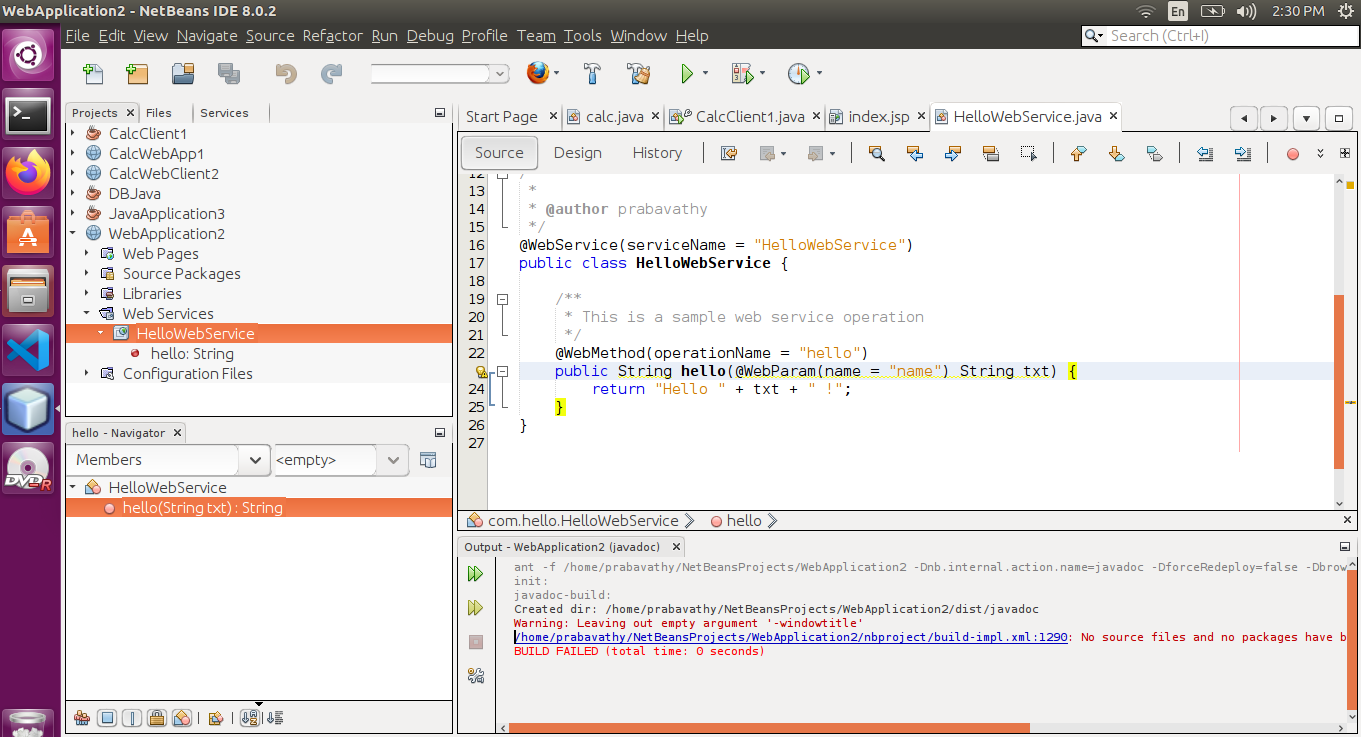
1. Create new web service in that application

Let us say our application name is **Web Application2**. Keep cursor over the **Web Application2** and create new web service.

Give name to the web service **HelloWebService** and package name can be com.hello.

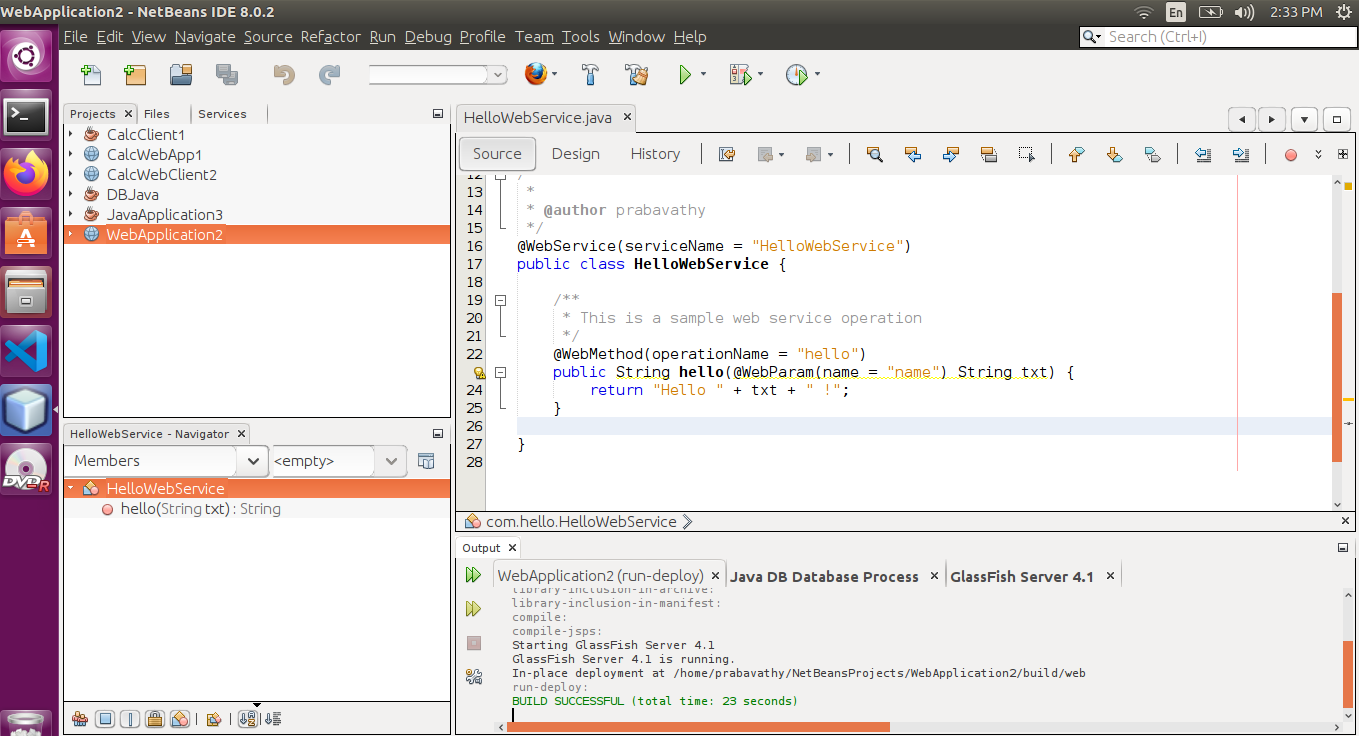


**Code in HelloWebService.java**



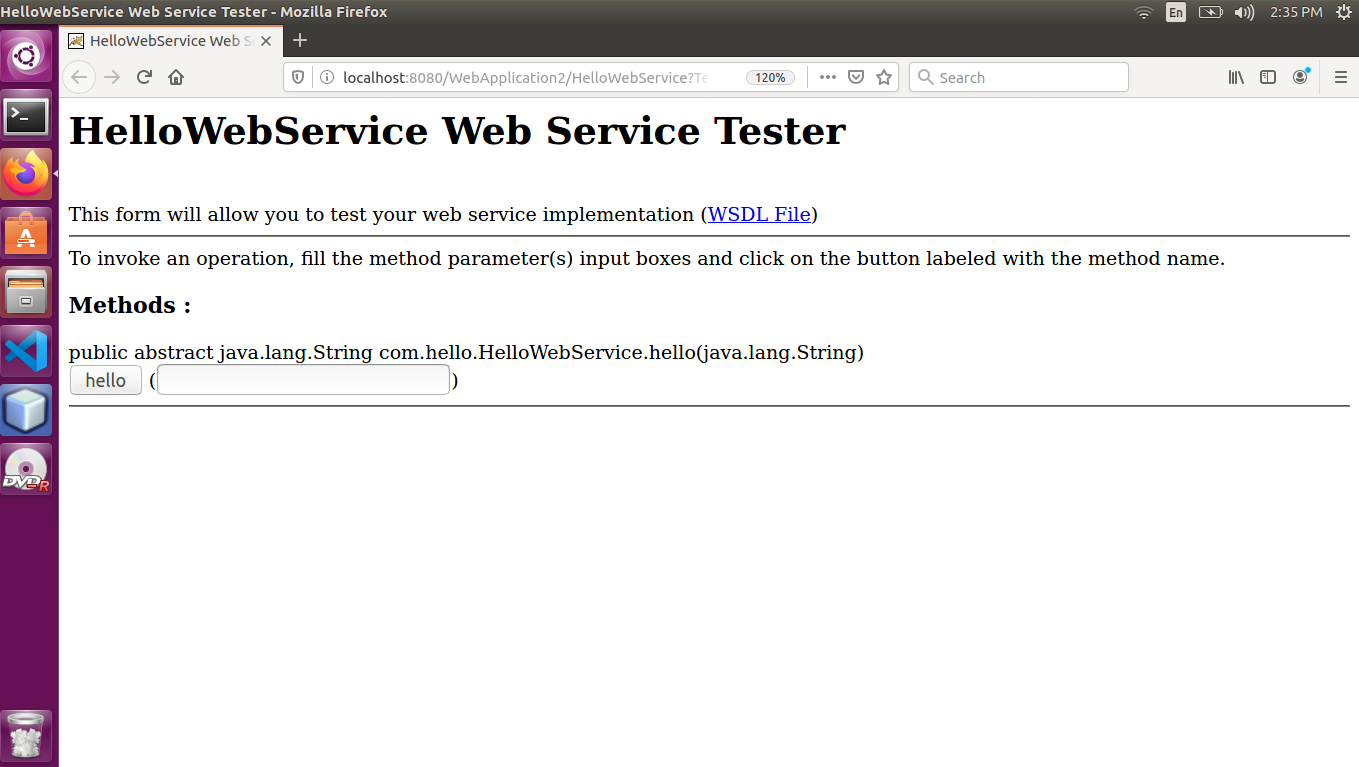
1. Web Service Deployment

Right Click on Web Application2. Clean and Build, Deploy.

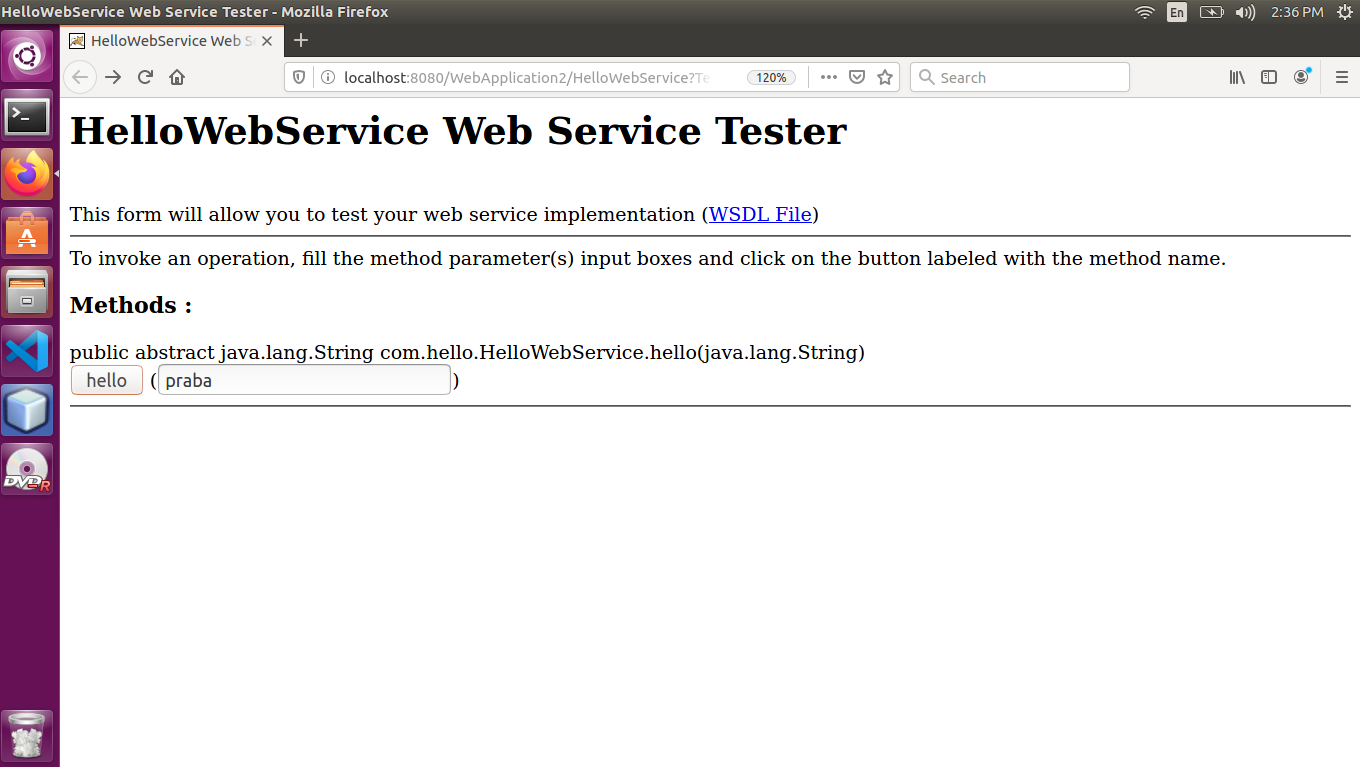


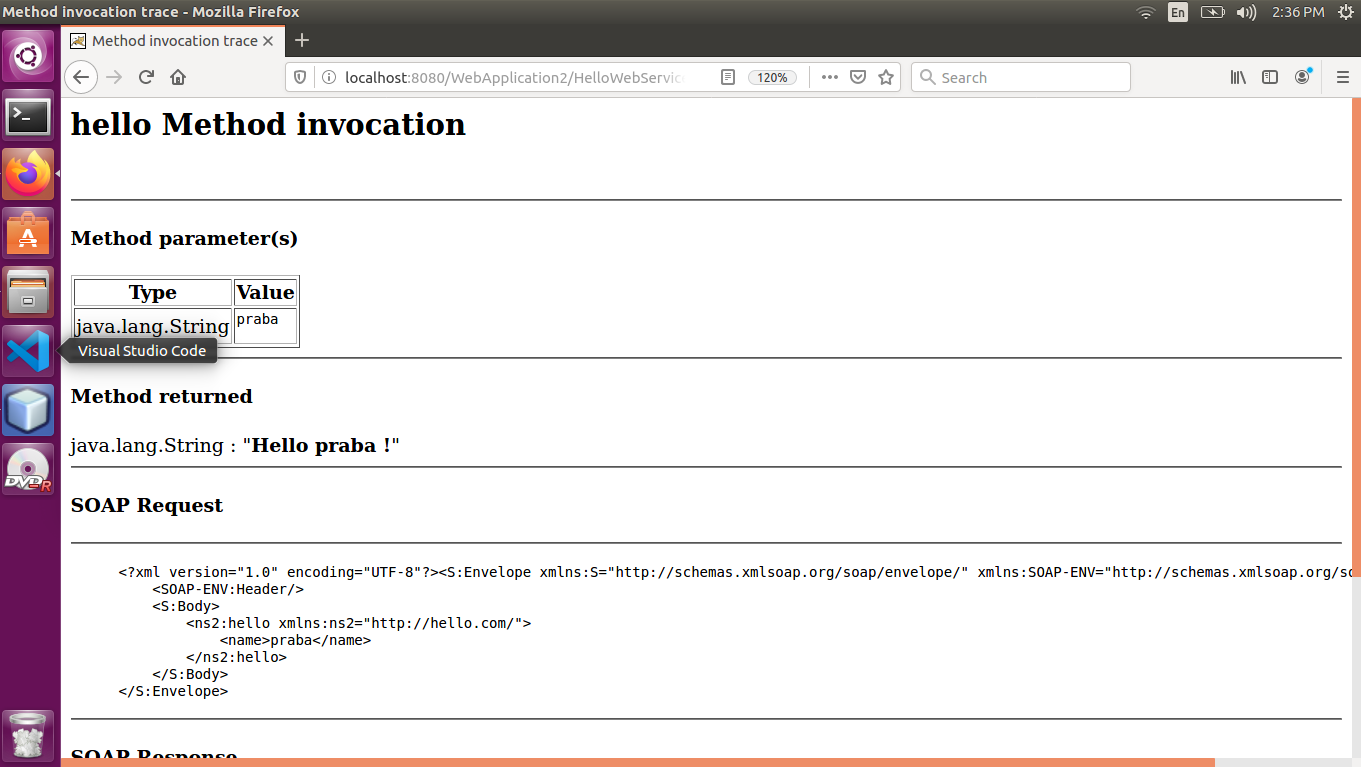
Goto Web Services->HelloWebService in Web Application2 folder and right click. Press Test Web Service.

It will open a link in the browser as follows.

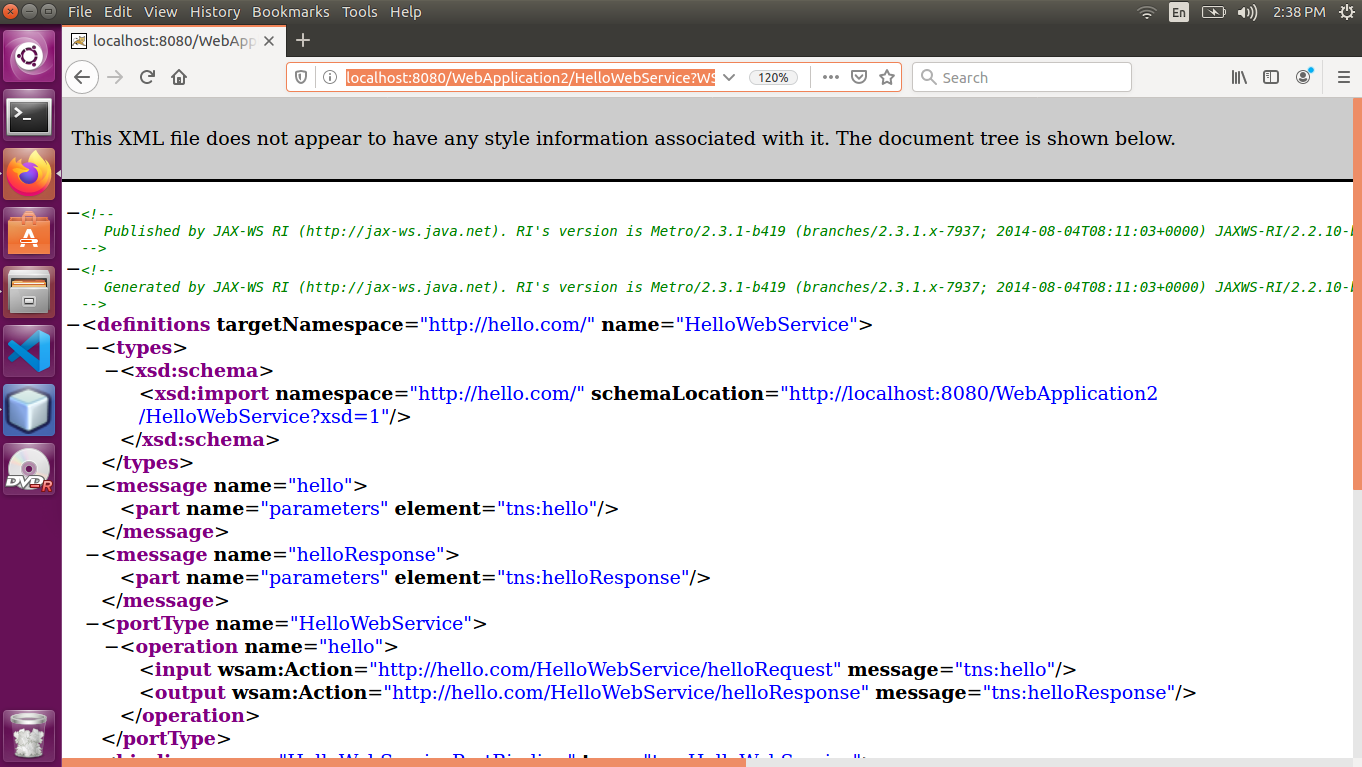


In the link, give your name as input and press hello button. It will generate output as Hello followed your name.





In the link, you will find WSDL url link. You need to click that. Copy the url of wsdl file.



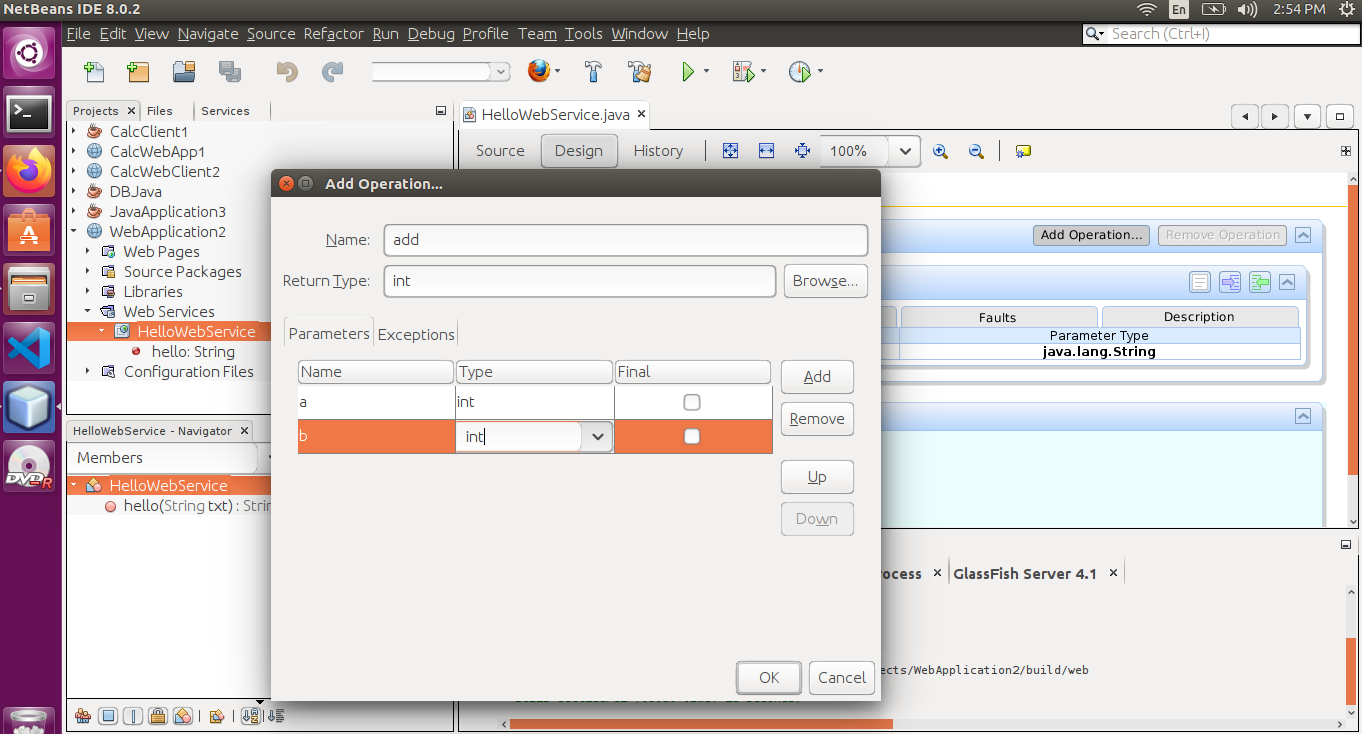
1. To add our own method

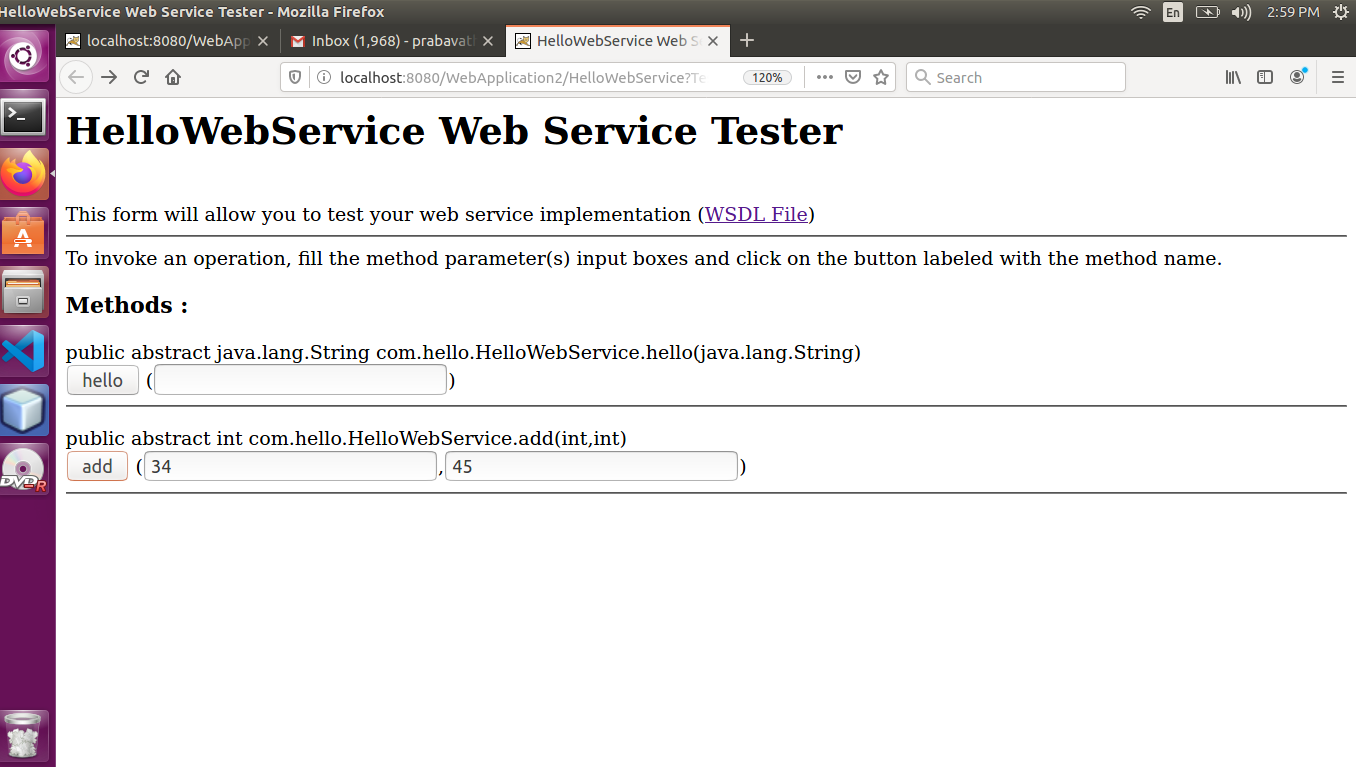
An another method namely ***add*** can be included (Use design tab of HelloWebService.java code and press add operation button) as follows. Similar to hello service, this operation can also be tested.

Need to change add method by the return statement

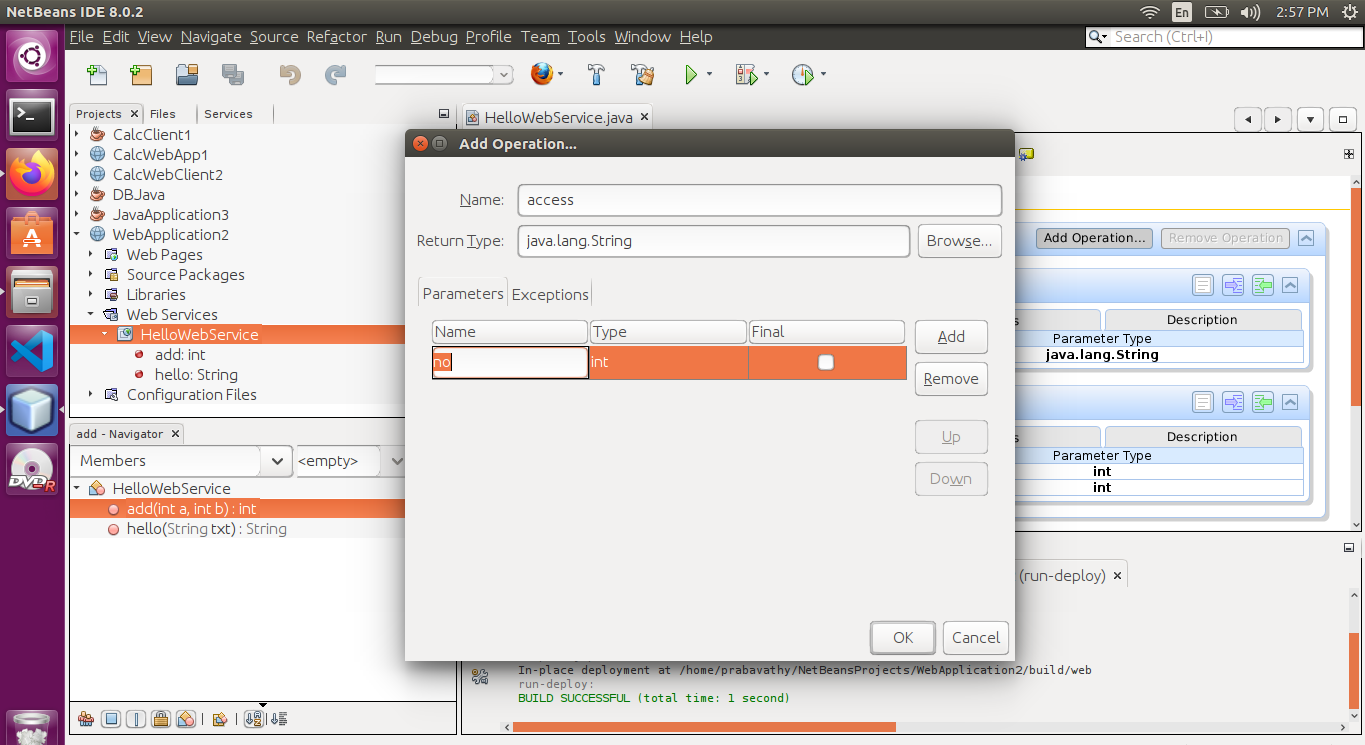
@WebMethod(operationName = "add")  
    public int add(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {  
        //TODO write your implementation code here:  
**return a+b;**

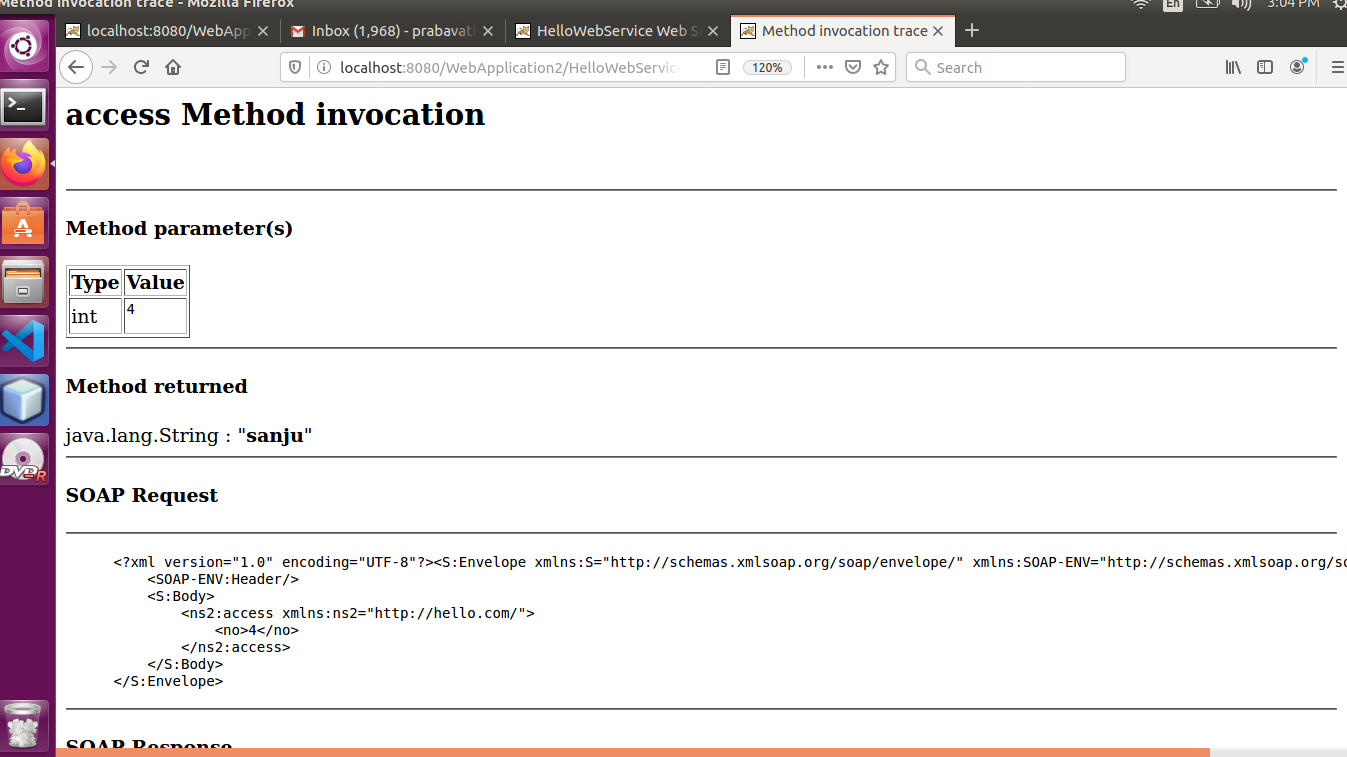
    }





1. To add access method, to contact the database. This access method will get no as input and returns string. Mysql table student is having no and name fields. Given number of a student, this method will give name of the student.



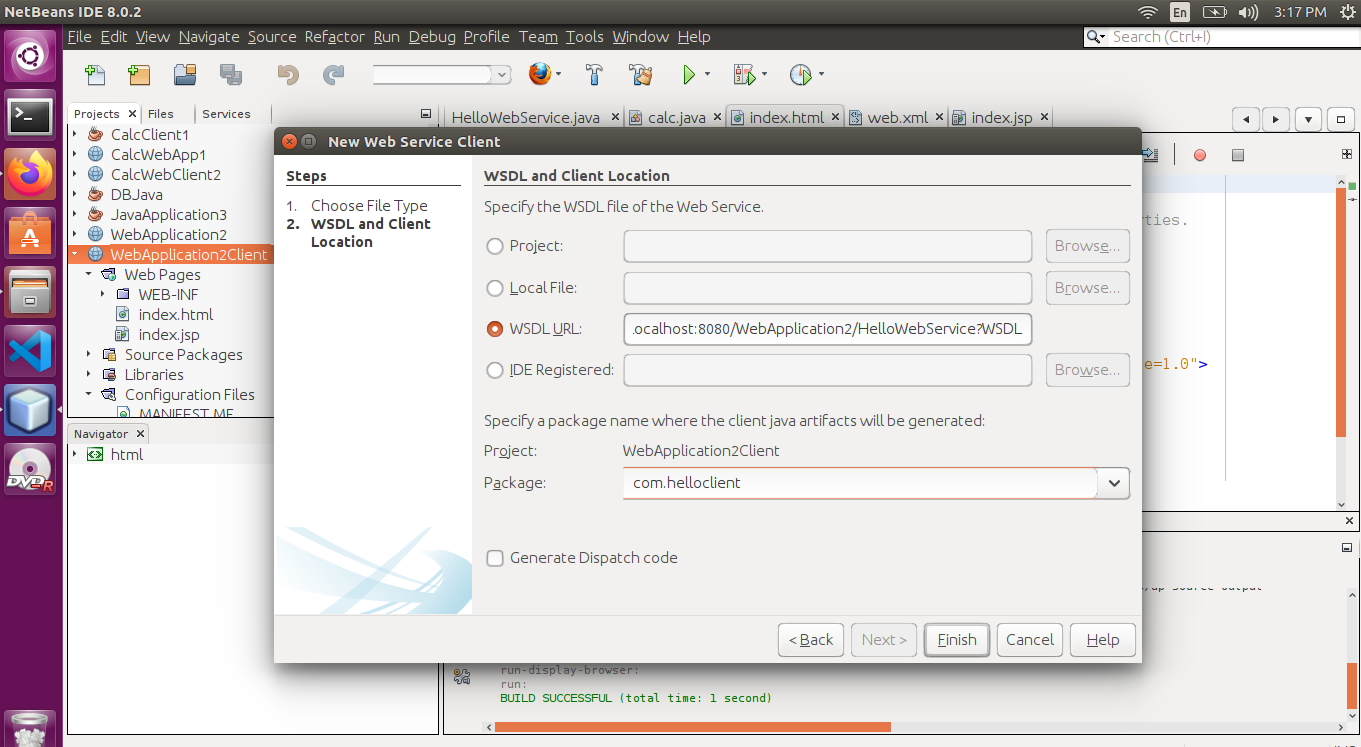


**Creation of WebApplication2Client**

1. Follow the same steps as in the creation of WebApplication2
2. Goto Web Pages -> WEB-INF Right Click and create **web.xml and keep the following content in web.xml.**

<?xml version="1.0" encoding="UTF-8"?>  
<!--  
To change this license header, choose License Headers in Project Properties.  
To change this template file, choose Tools | Templates  
and open the template in the editor.  
-->  
  
  
<web-app xmlns="<http://xmlns.jcp.org/xml/ns/javaee>"  
xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>"  
xsi:schemaLocation="<http://xmlns.jcp.org/xml/ns/javaee> <http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd>"  
version="3.1">  
    <session-config>  
        <session-timeout>  
            30  
        </session-timeout>  
    </session-config>  
    <welcome-file-list>  
<welcome-file>index.jsp</welcome-file>  
</welcome-file-list>  
      
</web-app>

1. Goto Web Pages Right Click and create index.jsp
2. Right click on Webapplication2Client and create new web service client (Use the wdsl url to create)



Goto WebApplication2Client🡪WebServiceReferences->HelloWebService🡪 HelloWebServicePort. You can find the ports add, hello and access. Drag each one to index.jsp. It will bring scriptlets. Change the input and run the code.

**Index.jsp**

**After**  <h1>Hello World!</h1>

**// Dragged hello webservice**

 <%-- start web service invocation --%><hr/>  
    <%  
    try {  
com.helloclient.HelloWebService\_Service service = new com.helloclient.HelloWebService\_Service();  
com.helloclient.HelloWebService port = service.getHelloWebServicePort();  
// TODO initialize WS operation arguments here  
**java.lang.String name = "praba";**  
// TODO process result here  
java.lang.String result = port.hello(name);  
out.println("Result = "+result);  
    } catch (Exception ex) {  
// TODO handle custom exceptions here  
    }  
    %>  
    <%-- end web service invocation --%><hr/>

**// Dragged add webservice**

     <<%-- start web service invocation --%><hr/>  
    <%  
    try {  
com.helloclient.HelloWebService\_Service service = new com.helloclient.HelloWebService\_Service();  
com.helloclient.HelloWebService port = service.getHelloWebServicePort();  
// TODO initialize WS operation arguments here  
**int a = 10;  
int b = 20;**  
// TODO process result here  
int result = port.add(a, b);  
out.println("Result = "+result);  
    } catch (Exception ex) {  
// TODO handle custom exceptions here  
    }  
    %>  
    <%-- end web service invocation --%><hr/>

**// Dragged access webservice**

<%-- start web service invocation --%><hr/>  
    <%  
    try {  
com.helloclient.HelloWebService\_Service service = new com.helloclient.HelloWebService\_Service();  
com.helloclient.HelloWebService port = service.getHelloWebServicePort();  
// TODO initialize WS operation arguments here  
**int no = 4;**  
// TODO process result here  
java.lang.String result = port.access(no);  
out.println("Result = "+result);  
    } catch (Exception ex) {  
// TODO handle custom exceptions here  
    }  
    %>  
    <%-- end web service invocation --%><hr/>

**Output**

