Name >> Yashchandra Upadhyay

Roll number >> 5

Practical number >> 3

Subject >> cloud computing

Practical name >> Implementing JaxWS webservices in NetBeans

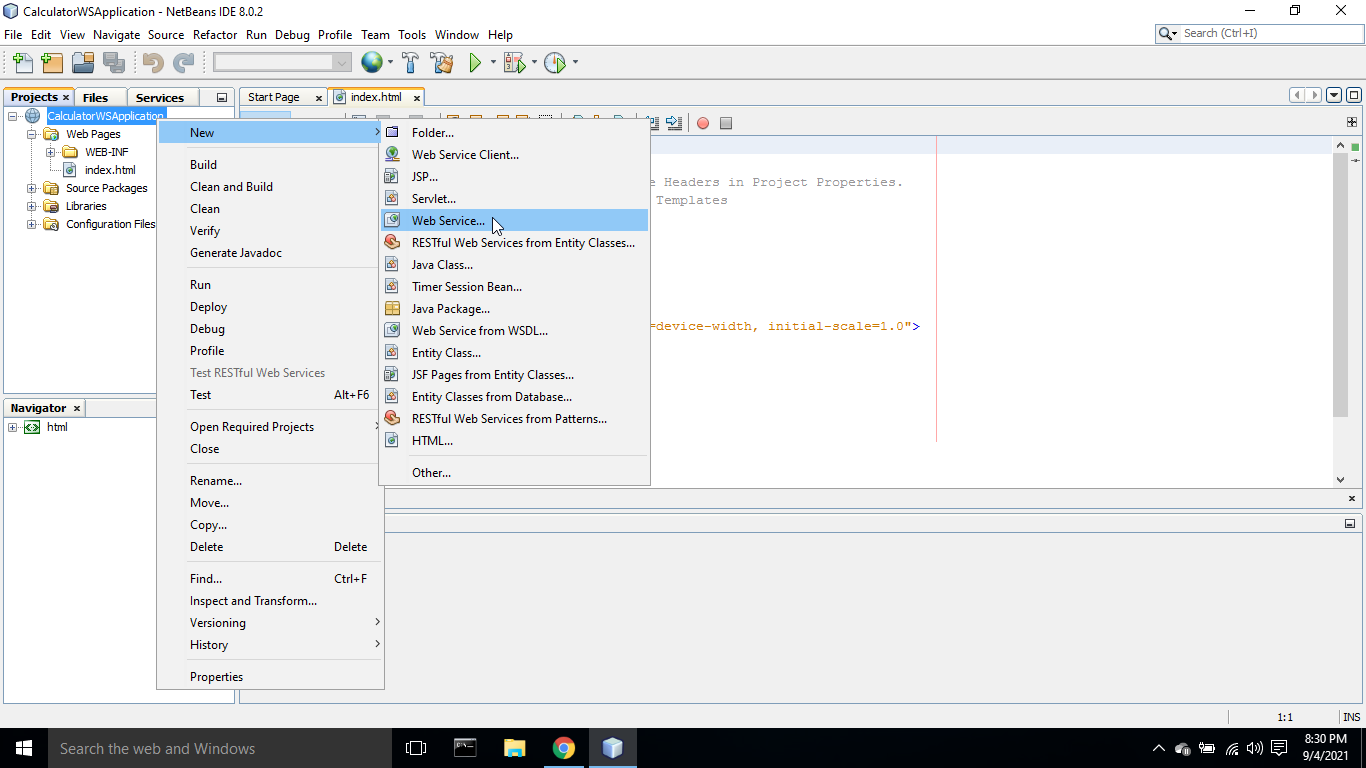
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Creating a Web Services

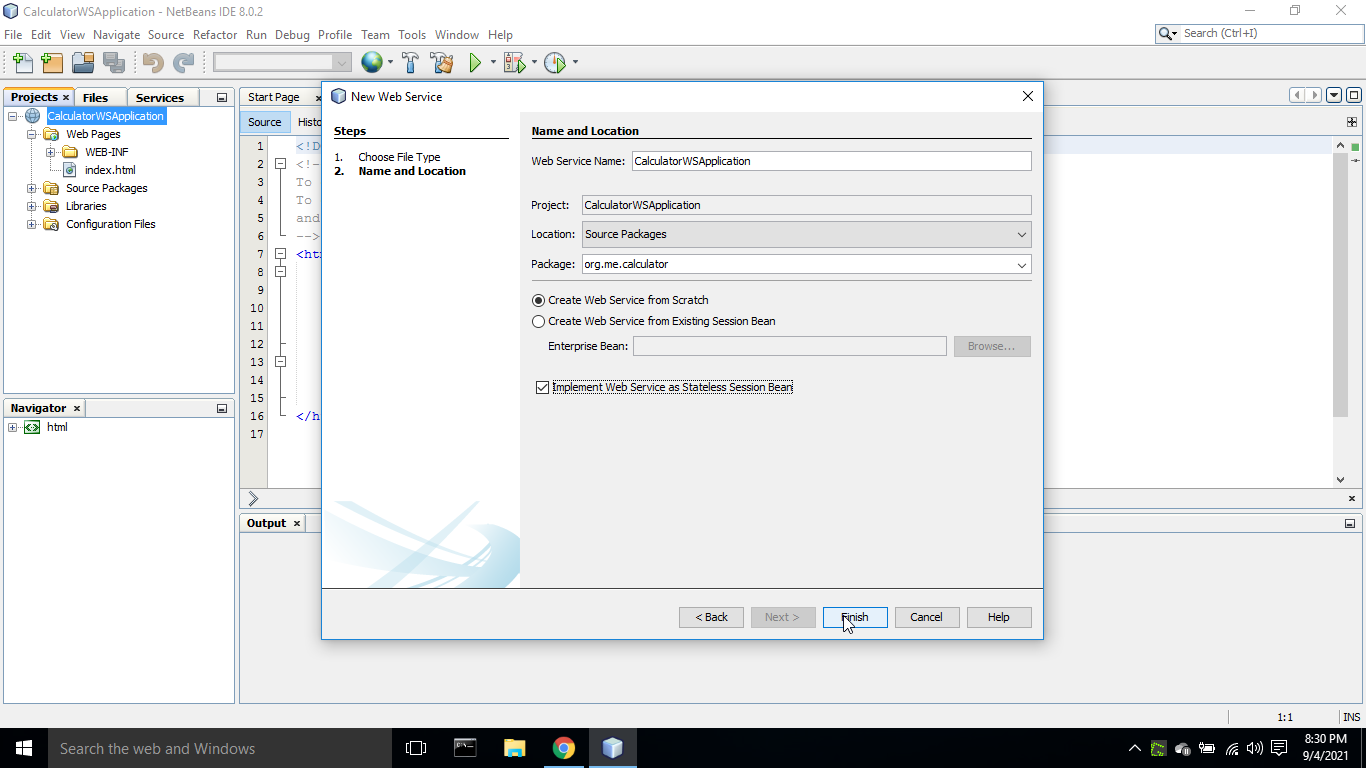
1. Choose File > New Project. Select Web Application from the Java Web.
2. Name the project CalculatorWSApplication. Select a location for the project. Click Next.
3. Select your server and Java EE version and click Finish.

# Creating a Web Service from a Java Class

1. Right-click the CalculatorWSApplication node and choose New > Web Service



2. Name the web service CalculatorWS and type org.me.calculator in Package.



3. Leave Create Web Service from Scratch selected. If you are creating a Java EE 6 project on

GlassFish,

4. select Implement Web Service as a Stateless Session Bean.

5. Click Finish. The Projects window displays the structure of the new web service and the source code is shown in the editor area.

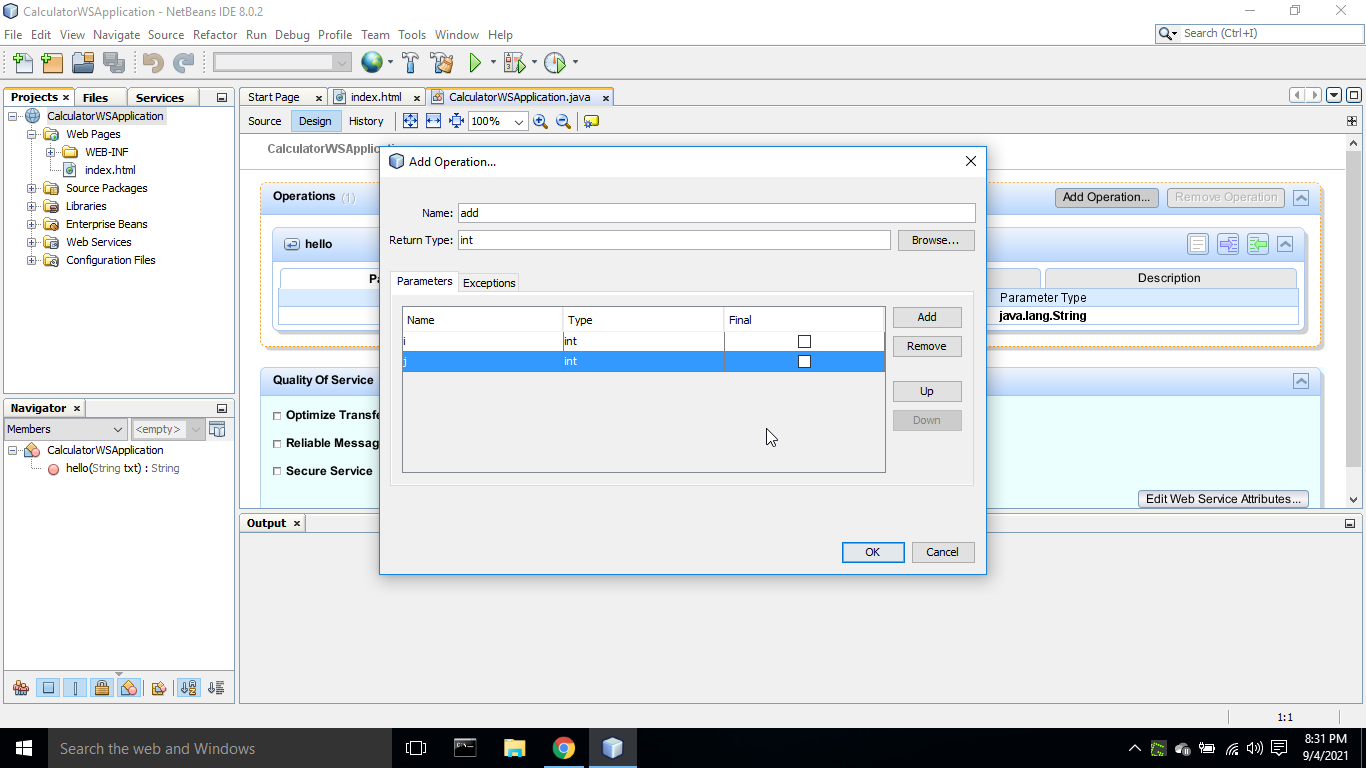
# Adding Operations to the Web Service

1. Change to the Design view in the editor.

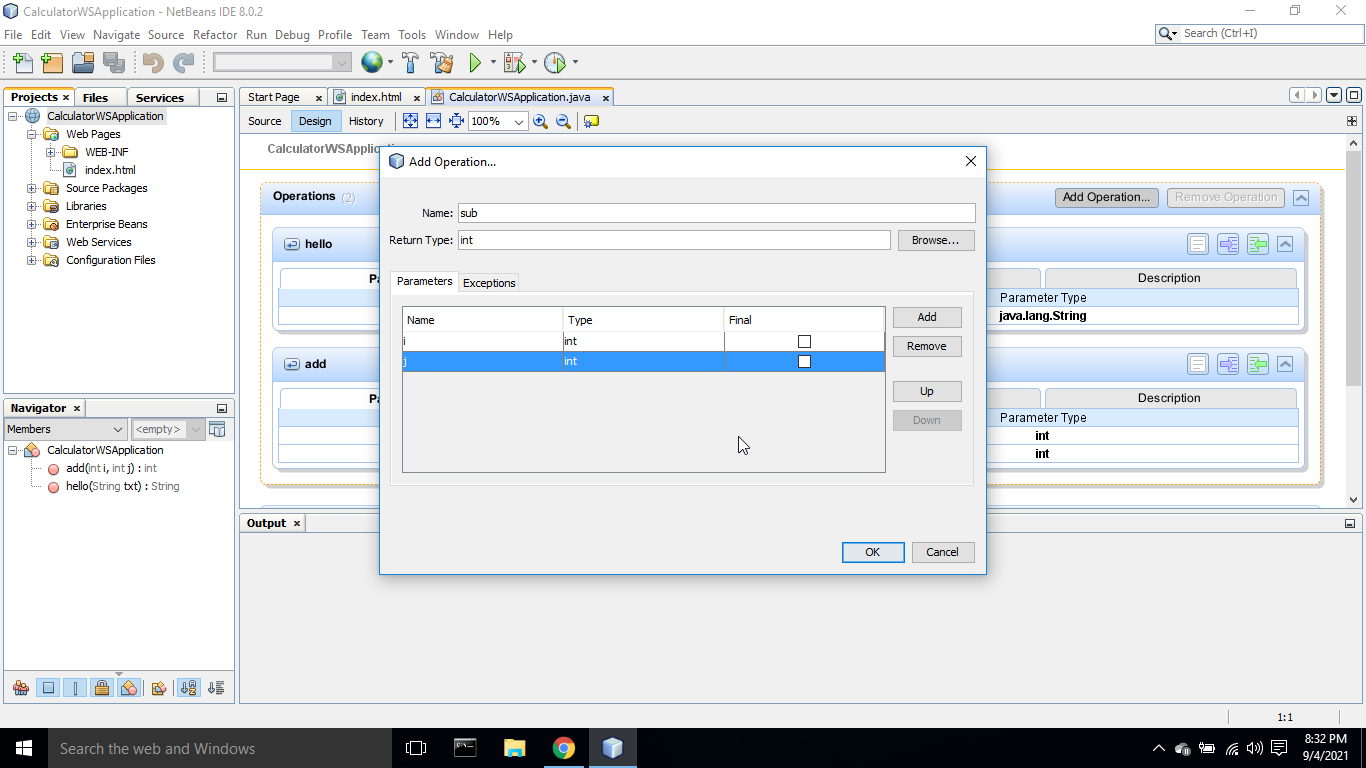
2. Click Add Operation in either the visual designer or the context menu. The Add Operation dialog opens.

3. In the upper part of the Add Operation dialog box, type add in Name and type int in the Return Type drop-down list.

4. In the lower part of the Add Operation dialog box, click Add and create a parameter of type int named i.

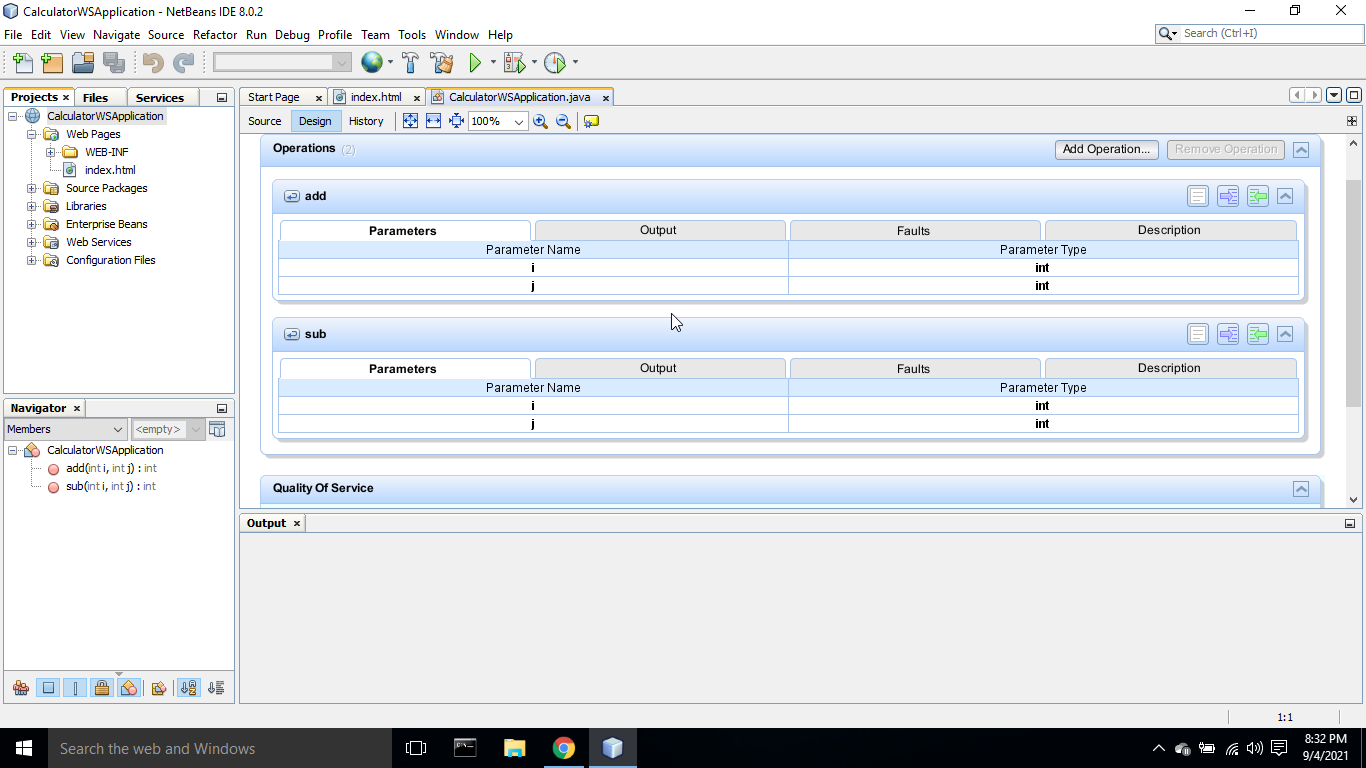
5. Click Add again and create a parameter of type int i called j.

Similarly, Create Operation for sub.



6. Click OK at the bottom of the Add Operation dialog box. You return to the editor.

7. The visual designer now displays the following:



8. Click Source. And code the following.

for add operation:

@WebMethod(operationName = "add")

    public int add(@WebParam(name = "i") int i, @WebParam(name = "j") int j) {

        int add\_num = i + j;

        return add\_num;

    }

for sub operation:

@WebMethod(operationName = "sub")

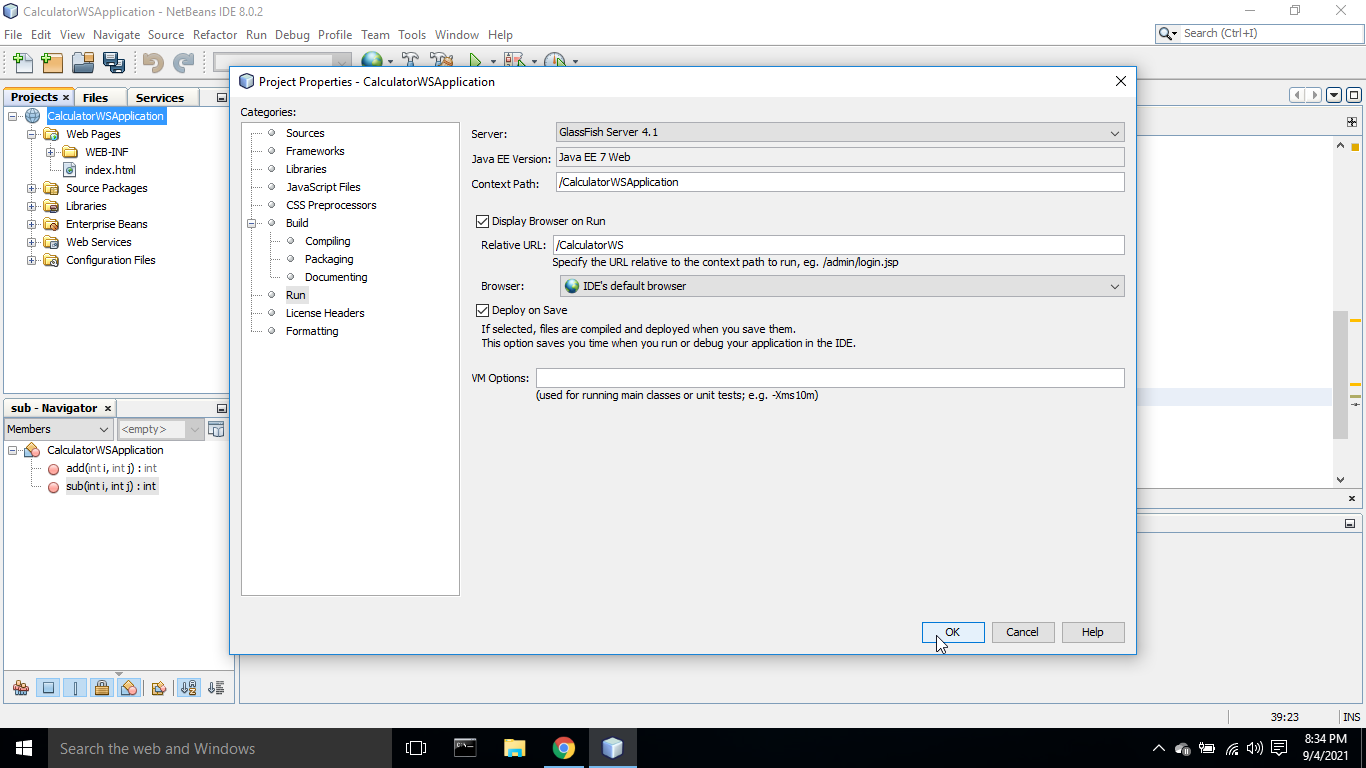
    public int sub(@WebParam(name = "i") int i, @WebParam(name = "j") int j) {

        int sub\_num = i - j;

        return sub\_num;

    }

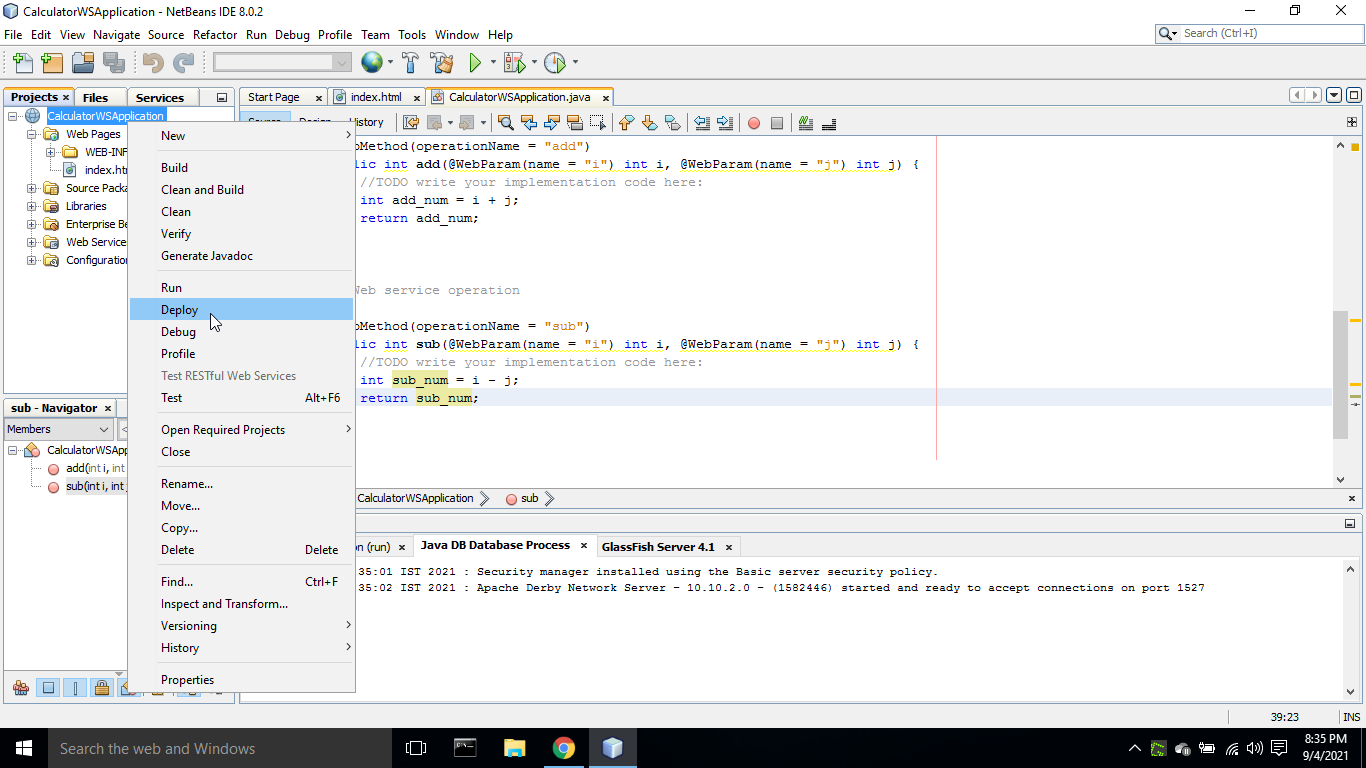
9. Now go to CalculatorWSApplication Properties and Add /CalculatorWS in relative URL in Run Properties and Click OK.



# Deploying and Testing the Web Service

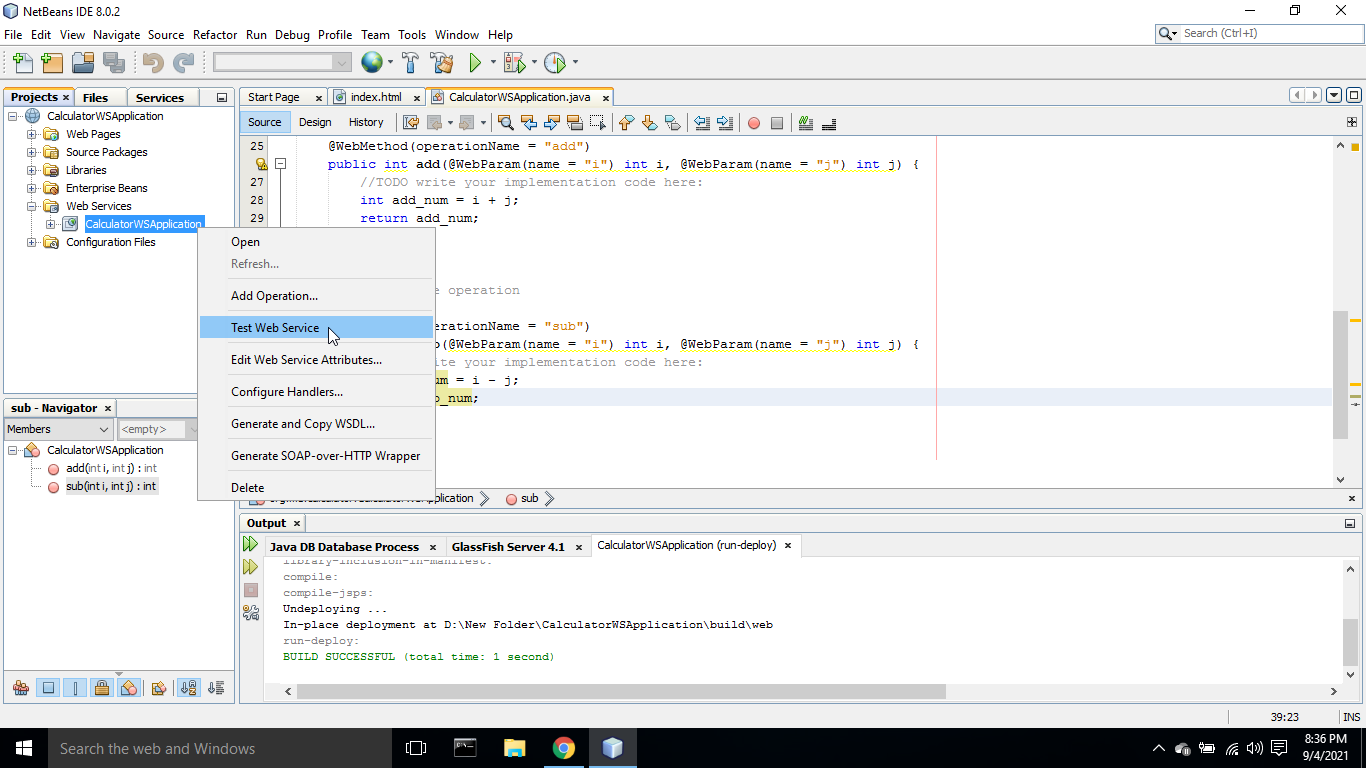
1. Right-click the project and choose Deploy. The IDE starts the application server, builds the

application, and deploys the application to the server



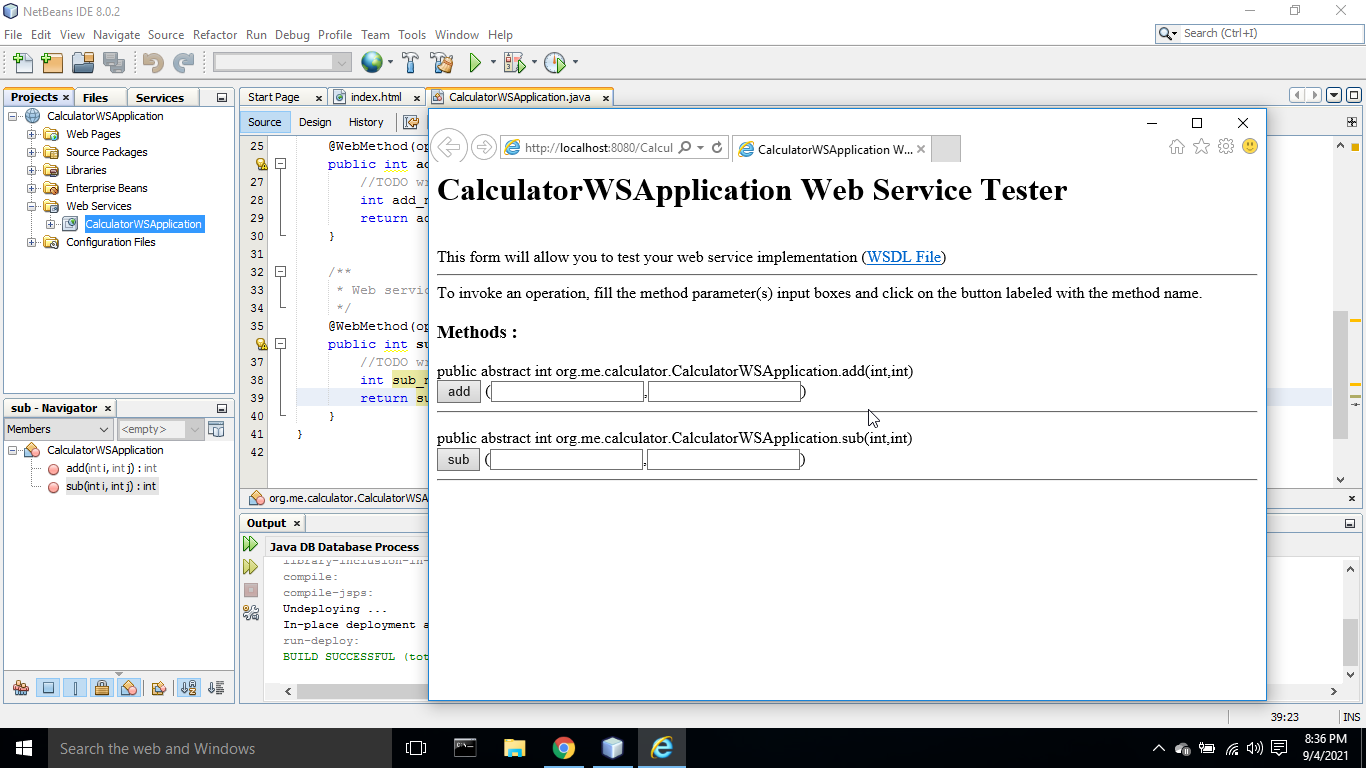
2. In the IDE's Projects tab, expand the Web Services node of the CalculatorWSApplication project.

Right-click the CalculatorWS node, and choose Test Web Service

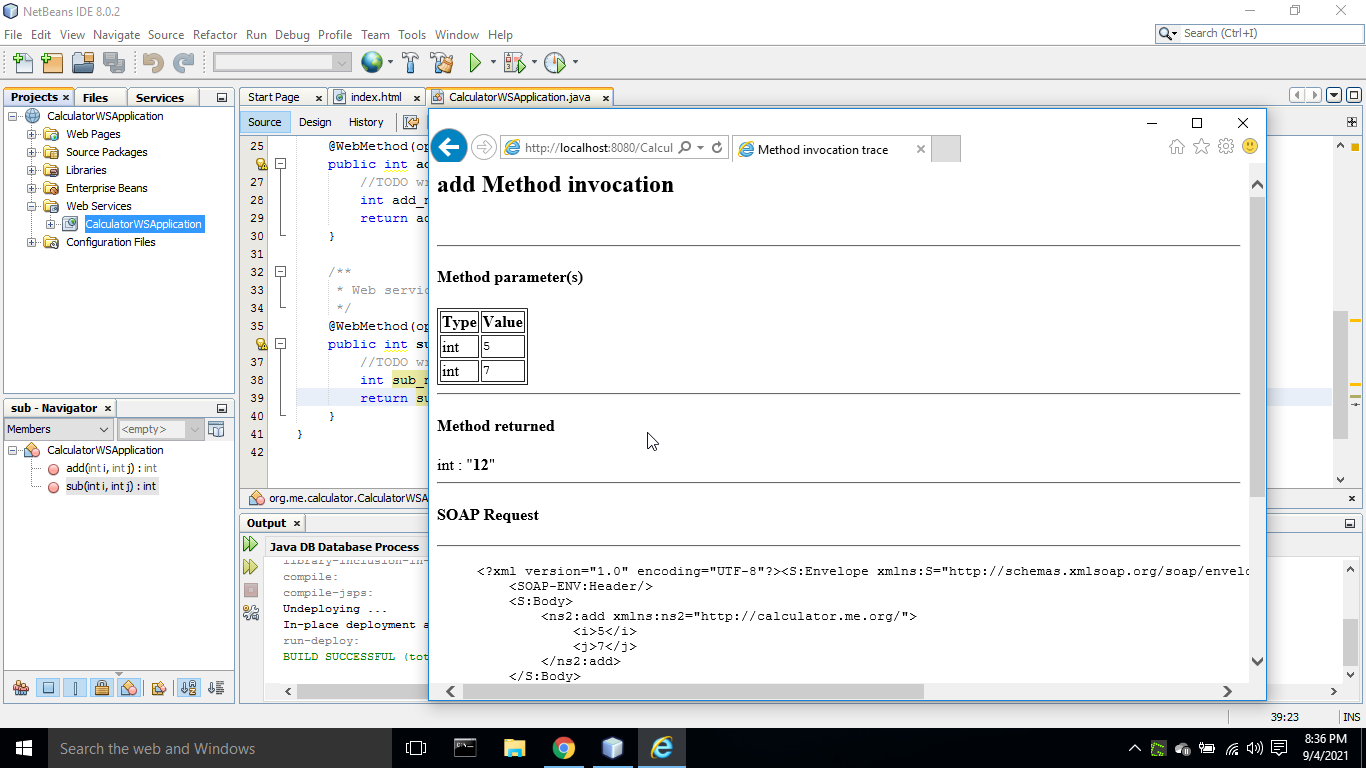


3. The IDE opens the tester page in your browser, if you deployed a web application to the GlassFish

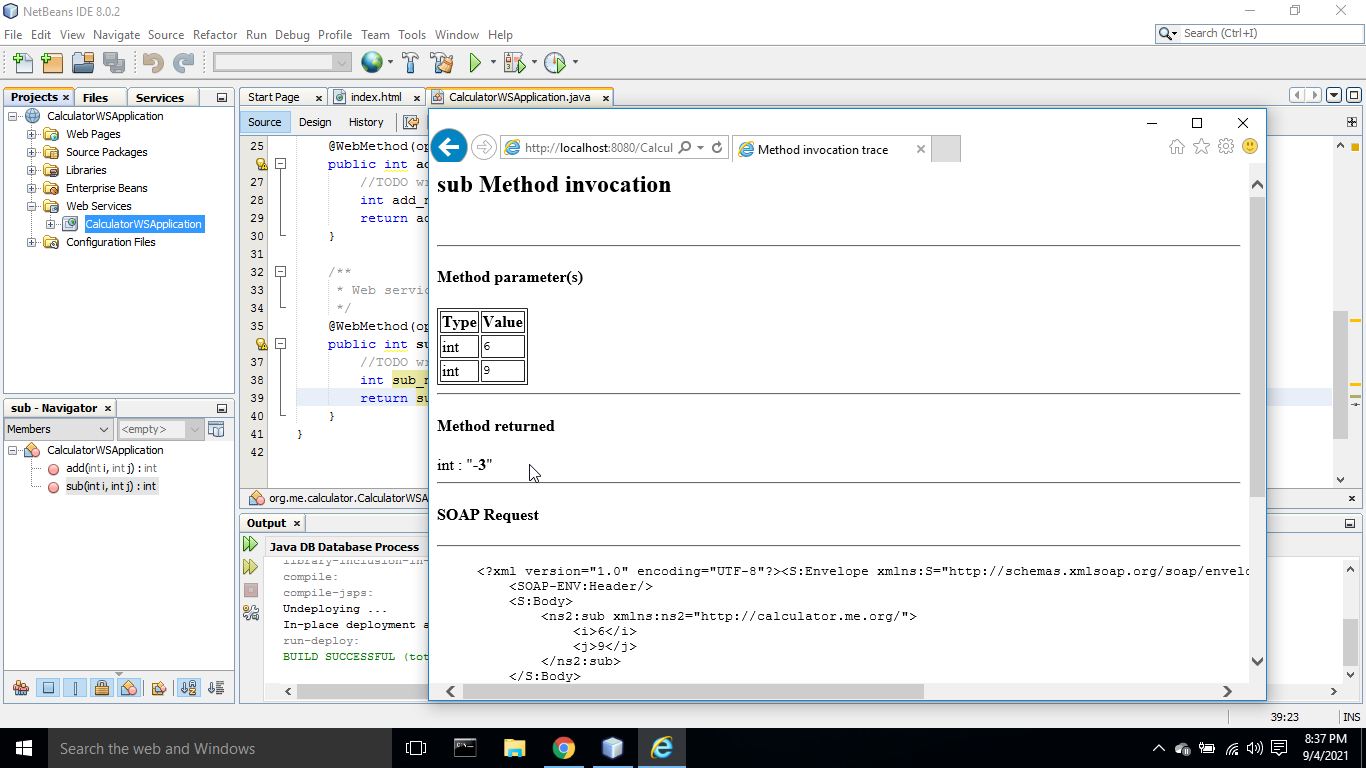
server.



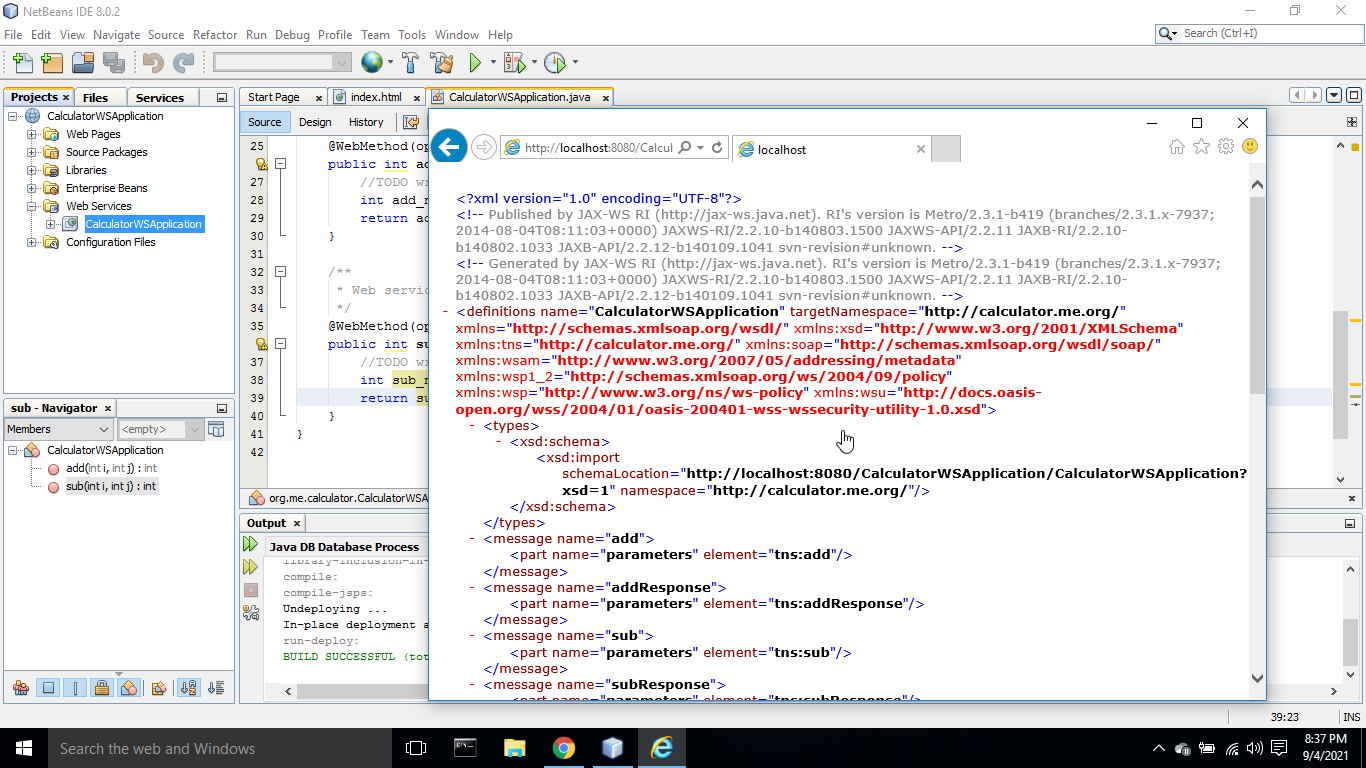
Add operation:



Sub operation:

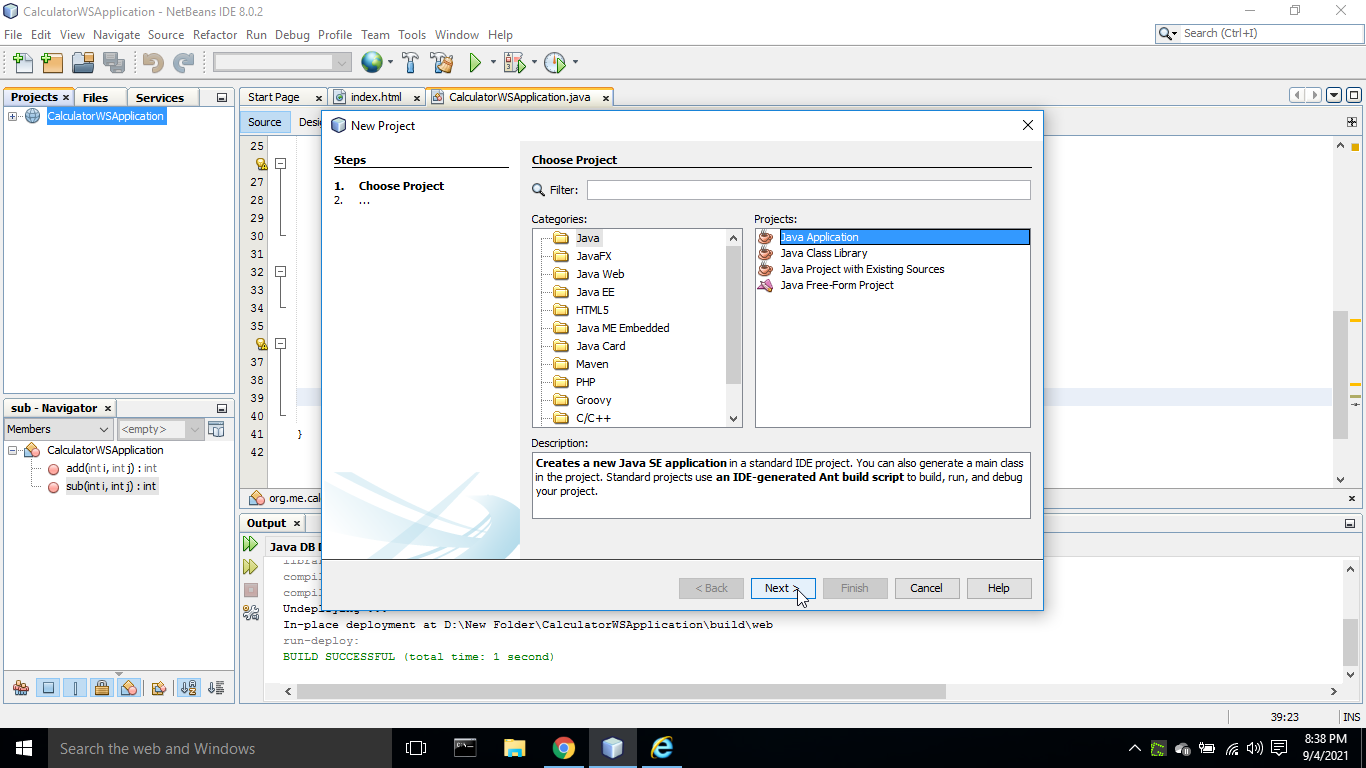


WSDL File



# Consuming the Webservice

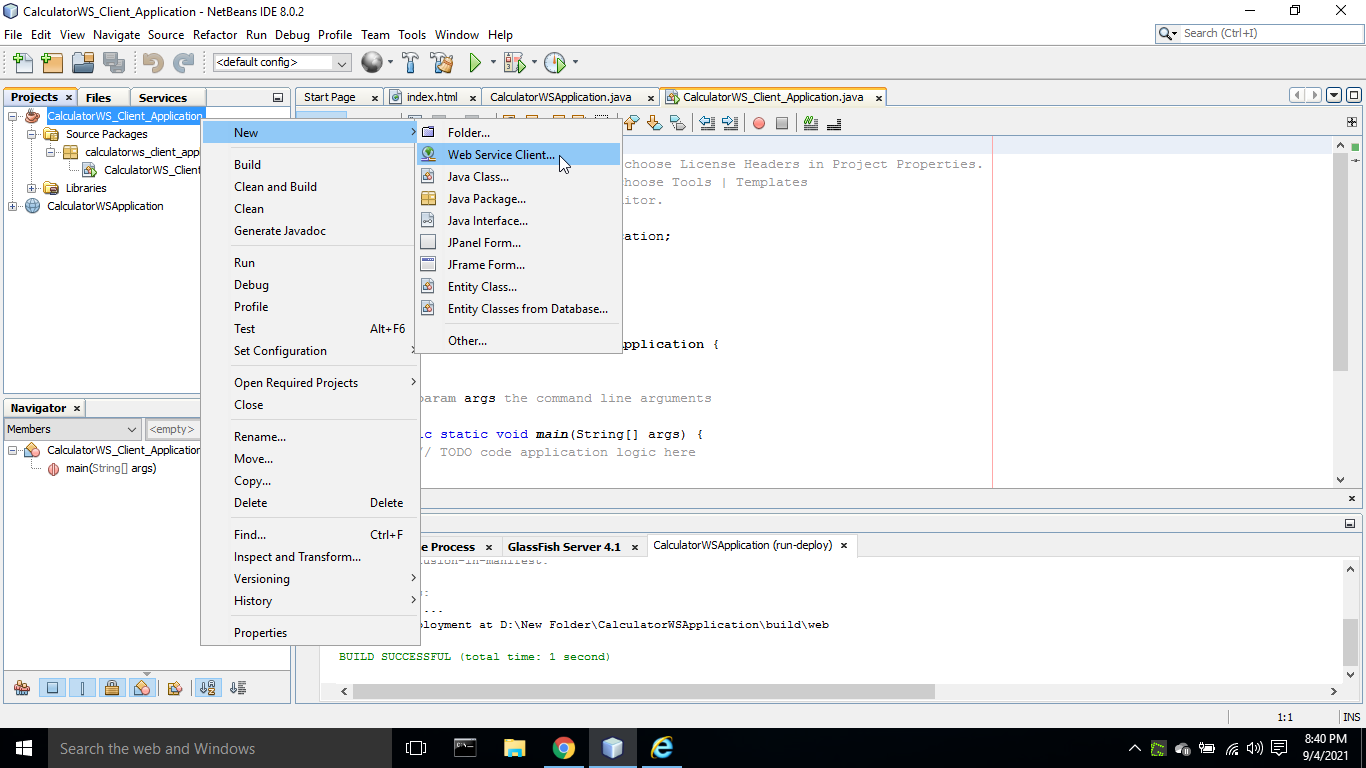
1. Choose File > New Project. Select Java Application from the Java category.



2. Name the project CalculatorWS\_Client\_Application.

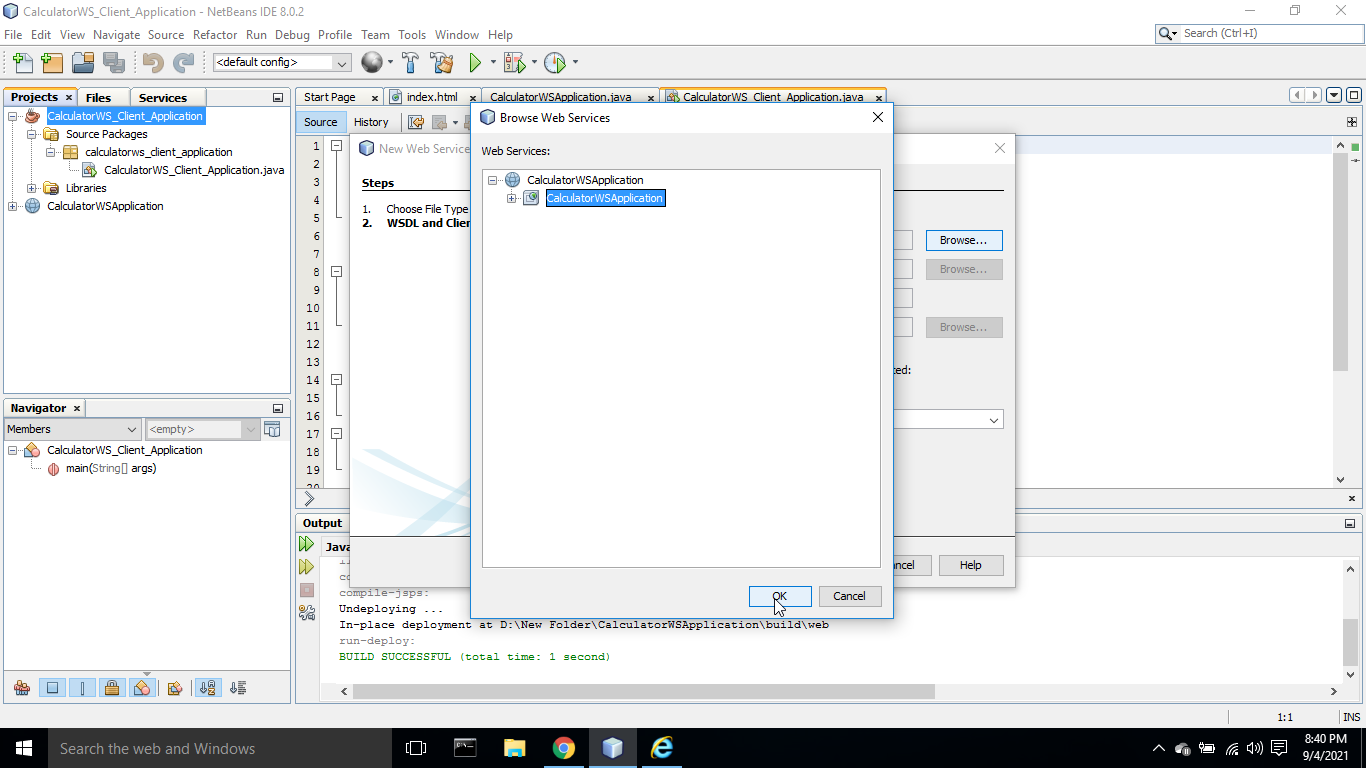
3. Leave Create Main Class selected and accept all other default settings. Click Finish.

4. Right Click CalculatorWS\_Client\_Application > New > Web Service Client

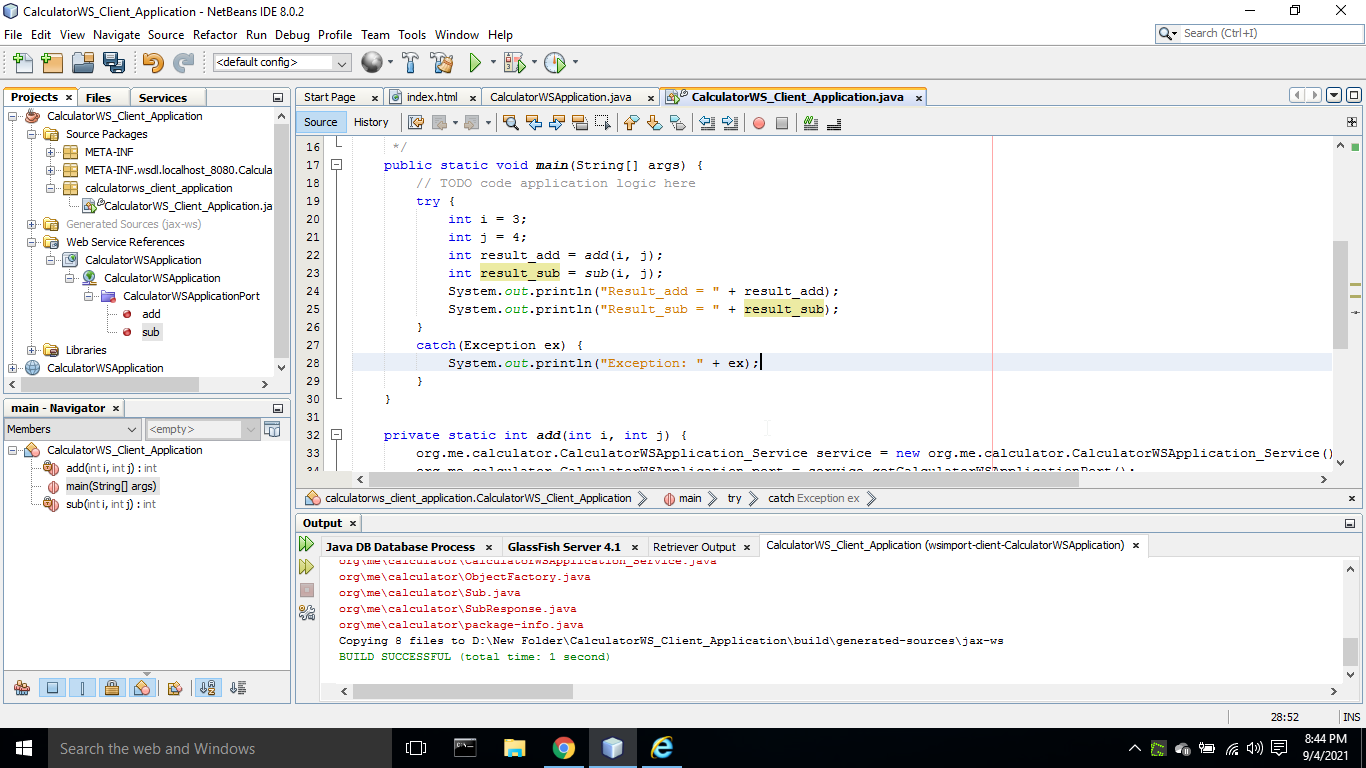


5. Click Browse > Select CalculatorWSApplication to add operations in client application

Now Click OK and Finish.



6. Now you add and sub operation nodes are added in CalculatorWS\_Client\_Application > Web Service Refrence folder



7. Now drag and drop add and sub nodes in Calculator\_WS\_Client\_Application source code editor.

8. In main() class add the following code:

try {

            int i = 3;

            int j = 4;

            int result\_add = add(i, j);

            int result\_sub = sub(i, j);

            System.out.println("Result\_add = " + result\_add);

            System.out.println("Result\_sub = " + result\_sub);

        }

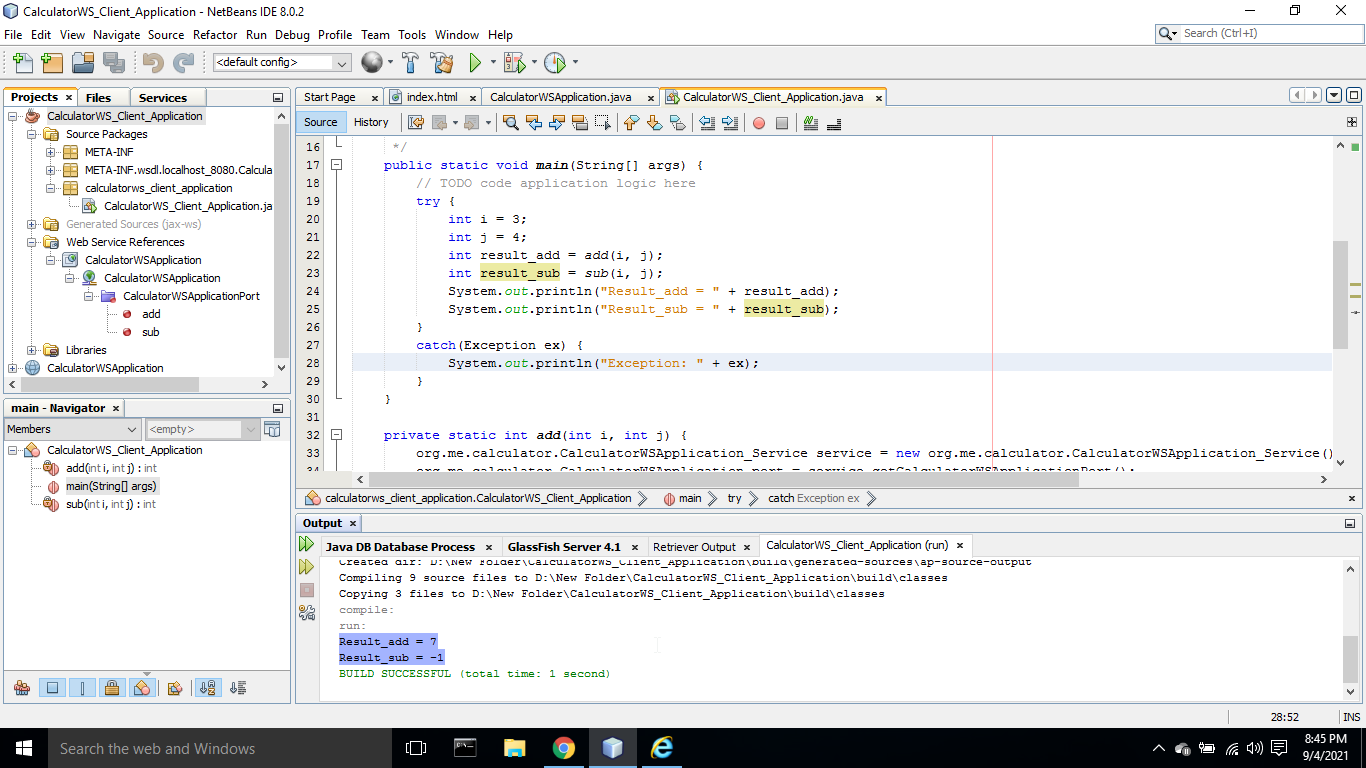
catch(Exception ex) {

            System.out.println("Exception: " + ex);

        }

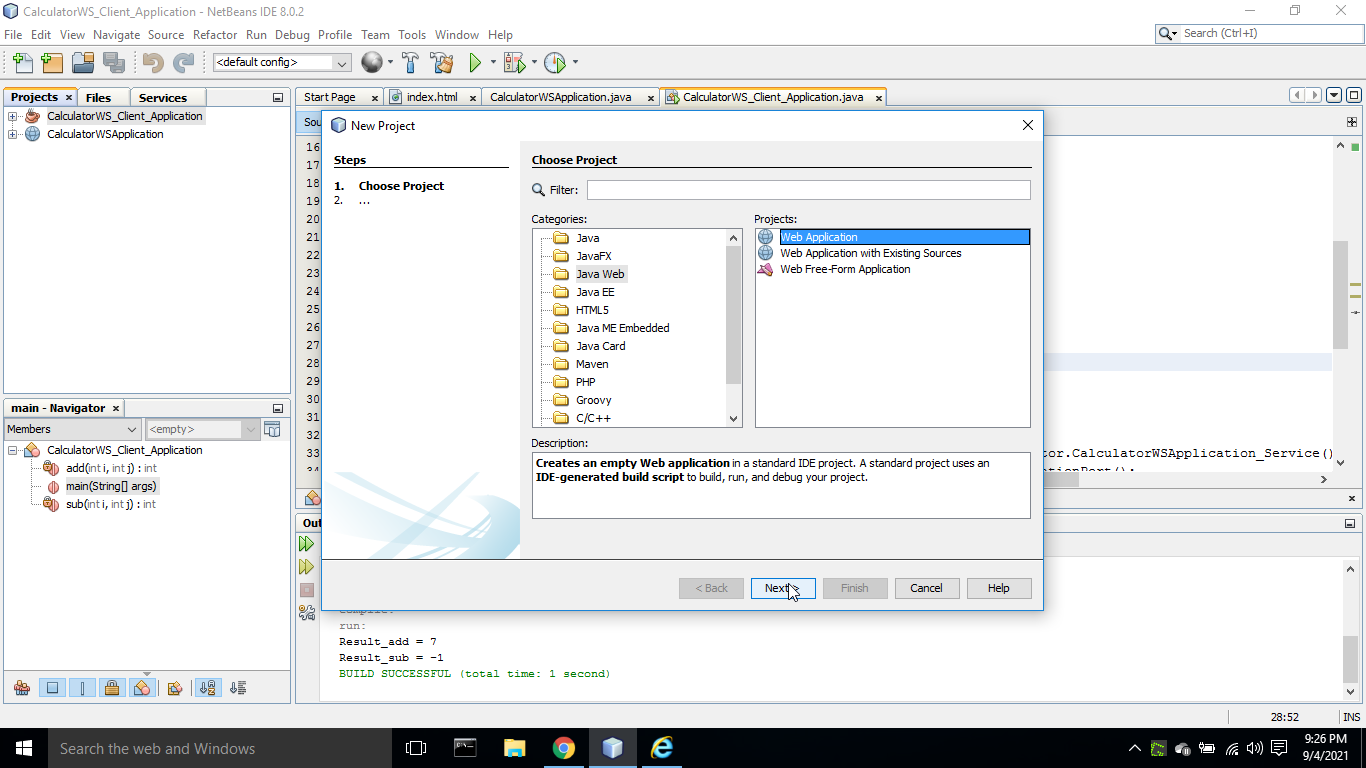
9. Right click on CalulcatorWS\_Client\_Application > Select Run

To get following output in Terminal

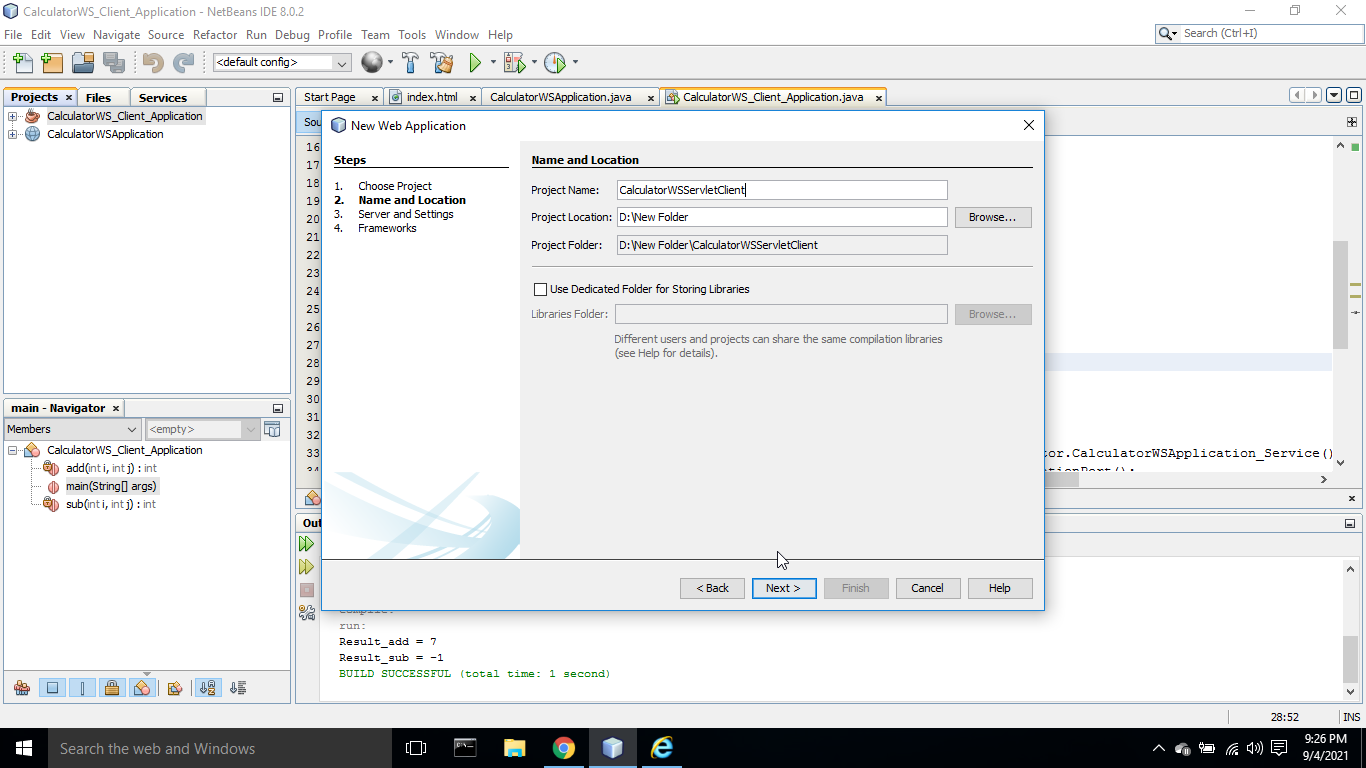


# Implement ServletClient

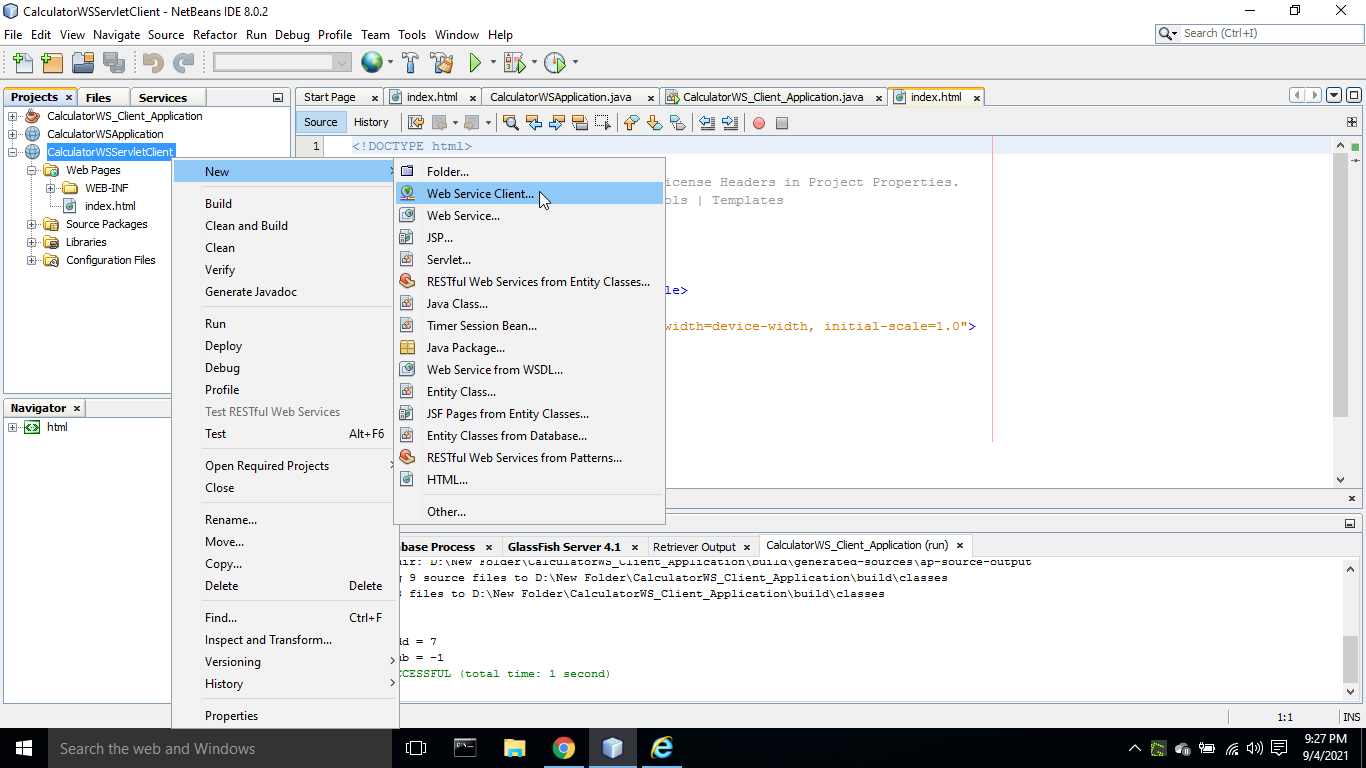
1. New Project > Java Web > Web Application



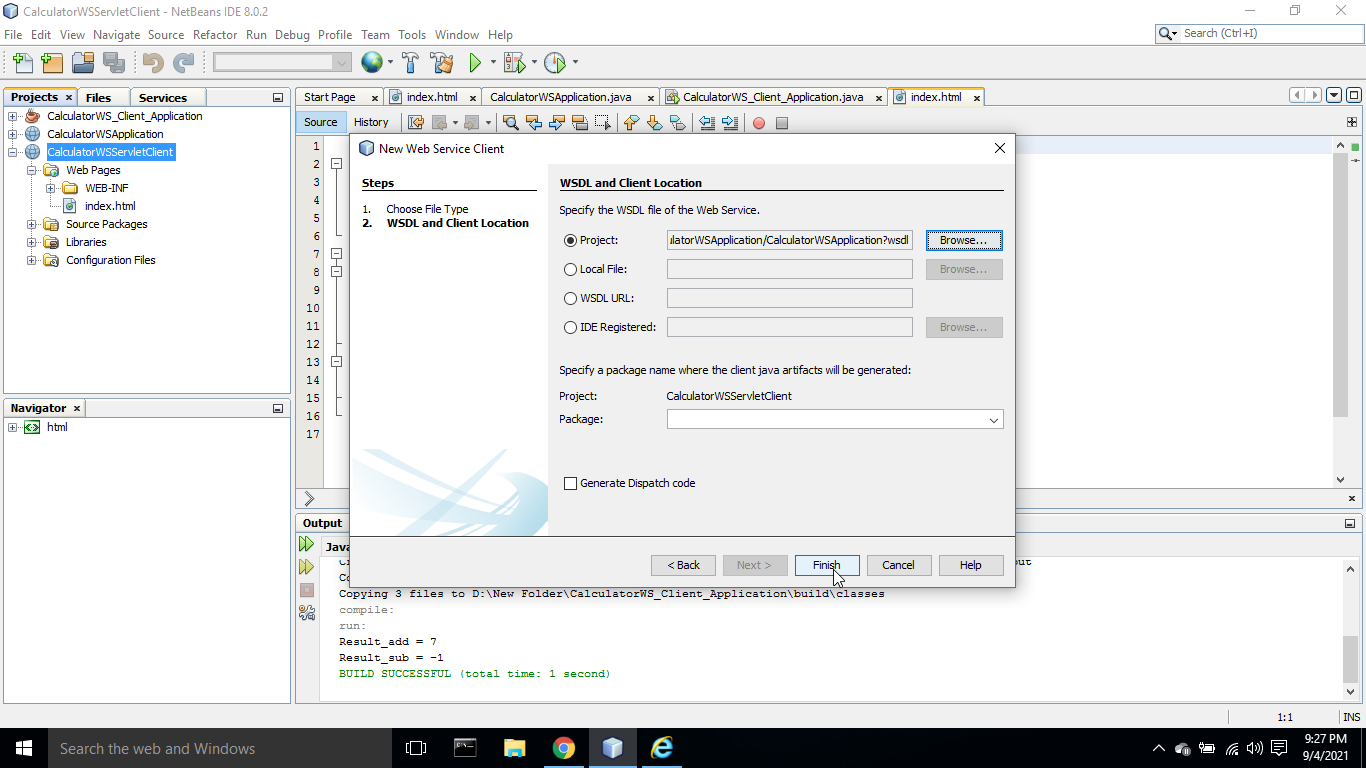
2. Give Project Name CalculatorWSServletClient keep other option as default and Finish.



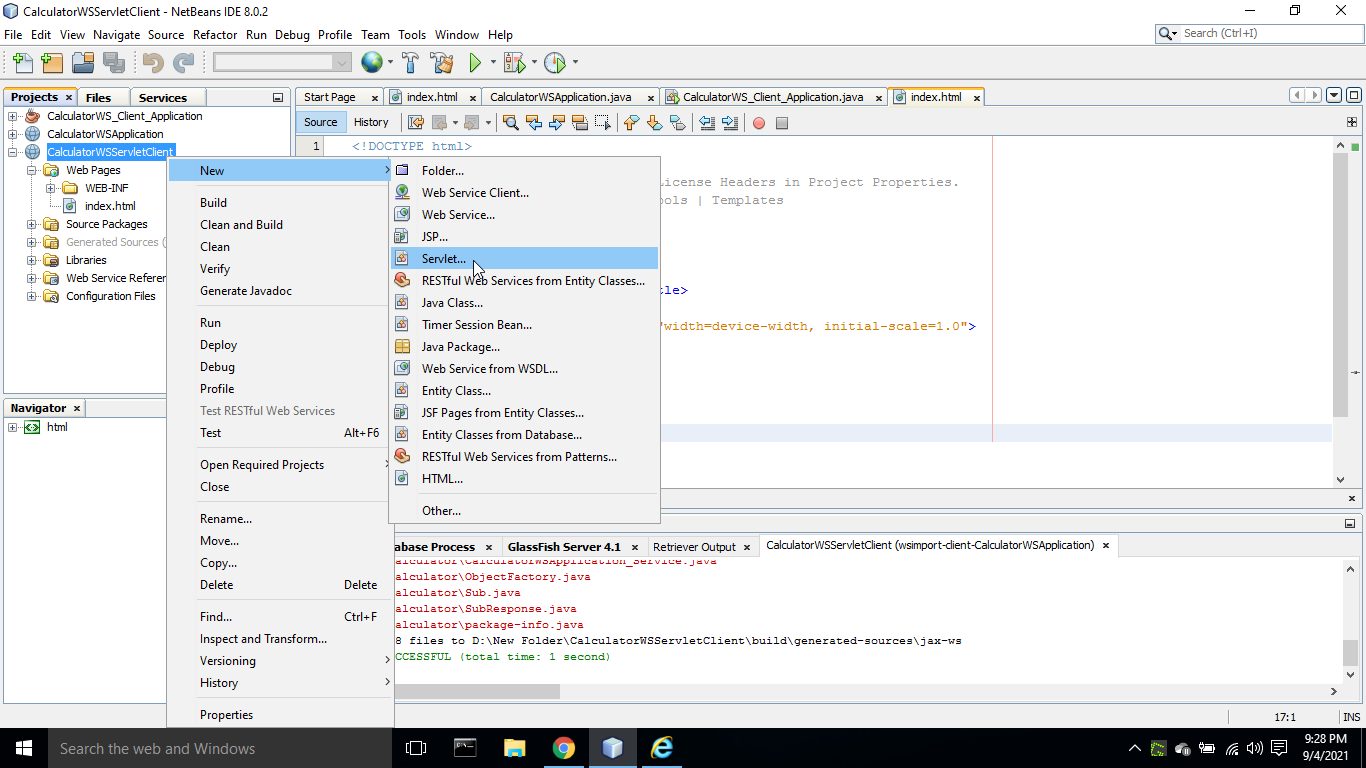
3. Right Click CalculatorWSServletClient > New > Web Service Client



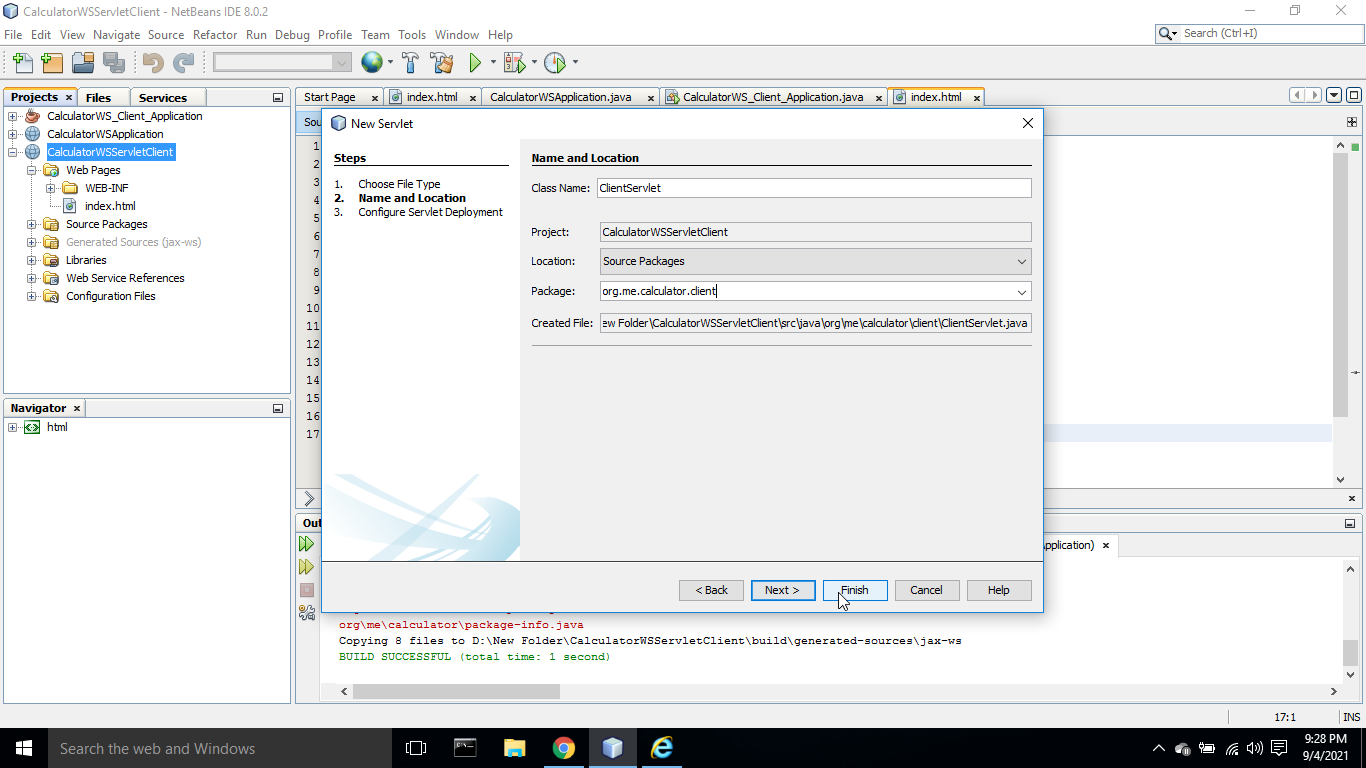
4. Choose Browse from pop up Window > Select CalculatorWSApplication to add Web Service Reference Folder with add and sub operation nodes

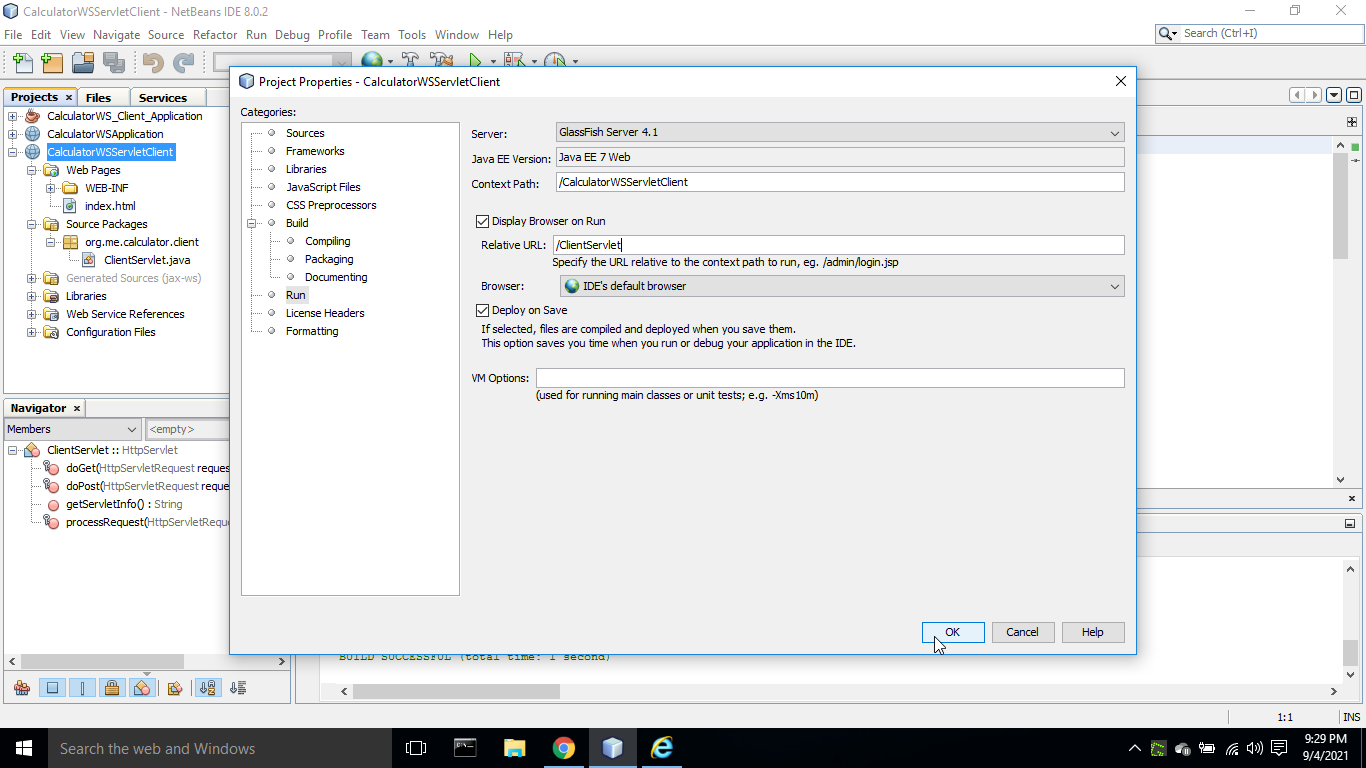


5. Right Click CalculatorWSServletClient > New > Servlet



6. Name class as ClientServlet and Package as org.me.calculator.client and leave other option default



7. Right Click CalculatorWSServletClient > Properties > Run

8. write /ClientServlet in relative URL > Click OK and

Now Drag and Drop add and sub operation nodes from Web Service Refrence Folder in ClientServlet.java file and add the following code in ClientServlet.java main() class source code

try {

            int i = 3;

            int j = 4;

            int result\_add = add(i, j);

            int result\_sub = sub(i, j);

            out.println("Result\_add = " + result\_add);

            out.println("Result\_sub = " + result\_sub);

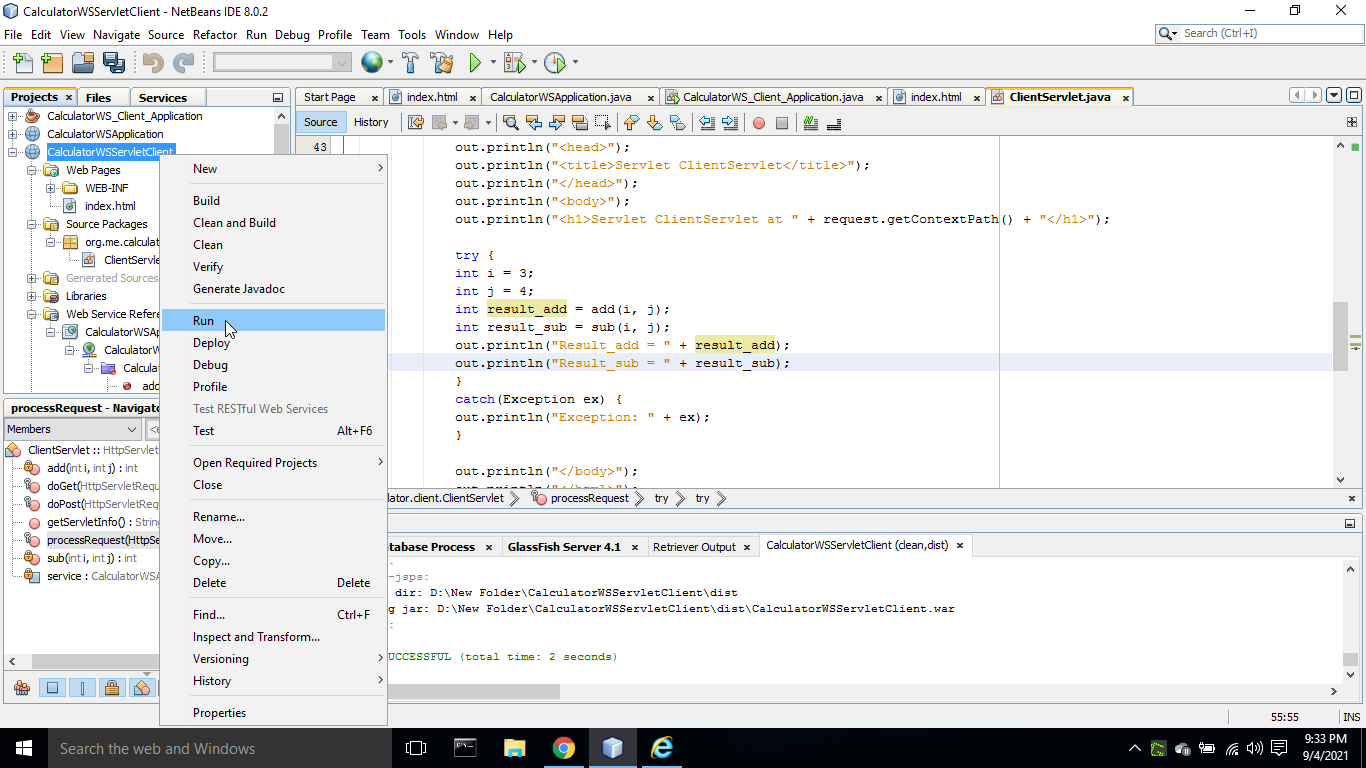
            }

         catch(Exception ex) {

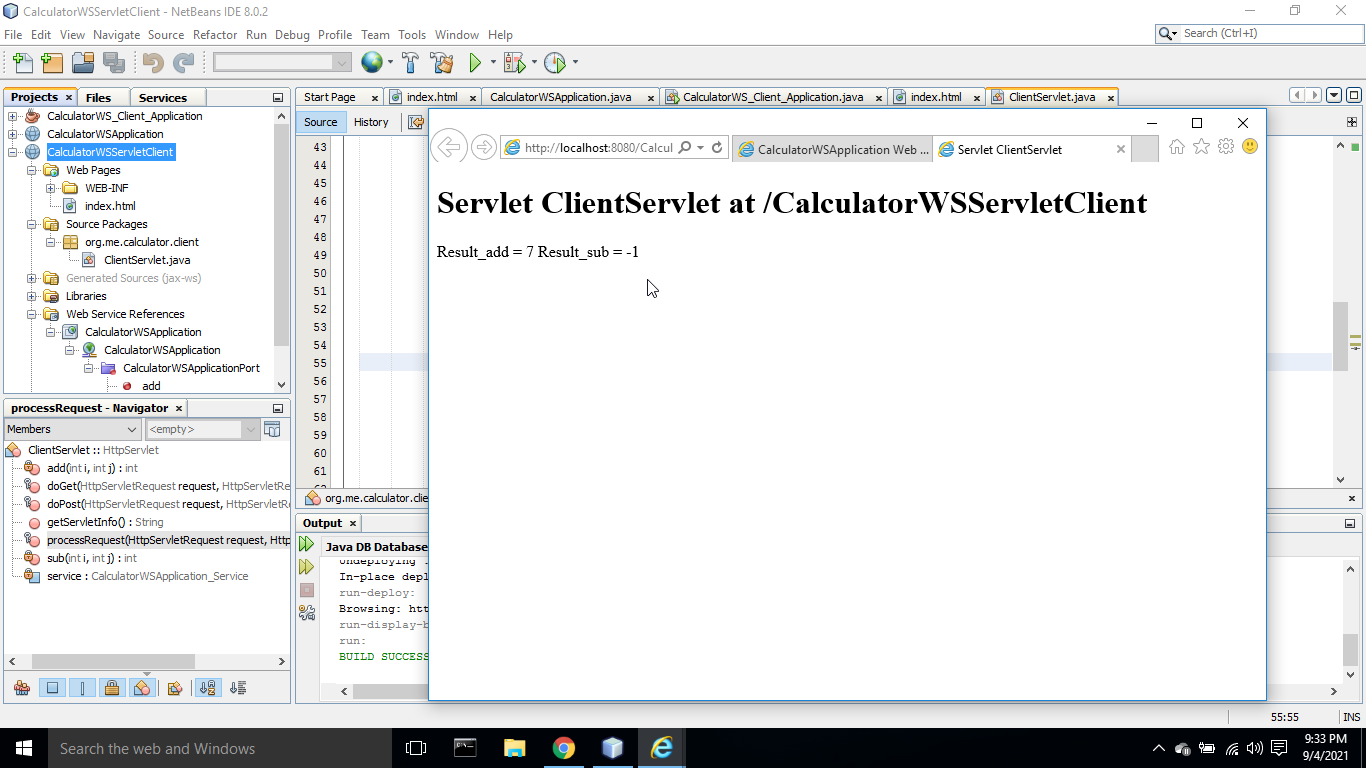
            out.println("Exception: " + ex);

            }

9. Right Click CalculatorWSServletClient > Run

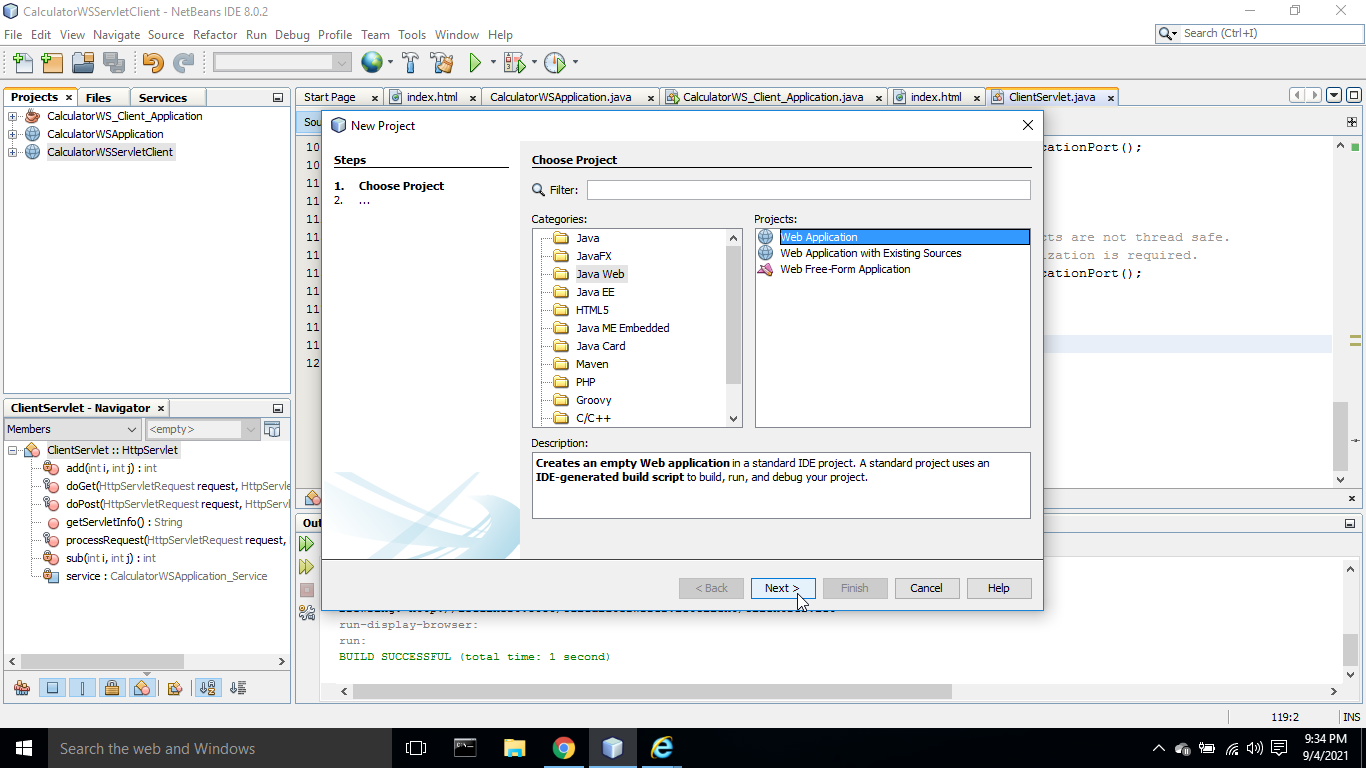


10. to get following output popup

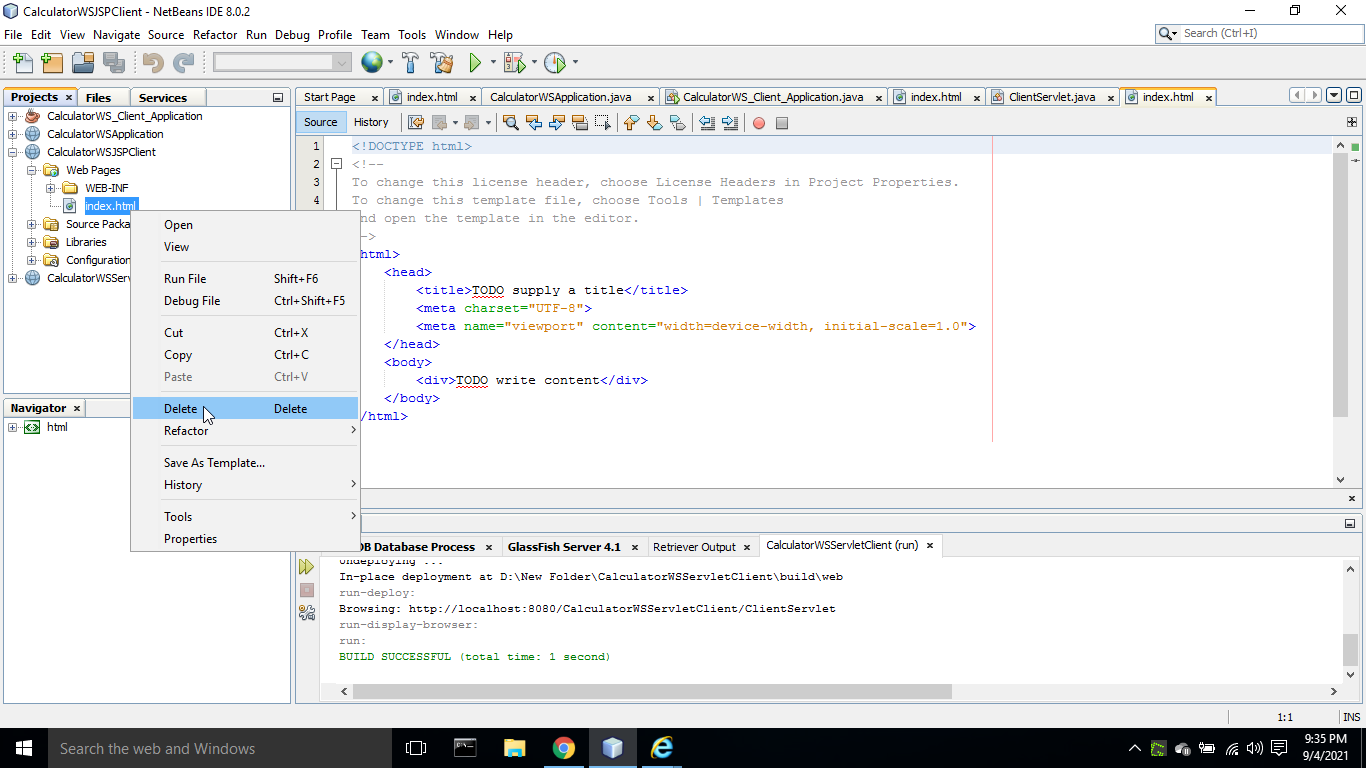


# JSP Page in Web Application

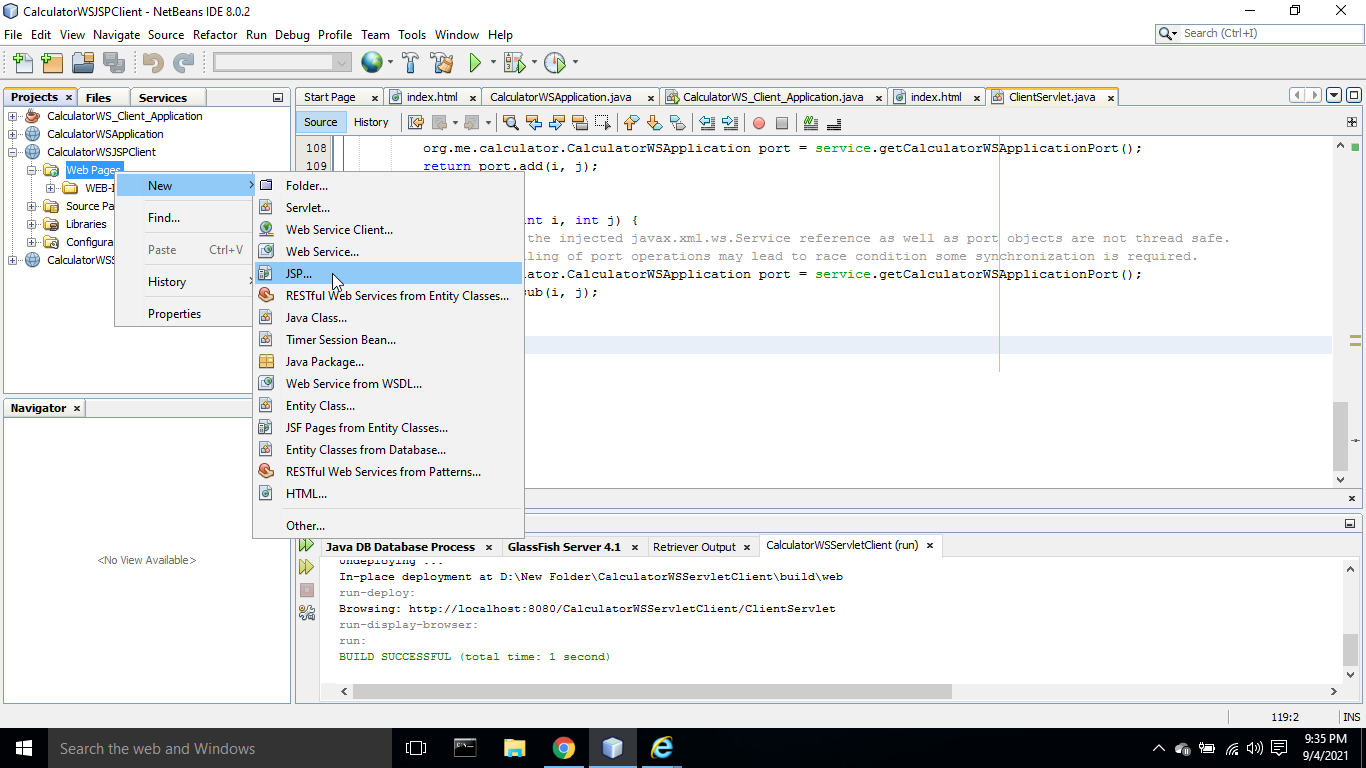
1. New Project > Java Web > Web Application and Name it as CalculatorWSJSPClient



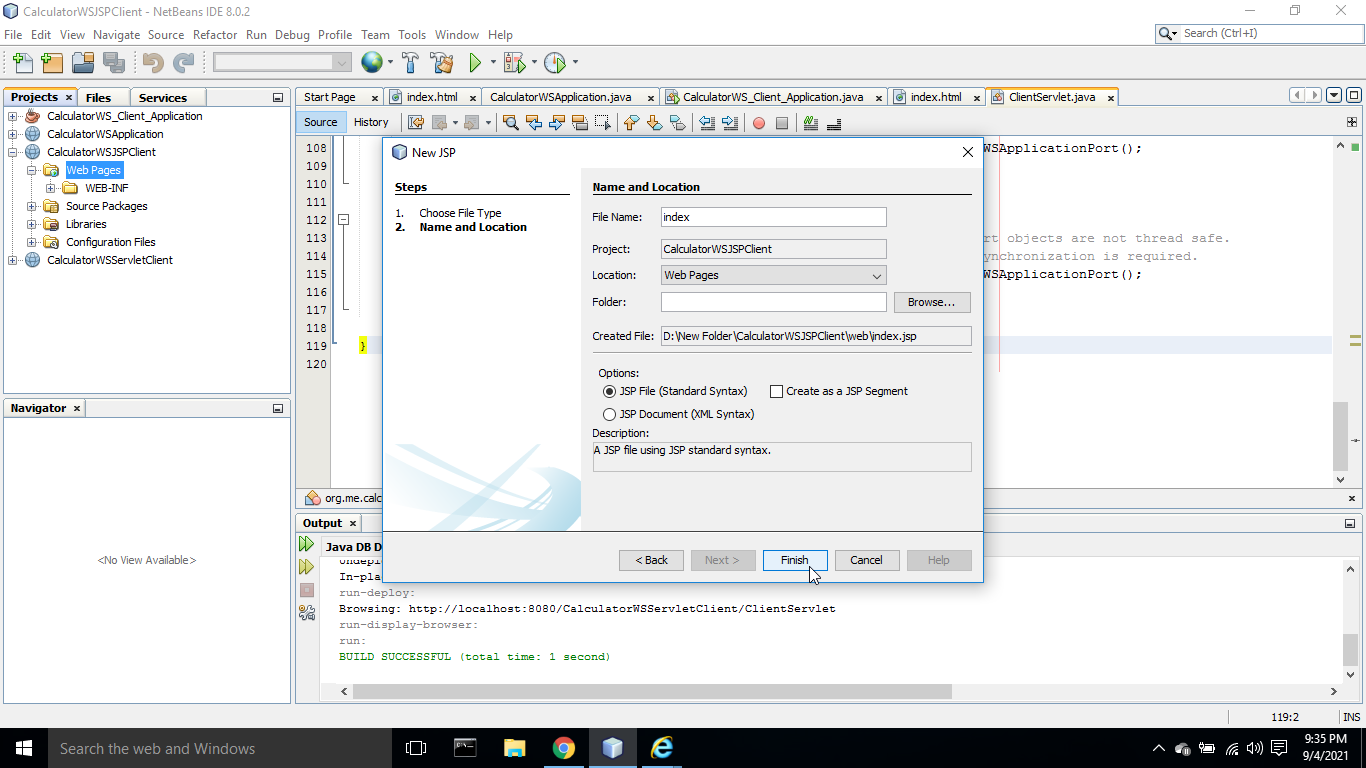
2. CalculatorWSJSPClient > Web Pages and Delete index.html file



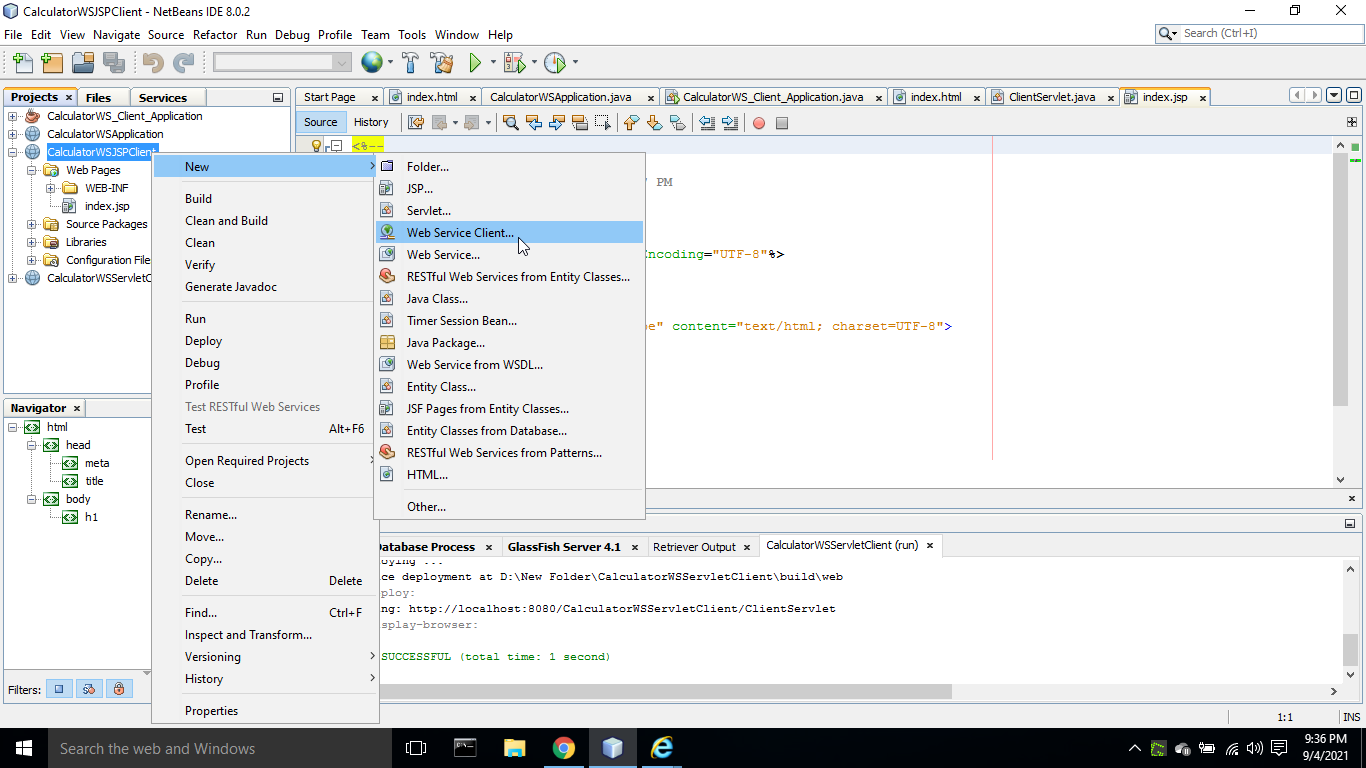
3. CalculatorWSJSPClient > Web Pages and Right Click on Web Pages and choose JSP



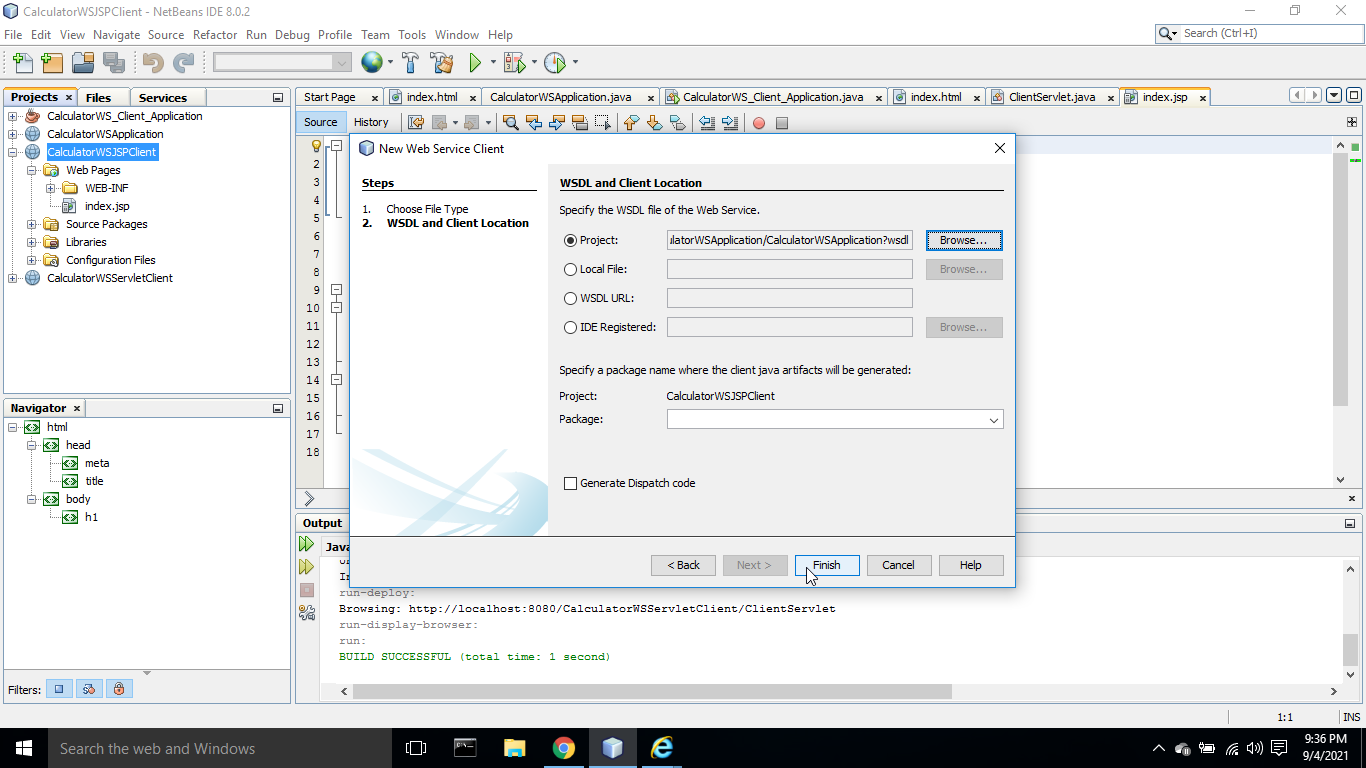
4. set File Name as index and Choose Options as JSP File



5. Right Click on CalculatorWSJSPClient > New > Web Service Client

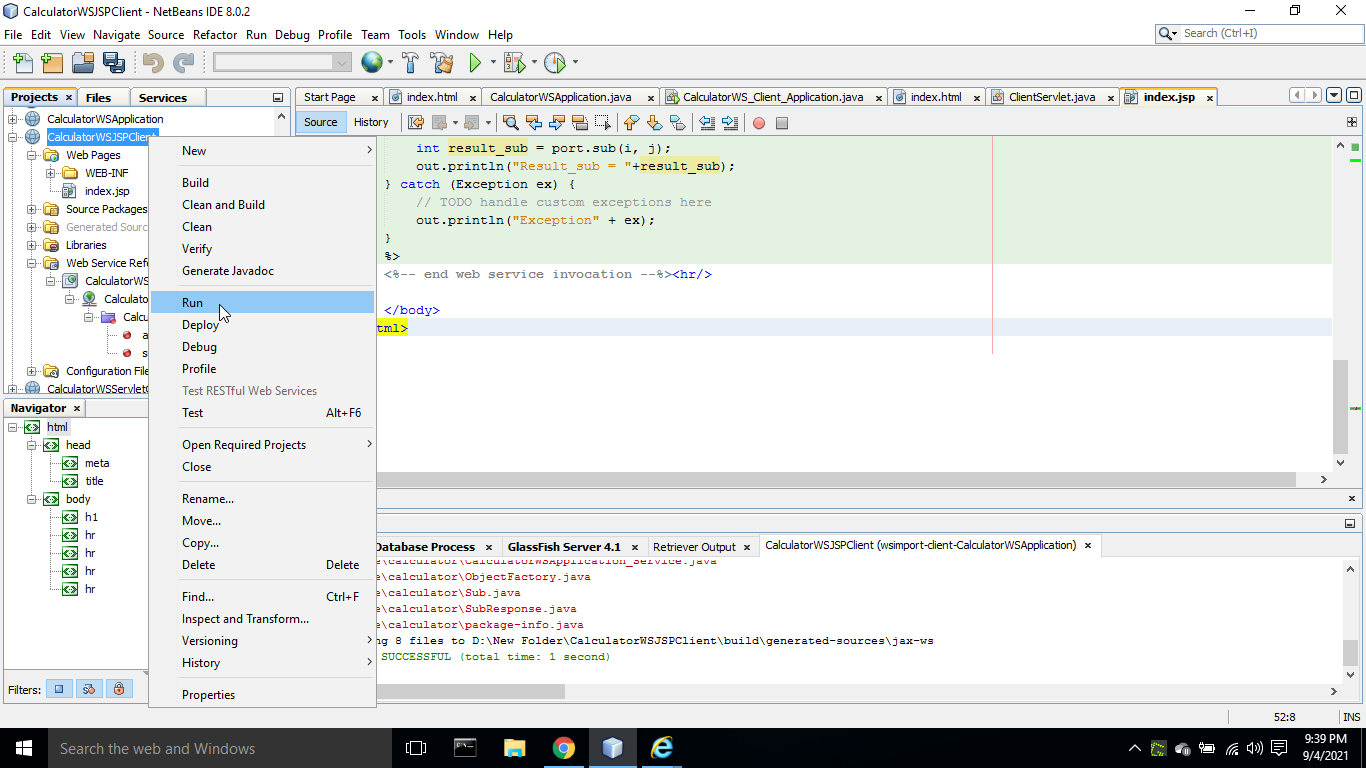


6. under WSDL and Client Location > Browse > Select CalculatorWSApplication

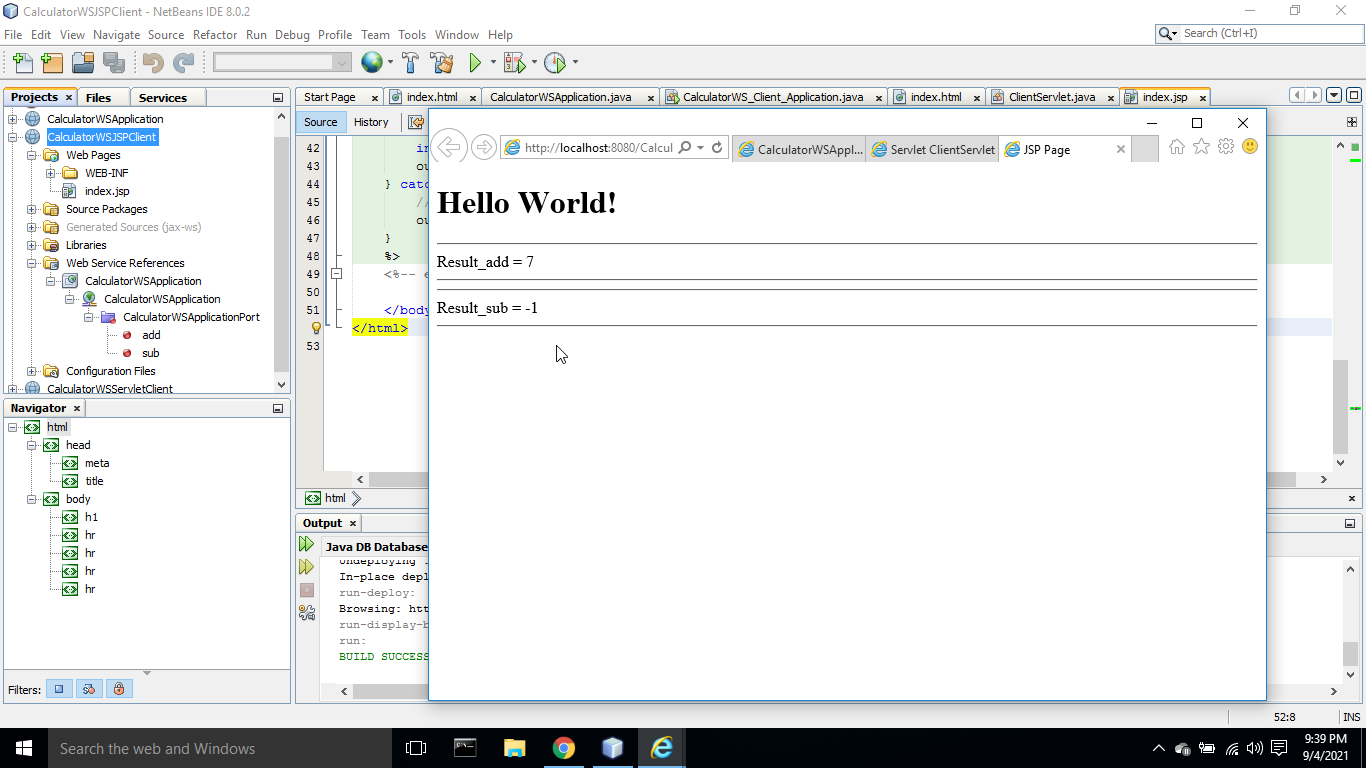


7. Drag and Drop add and sub operation from CalculatorWSJSPClient > Web Service Reference in index.jsp source code editor inside <h1> tag inside <body> tag to add operations in jsp file and specify values for i and j.

8. Right Click CalculatorWSJSPClient > Run



8. JSP output for both add and sub operations



# Source Codes :

### CalculatorWSApplication file source code:

/\*

 \* 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.

 \*/

package org.me.calculator;

import javax.jws.WebService;

import javax.jws.WebMethod;

import javax.jws.WebParam;

import javax.ejb.Stateless;

/\*\*

 \*

 \* @author yashchandra05

 \*/

@WebService(serviceName = "CalculatorWSApplication")

@Stateless()

public class CalculatorWSApplication {

    /\*\*

     \* Web service add operation

     \*/

    @WebMethod(operationName = "add")

    public int add(@WebParam(name = "i") int i, @WebParam(name = "j") int j) {

        int add\_num = i + j;

        return add\_num;

    }

    /\*\*

     \* Web service sub operation

     \*/

    @WebMethod(operationName = "sub")

    public int sub(@WebParam(name = "i") int i, @WebParam(name = "j") int j) {

        int sub\_num = i - j;

        return sub\_num;

    }

}

### CalculatorWS\_Client\_Application file Source Code:

/\*

 \* 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.

 \*/

package calculatorws\_client\_application;

/\*\*

 \*

 \* @author yashchandra05

 \*/

public class CalculatorWS\_Client\_Application {

    /\*\*

     \* @param args the command line arguments

     \*/

    public static void main(String[] args) {

        try {

            int i = 3;

            int j = 4;

            int result\_add = add(i, j);

            int result\_sub = sub(i, j);

            System.out.println("Result\_add = " + result\_add);

            System.out.println("Result\_sub = " + result\_sub);

        }

        catch(Exception ex) {

            System.out.println("Exception: " + ex);

        }

    }

    private static int add(int i, int j) {

        org.me.calculator.CalculatorWSApplication\_Service service = new org.me.calculator.CalculatorWSApplication\_Service();

        org.me.calculator.CalculatorWSApplication port = service.getCalculatorWSApplicationPort();

        return port.add(i, j);

    }

    private static int sub(int i, int j) {

        org.me.calculator.CalculatorWSApplication\_Service service = new org.me.calculator.CalculatorWSApplication\_Service();

        org.me.calculator.CalculatorWSApplication port = service.getCalculatorWSApplicationPort();

        return port.sub(i, j);

    }

}

### ClientServlet.java file Source Code:

/\*

 \* 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.

 \*/

package org.me.calculator.client;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.xml.ws.WebServiceRef;

import org.me.calculator.CalculatorWSApplication\_Service;

/\*\*

 \*

 \* @author yashchandra05

 \*/

@WebServlet(name = "ClientServlet", urlPatterns = {"/ClientServlet"})

public class ClientServlet extends HttpServlet {

    @WebServiceRef(wsdlLocation = "WEB-INF/wsdl/localhost\_8080/CalculatorWSApplication/CalculatorWSApplication.wsdl")

    private CalculatorWSApplication\_Service service;

    /\*\*

     \* Processes requests for both HTTP <code>GET</code> and <code>POST</code>

     \* methods.

     \*

     \* @param request servlet request

     \* @param response servlet response

     \* @throws ServletException if a servlet-specific error occurs

     \* @throws IOException if an I/O error occurs

     \*/

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)

            throws ServletException, IOException {

        response.setContentType("text/html;charset=UTF-8");

        try (PrintWriter out = response.getWriter()) {

            out.println("<!DOCTYPE html>");

            out.println("<html>");

            out.println("<head>");

            out.println("<title>Servlet ClientServlet</title>");

            out.println("</head>");

            out.println("<body>");

            out.println("<h1>Servlet ClientServlet at " + request.getContextPath() + "</h1>");

            try {

            int i = 3;

            int j = 4;

            int result\_add = add(i, j);

            int result\_sub = sub(i, j);

            out.println("Result\_add = " + result\_add);

            out.println("Result\_sub = " + result\_sub);

            }

            catch(Exception ex) {

            out.println("Exception: " + ex);

            }

            out.println("</body>");

            out.println("</html>");

        }

    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

    /\*\*

     \* Handles the HTTP <code>GET</code> method.

     \*

     \* @param request servlet request

     \* @param response servlet response

     \* @throws ServletException if a servlet-specific error occurs

     \* @throws IOException if an I/O error occurs

     \*/

    @Override

    protected void doGet(HttpServletRequest request, HttpServletResponse response)

            throws ServletException, IOException {

        processRequest(request, response);

    }

    /\*\*

     \* Handles the HTTP <code>POST</code> method.

     \*

     \* @param request servlet request

     \* @param response servlet response

     \* @throws ServletException if a servlet-specific error occurs

     \* @throws IOException if an I/O error occurs

     \*/

    @Override

    protected void doPost(HttpServletRequest request, HttpServletResponse response)

            throws ServletException, IOException {

        processRequest(request, response);

    }

    /\*\*

     \* Returns a short description of the servlet.

     \*

     \* @return a String containing servlet description

     \*/

    @Override

    public String getServletInfo() {

        return "Short description";

    }// </editor-fold>

    private int add(int i, int j) {

        org.me.calculator.CalculatorWSApplication port = service.getCalculatorWSApplicationPort();

        return port.add(i, j);

    }

    private int sub(int i, int j) {

        org.me.calculator.CalculatorWSApplication port = service.getCalculatorWSApplicationPort();

        return port.sub(i, j);

    }

}

### index.jsp file Souce Code:

<%--

    Document   : index

    Created on : Sep 4, 2021, 9:35:47 PM

    Author     : yashchandra05

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

        <title>JSP Page</title>

    </head>

    <body>

        <h1>Hello World!</h1>

            <%-- start web service invocation --%><hr/>

    <%

    try {

    org.me.calculator.CalculatorWSApplication\_Service service = new org.me.calculator.CalculatorWSApplication\_Service();

    org.me.calculator.CalculatorWSApplication port = service.getCalculatorWSApplicationPort();

    int i = 3;

    int j = 4;

    int result\_add = port.add(i, j);

    out.println("Result\_add = "+result\_add);

    } catch (Exception ex) {

        out.println("Exception" + ex);

    }

    %>

    <%-- end web service invocation --%><hr/>

        <%-- start web service invocation --%><hr/>

    <%

    try {

    org.me.calculator.CalculatorWSApplication\_Service service = new org.me.calculator.CalculatorWSApplication\_Service();

    org.me.calculator.CalculatorWSApplication port = service.getCalculatorWSApplicationPort();

    int i = 3;

    int j = 4;

    int result\_sub = port.sub(i, j);

    out.println("Result\_sub = "+result\_sub);

    } catch (Exception ex) {

        out.println("Exception" + ex);

    }

    %>

    <%-- end web service invocation --%><hr/>

    </body>

</html>