**PRACTICAL: 8**

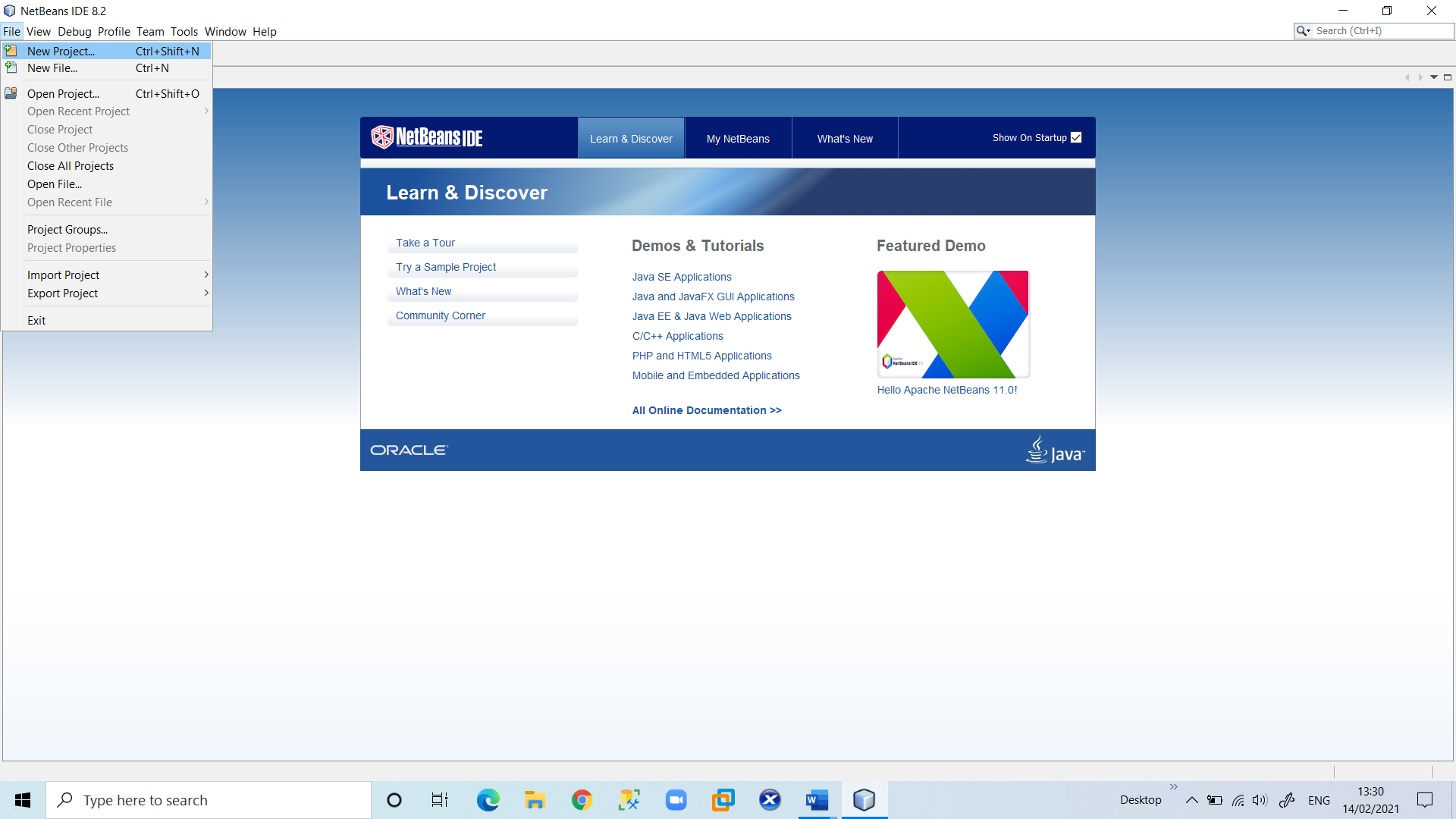
**IMPLEMENTING “BIG” WEB SERVICE**

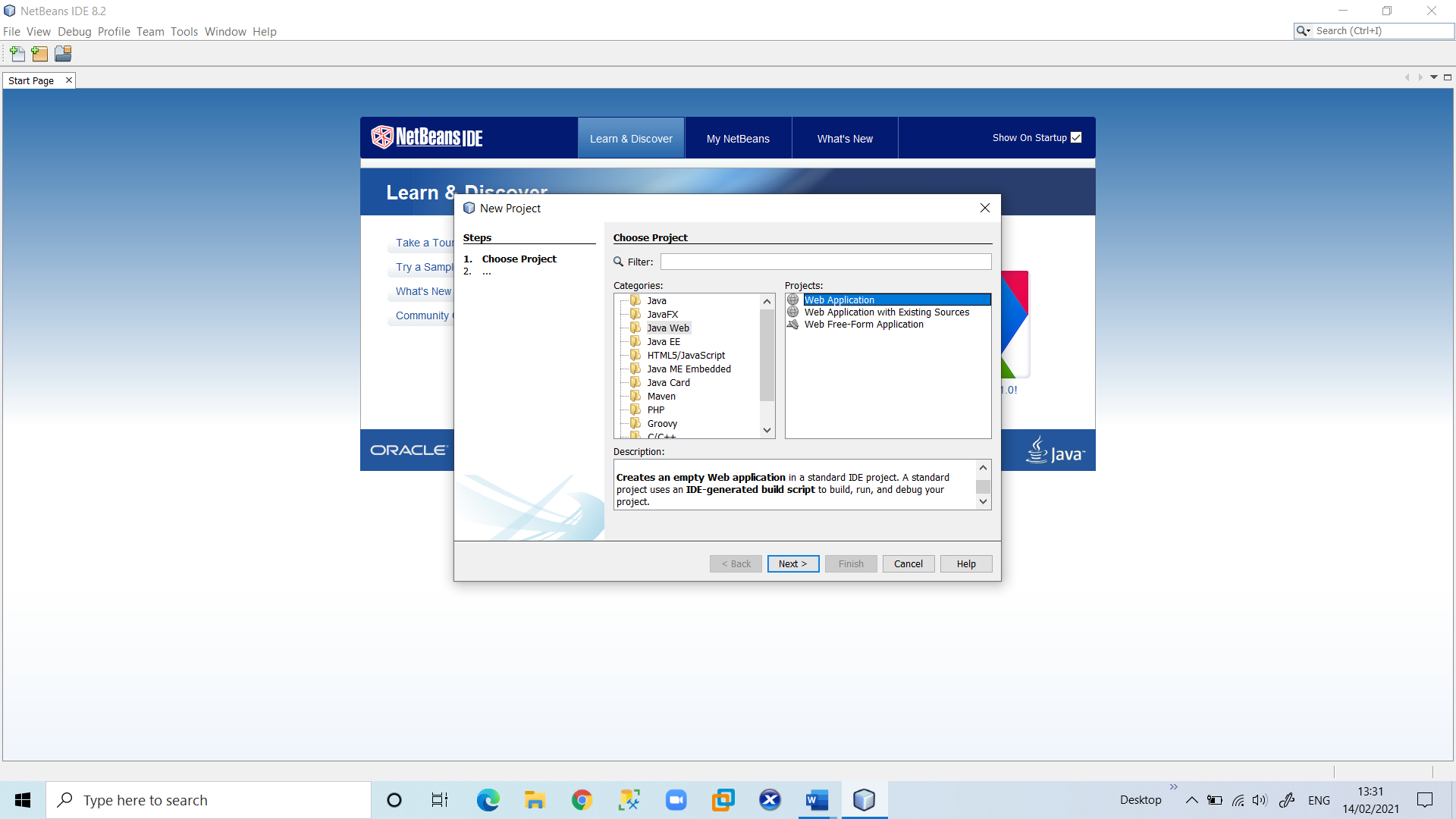
**Steps:**

**1) Creating a Web Service**

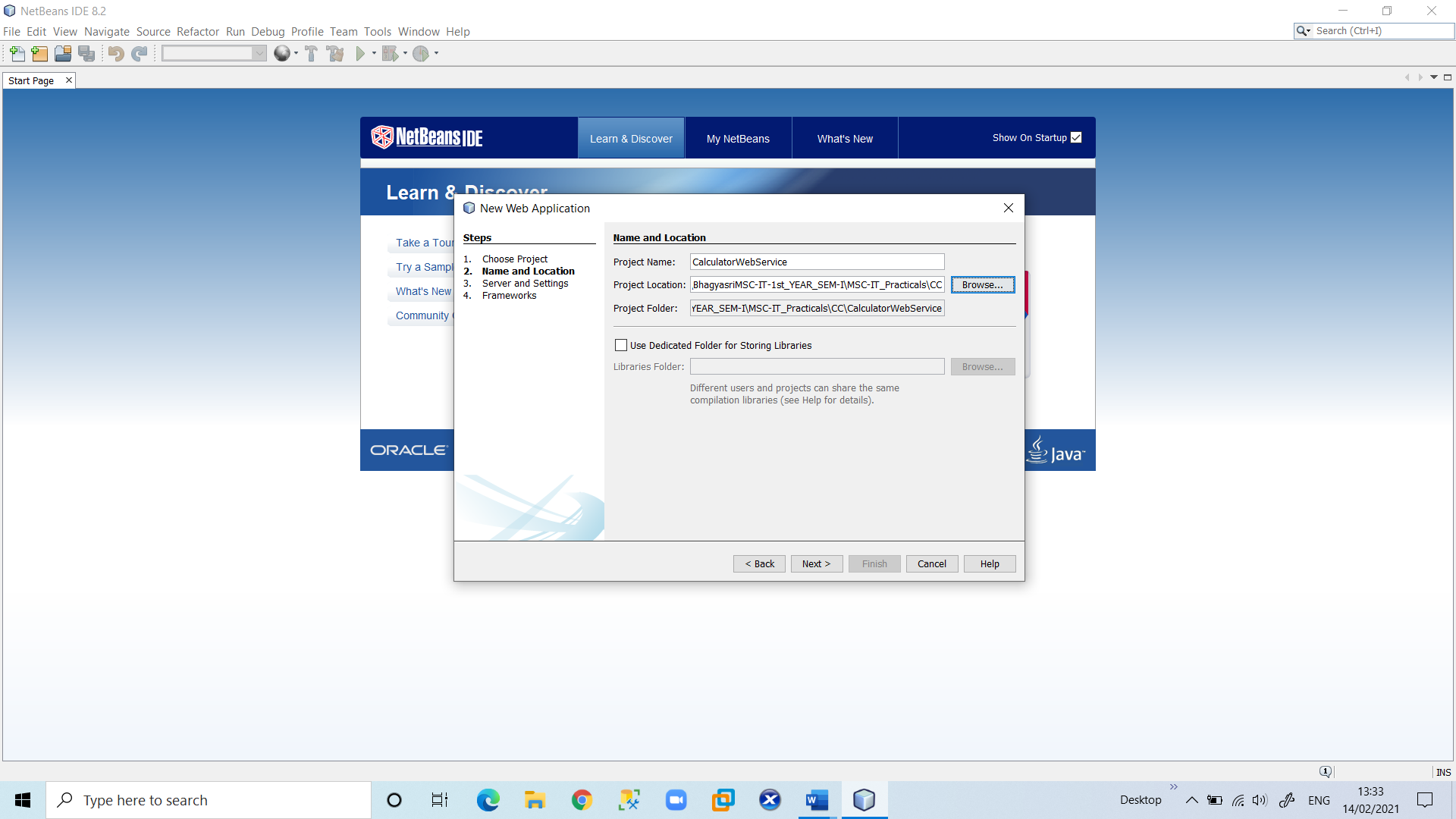
A. Choosing a Container: Open NetBeans IDE 8.2

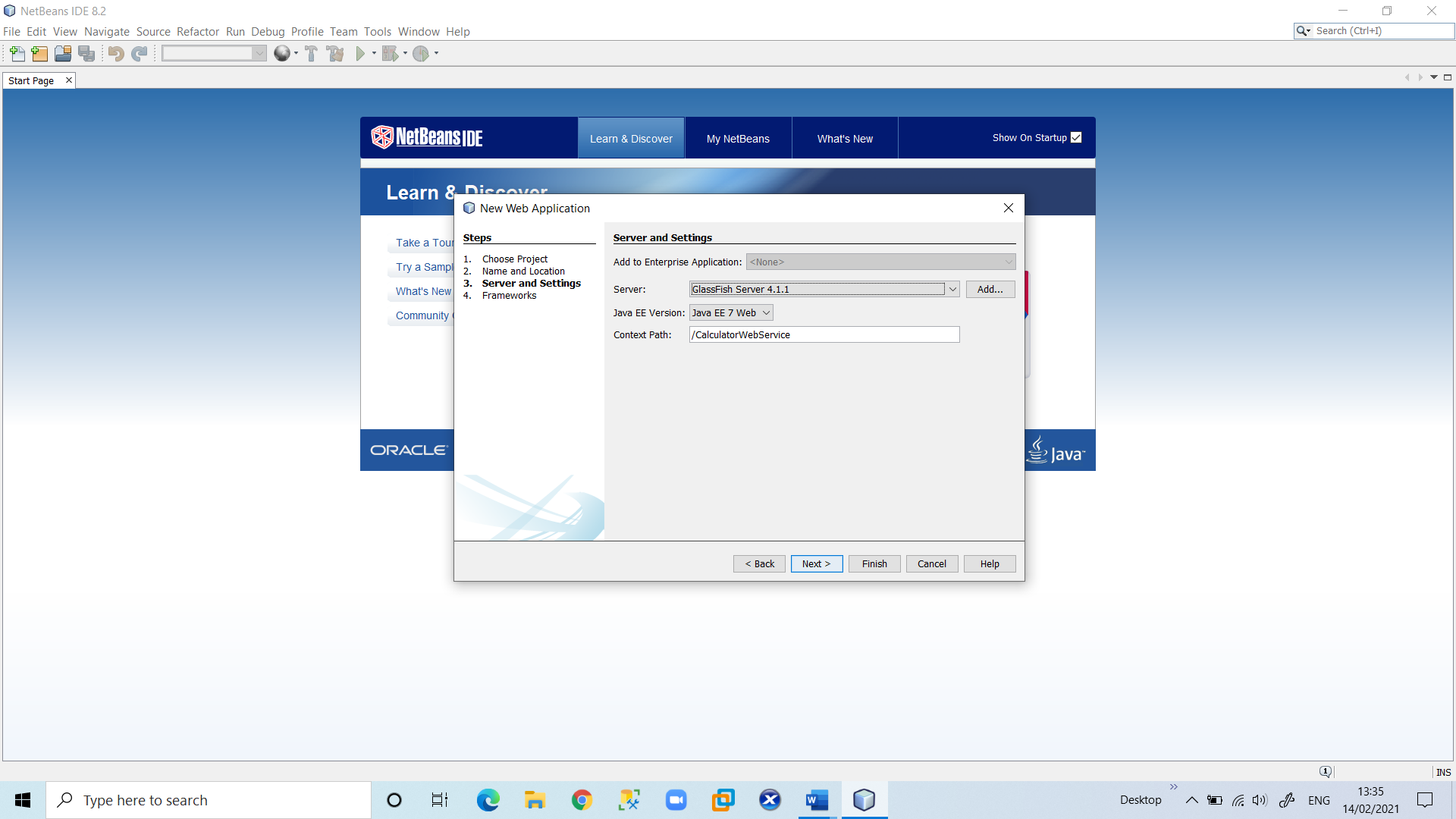
1. Choose **File > New Project**. Select **Web Application** from the Java Web.

2. Name the project “**CalculatorWebService**”. Select a location for the project. Click **Next**.



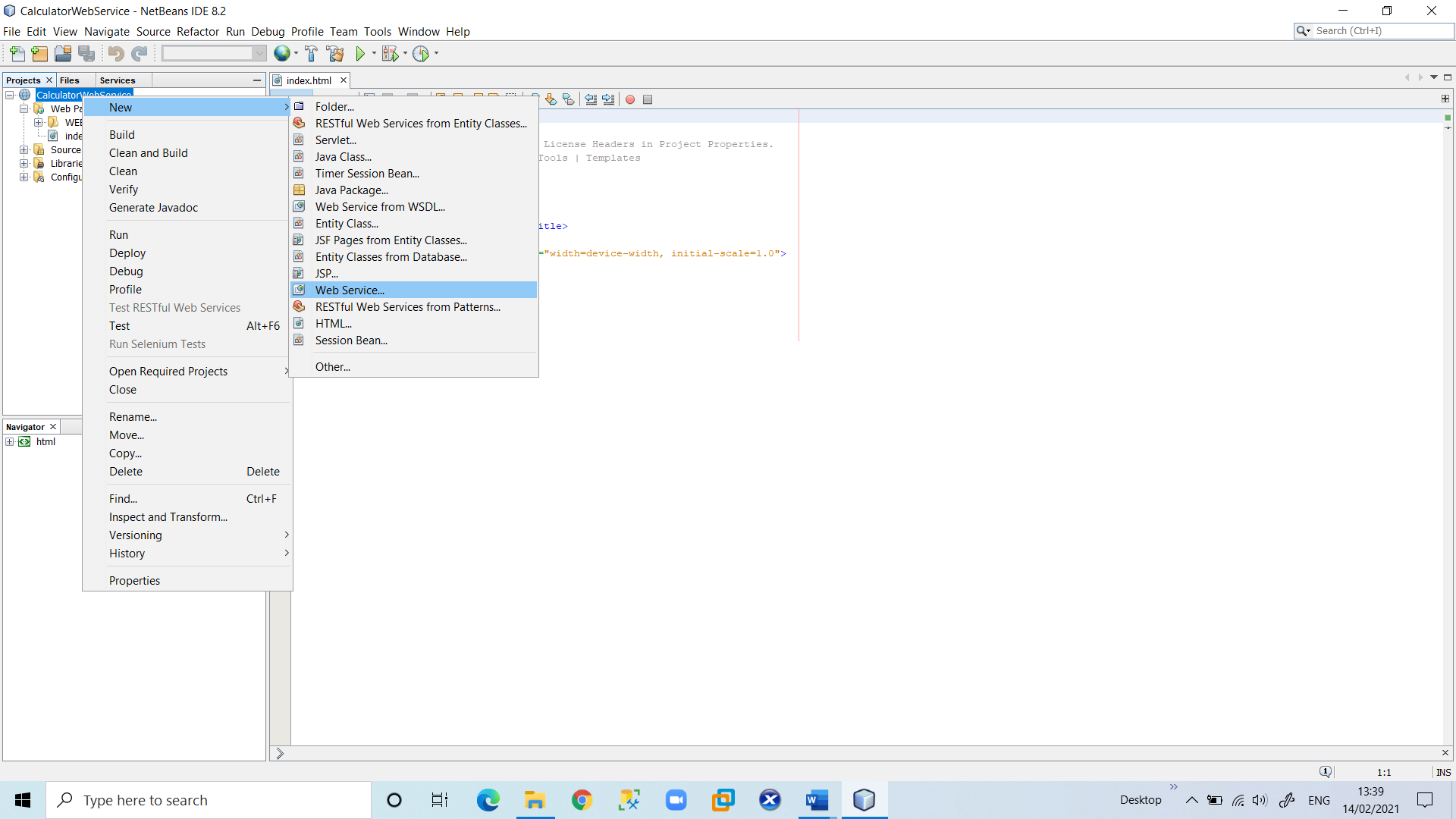
2. Name the project “**CalculatorWebService**”. Select a location for the project. Click **Next**.

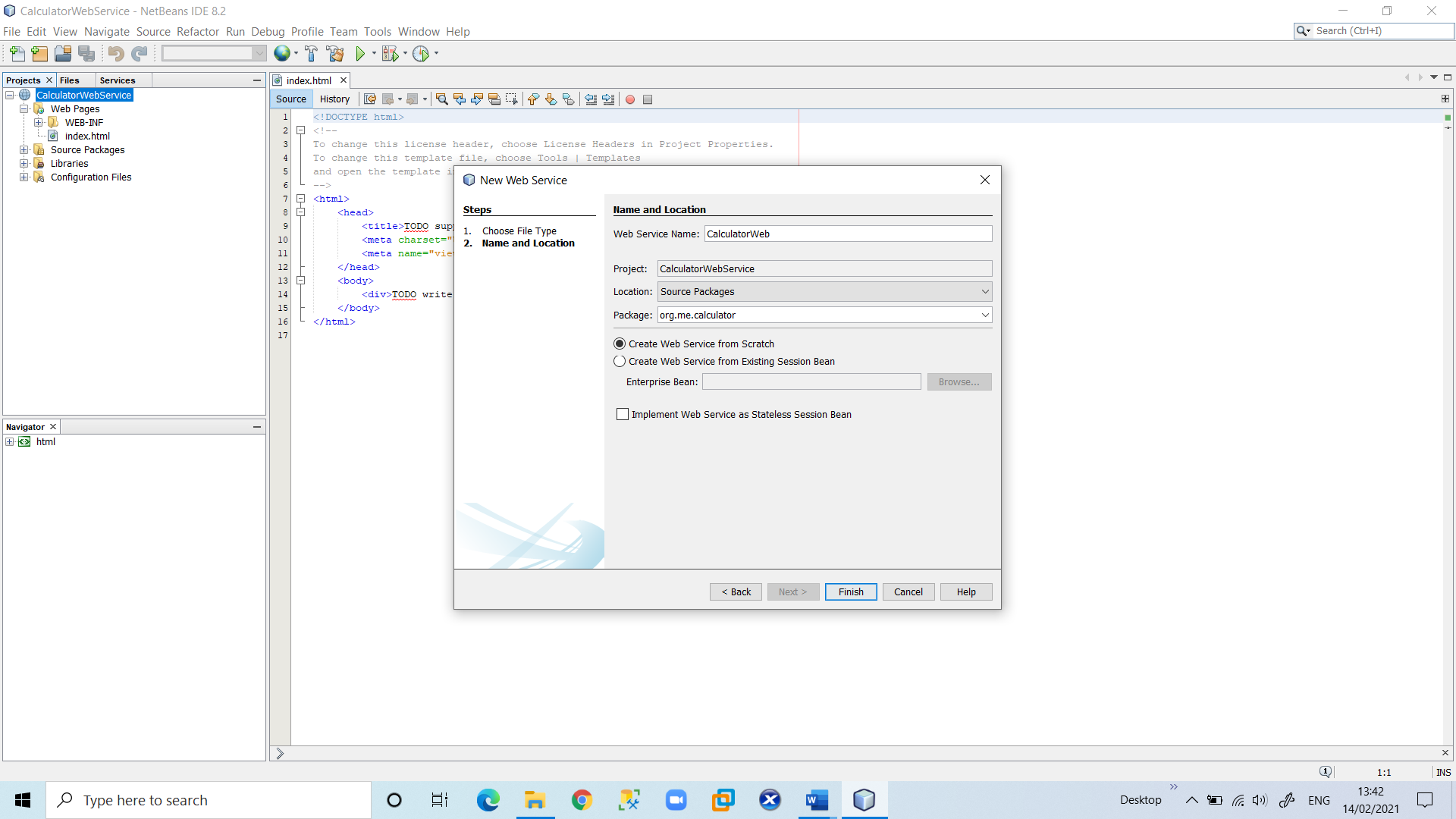


3. Select your server and Java EE version and click Finish. (Do not do anything in Framework)

B. Creating a Web Service from a Java Class

1. Right-click the “**CalculatorWebService**” node and choose **New > Web Service**.

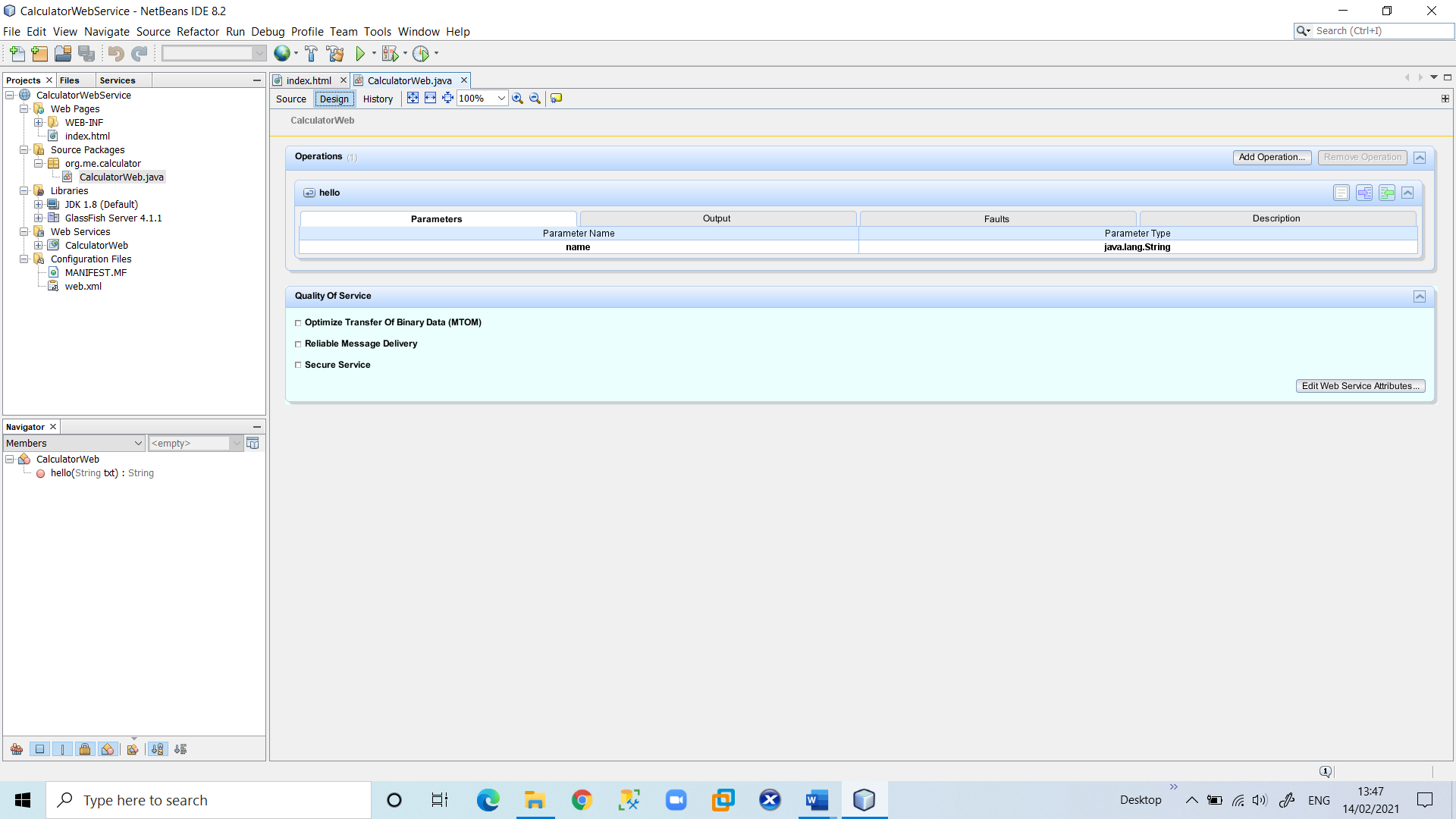


2. Name the web service “**CalculatorWeb**” and type **org.me.calculator** in Package. Leave Create Web Service from Scratch selected. 

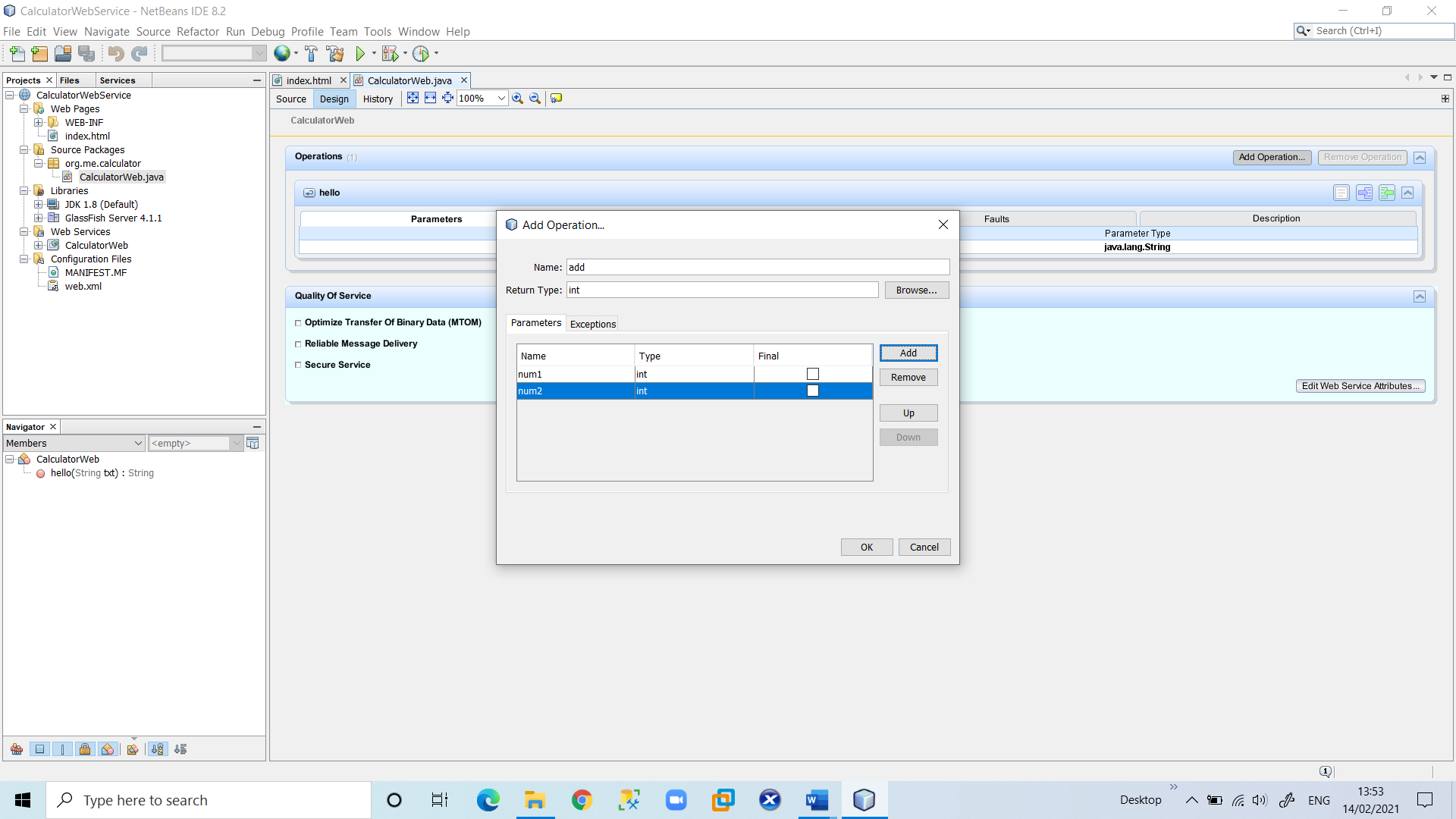
Click **Finish**. The Projects window displays the structure of the new web service and the source code is shown in the editor area.

**2) Adding an Operation to the Web Service: The goal of this exercise is to add to the web service an operation that adds two numbers received from a client. The NetBeans IDE provides a dialog for adding an operation to a web service. You can open this dialog either in the web service visual designer or in the web service context menu.**

A. To add an operation to the web service:

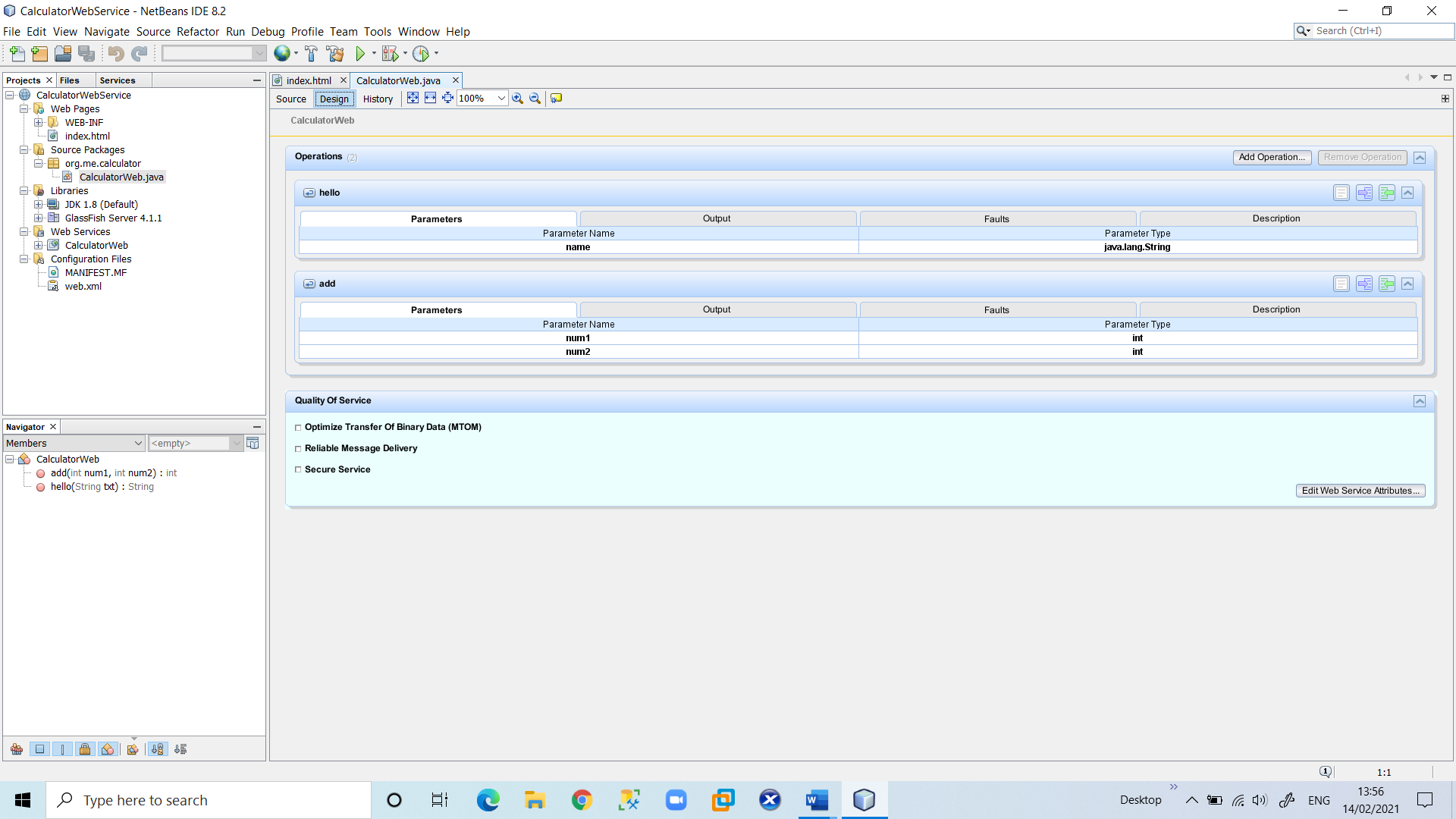
1. Change to the Design view in the editor.

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

In the upper part of the Add Operation dialog box, type **add** in Name and type **int** in the Return Type drop-down list. In the lower part of the Add Operation dialog box, click **Add** and create a parameter of type int named **num1**. Click Add again and create a parameter of type int called **num2**. You now see the following:

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

The visual designer now displays the following:



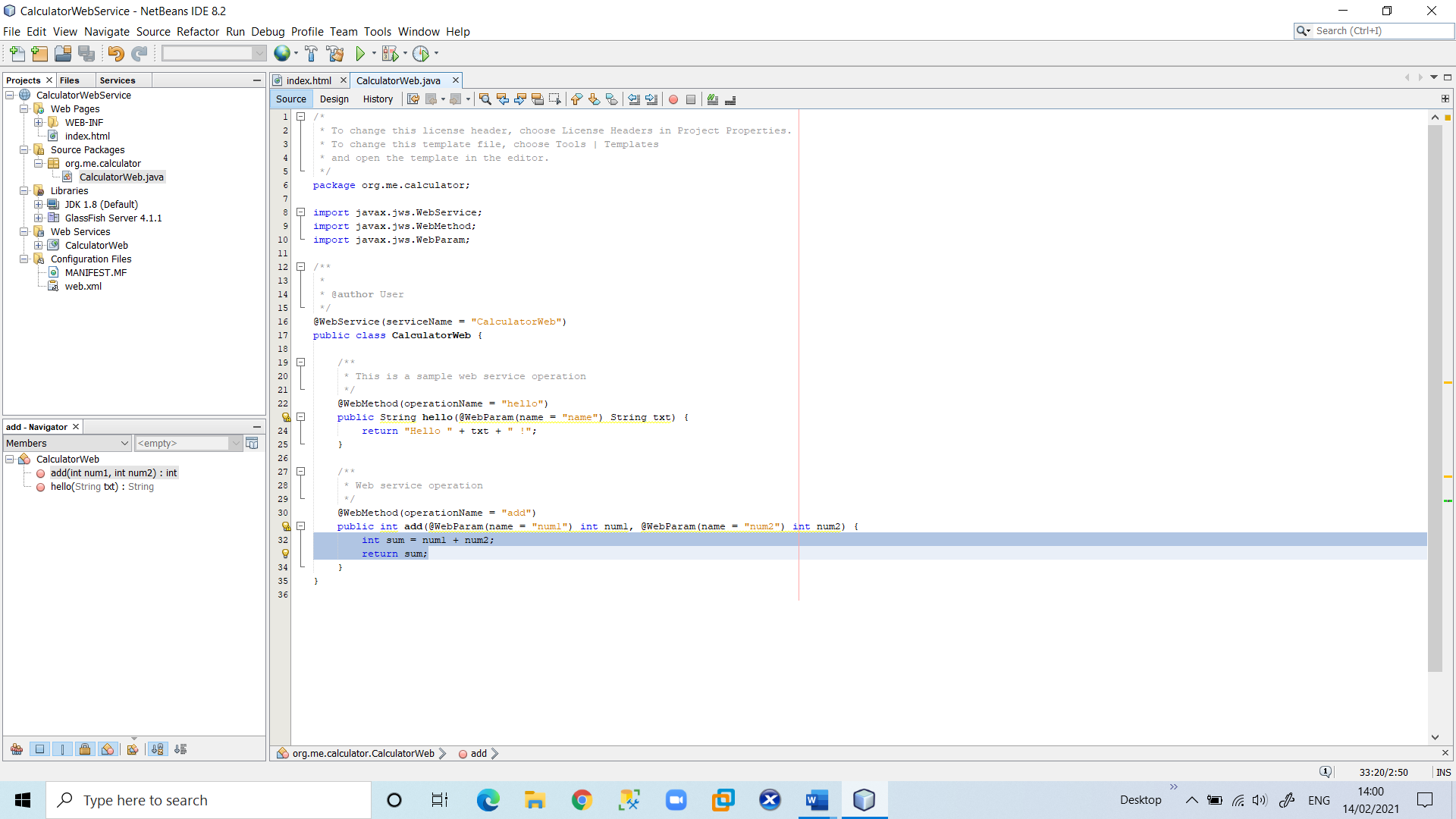
4. Click Source. And code the following.

@WebMethod(operationName = "add")

public int add(@WebParam(name = "num1") int num1, @WebParam(name = " num2") int num2) {

int sum = num1 + num2;

return sum; }

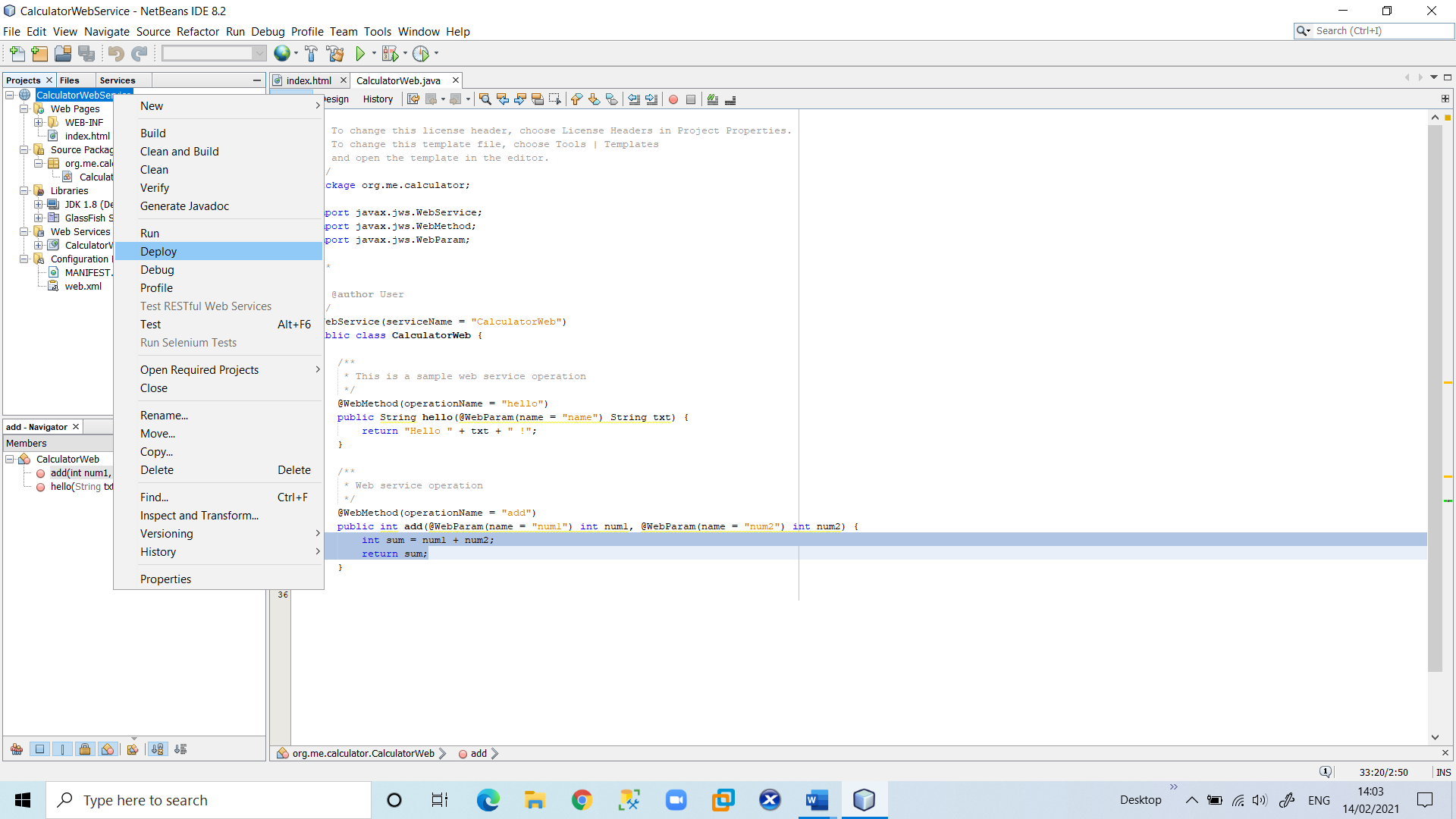


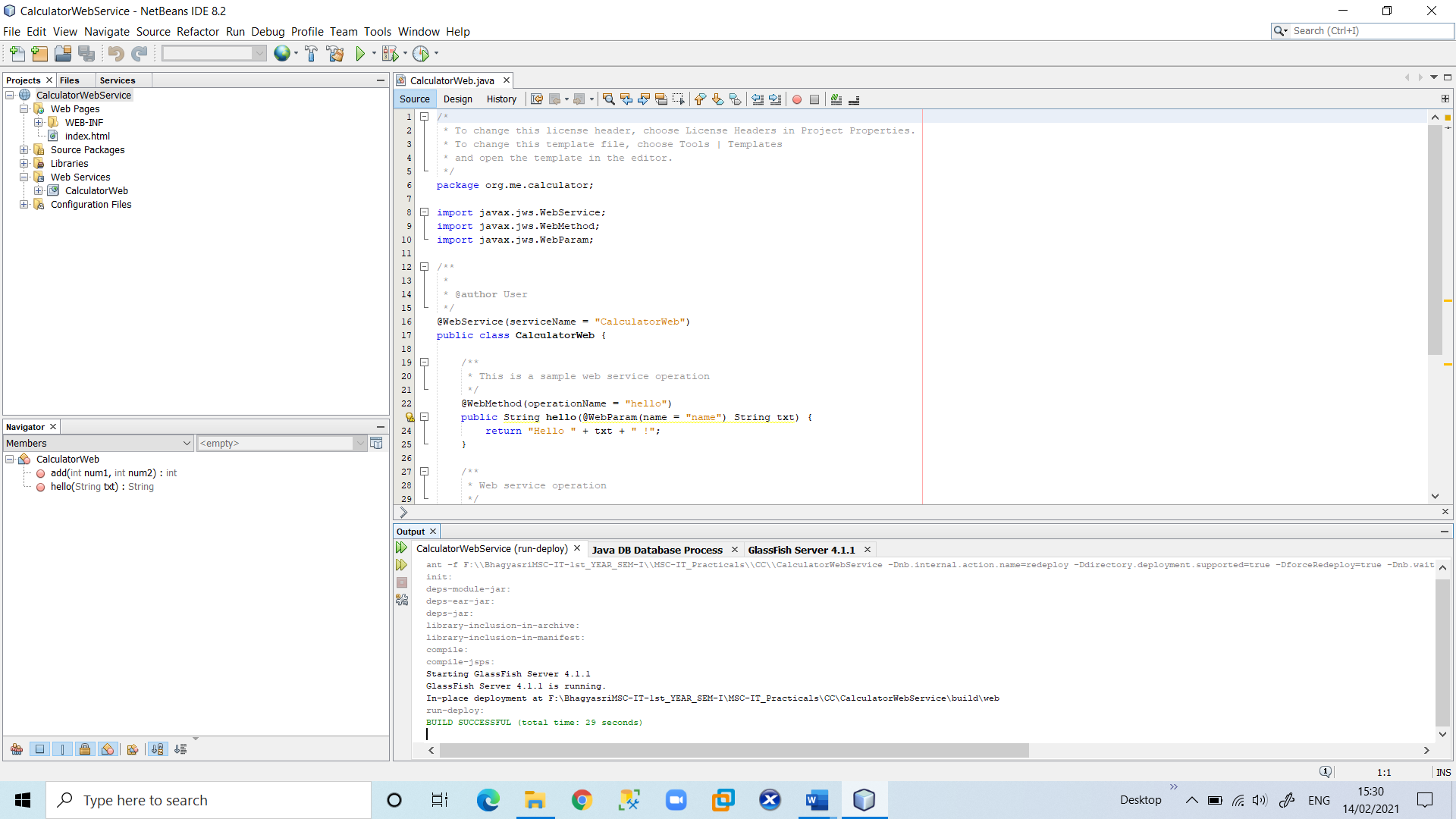
**3) Deploying and Testing the Web Service**

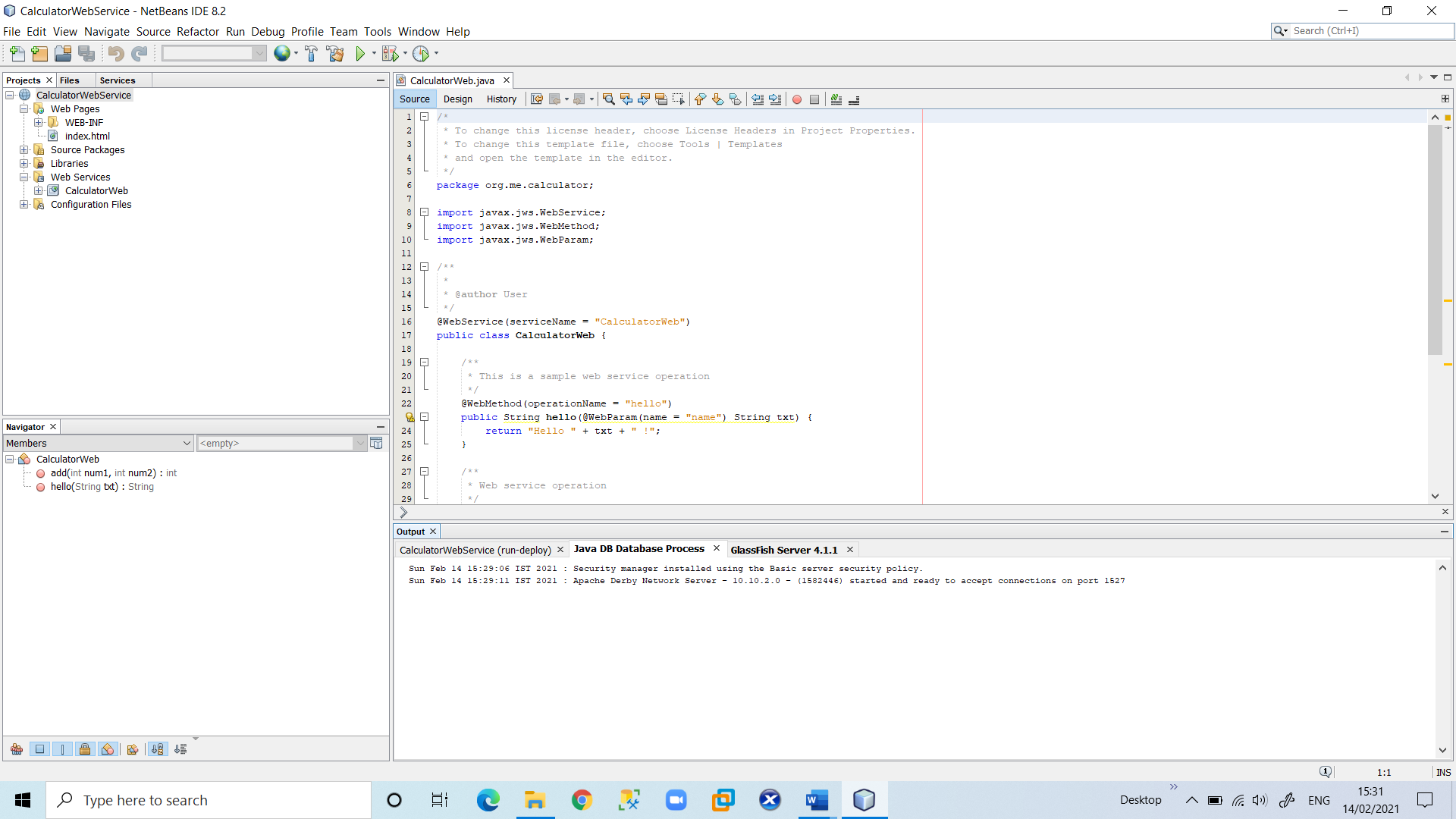
After you deploy a web service to a server, you can use the IDE to open the server's test client, if the server has a test client. The GlassFish and WebLogic servers provide test clients.

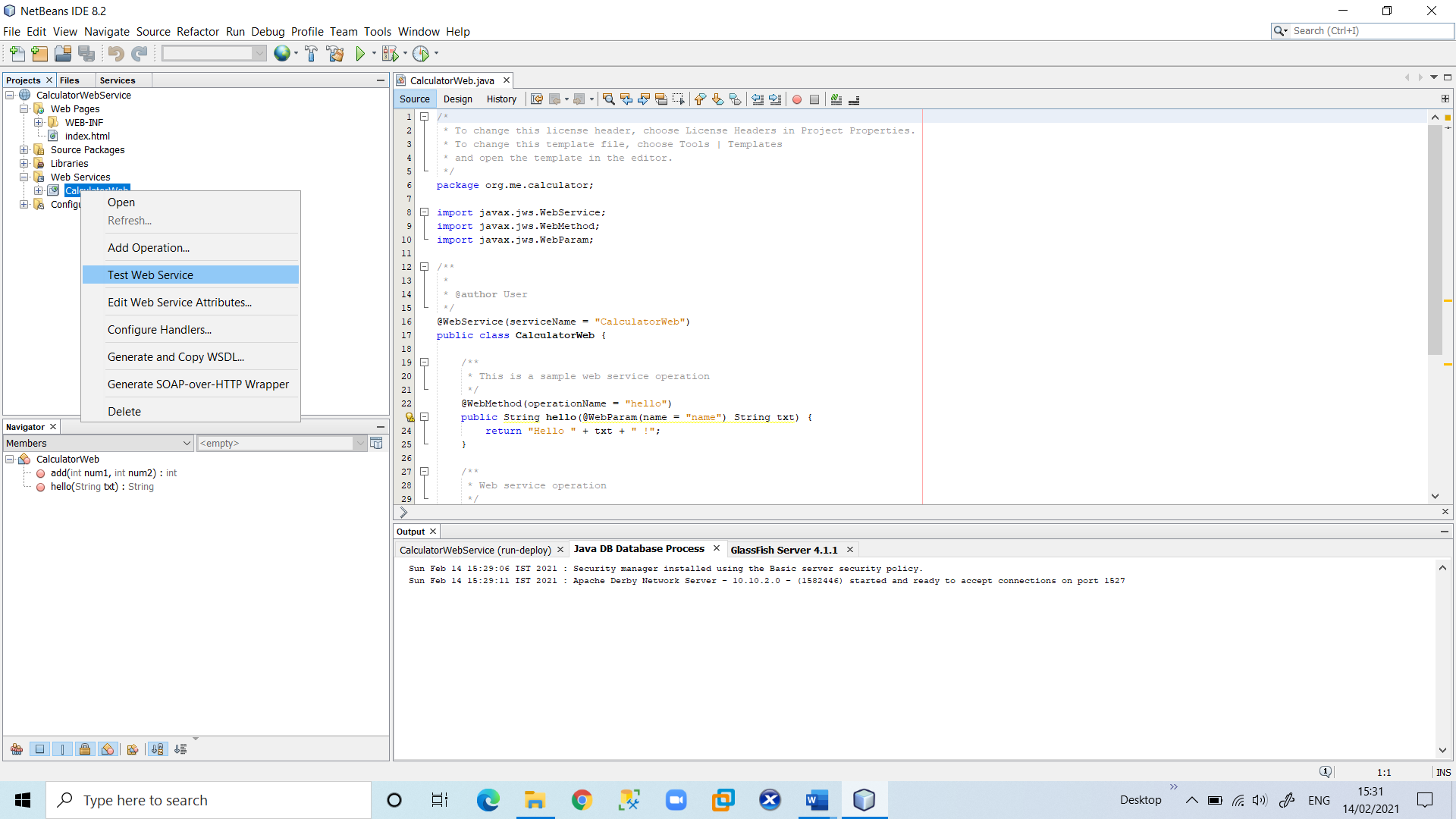
A. To test successful deployment to a GlassFish or WebLogic server:

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



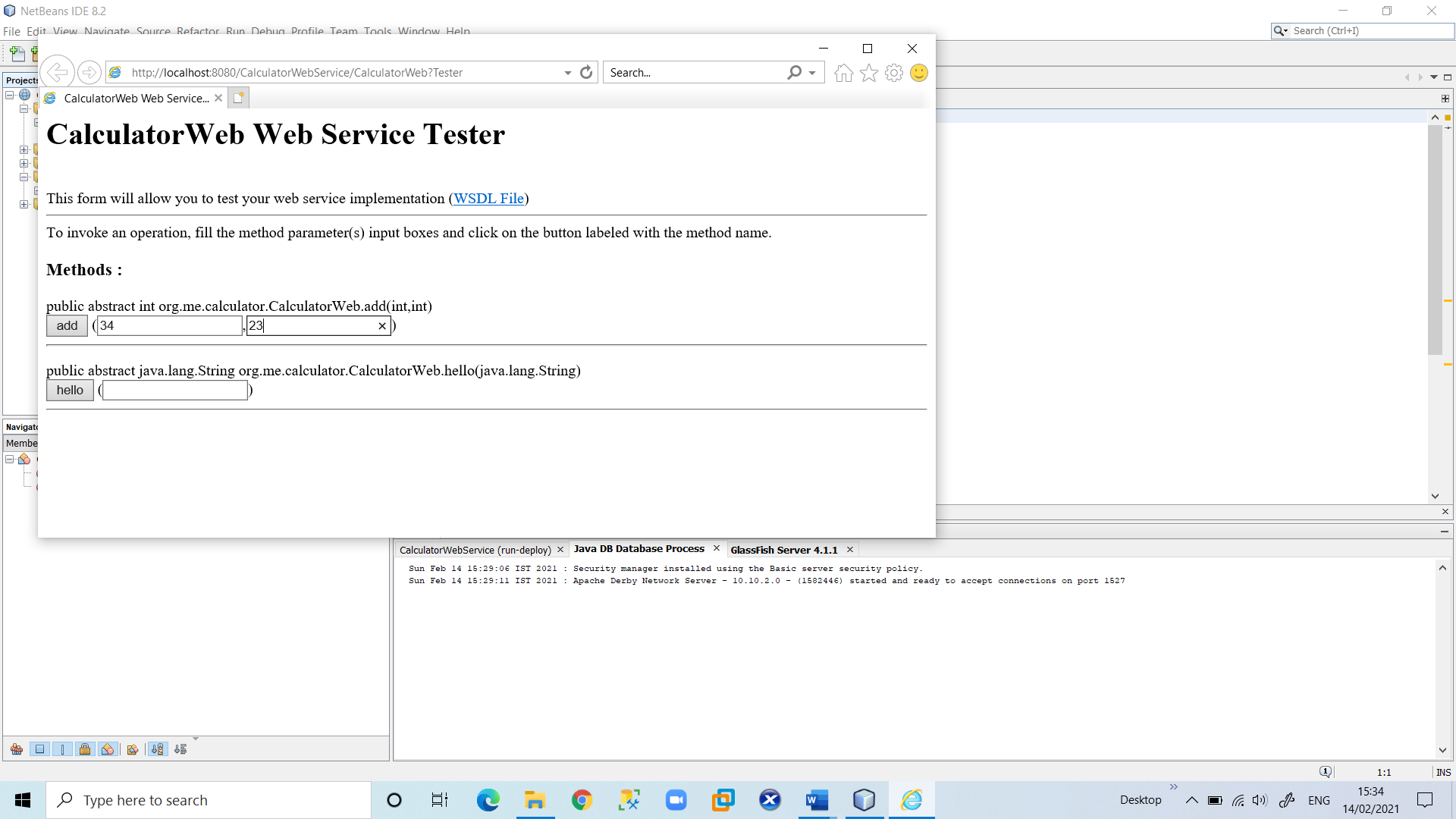




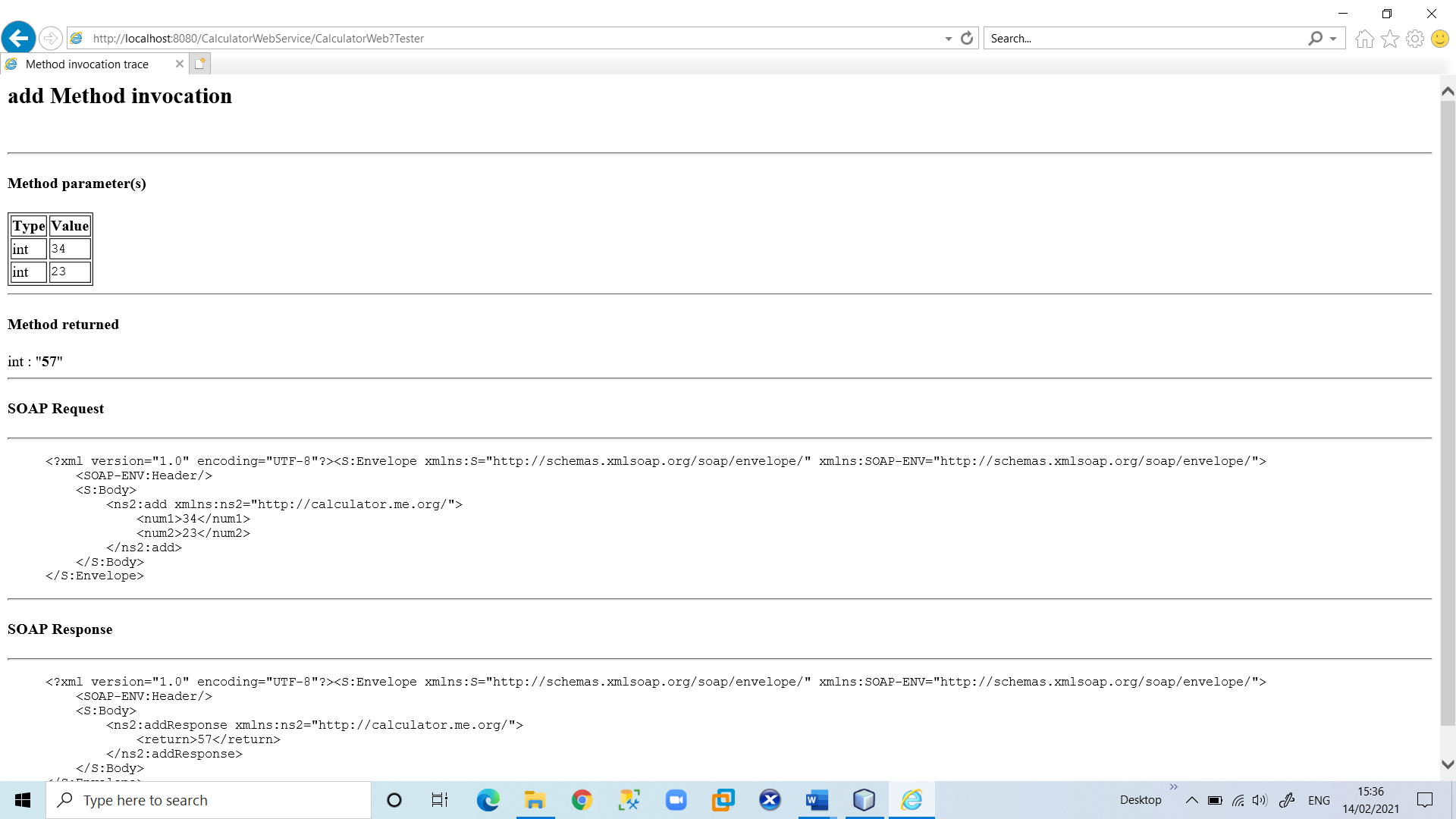
2. In the IDE's Projects tab, expand the Web Services node of the “**CalculatorWebService**” project. Right-click the “**CalculatorWeb**” node, and choose **Test Web Service**.

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

3. If you deployed to the GlassFish server, type two numbers in the tester page, as shown below and press enter for output.



4. The sum of the two numbers is displayed:



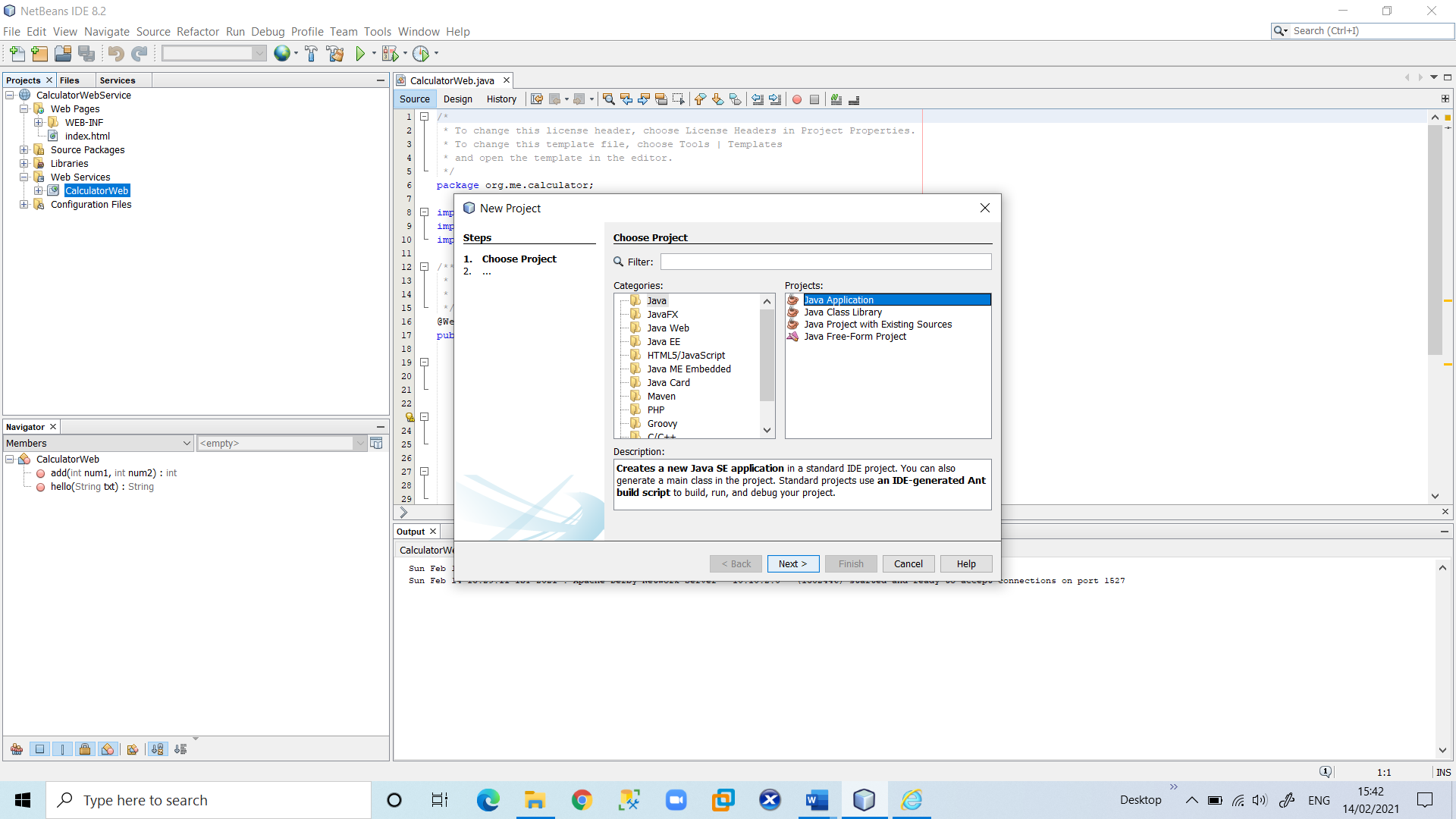
**4) Consuming the Web Service**

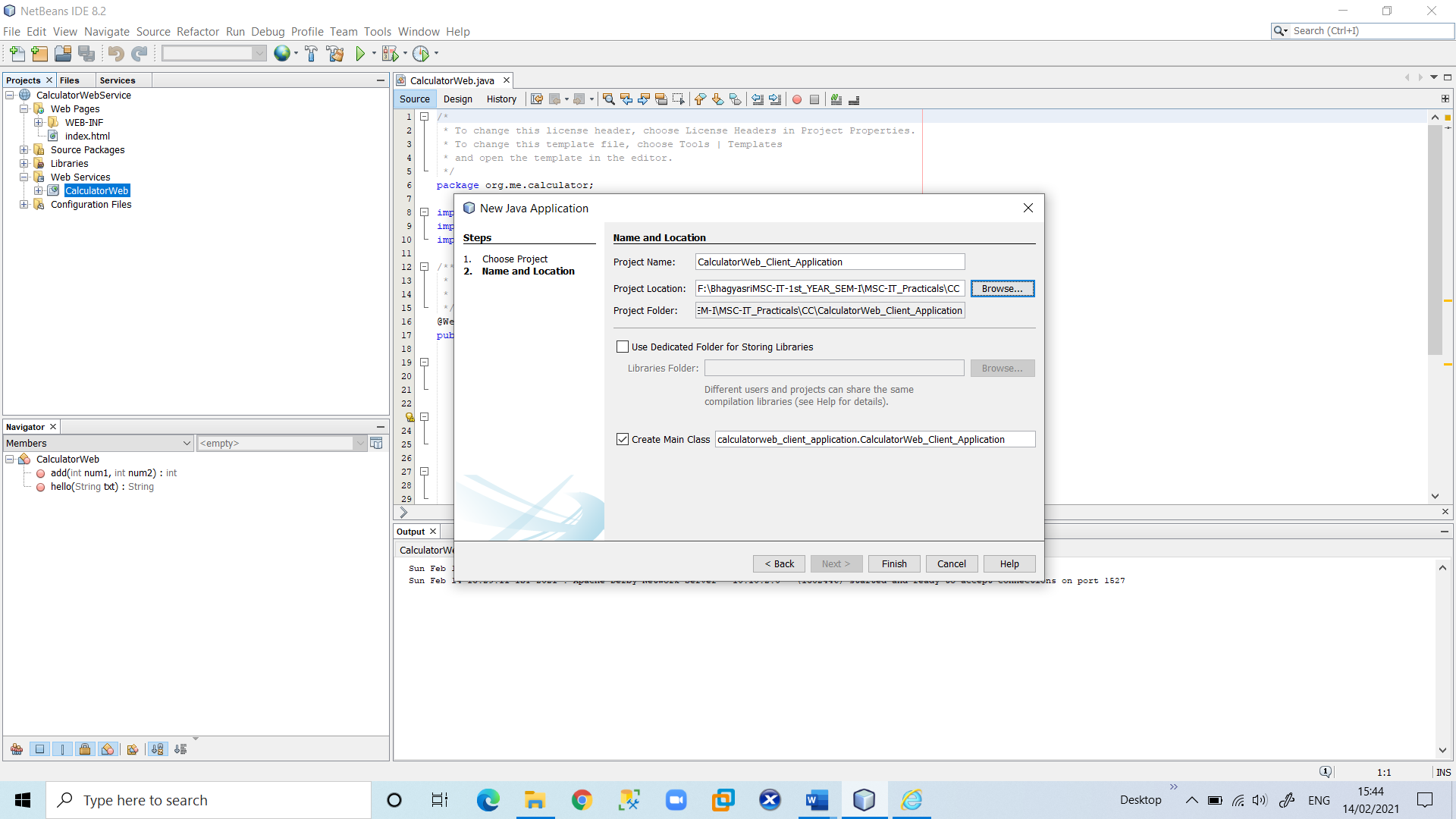
Now that you have deployed the web service, you need to create a client to make use of the web service's add method.

**1.Client: Java Class in Java SE Application**

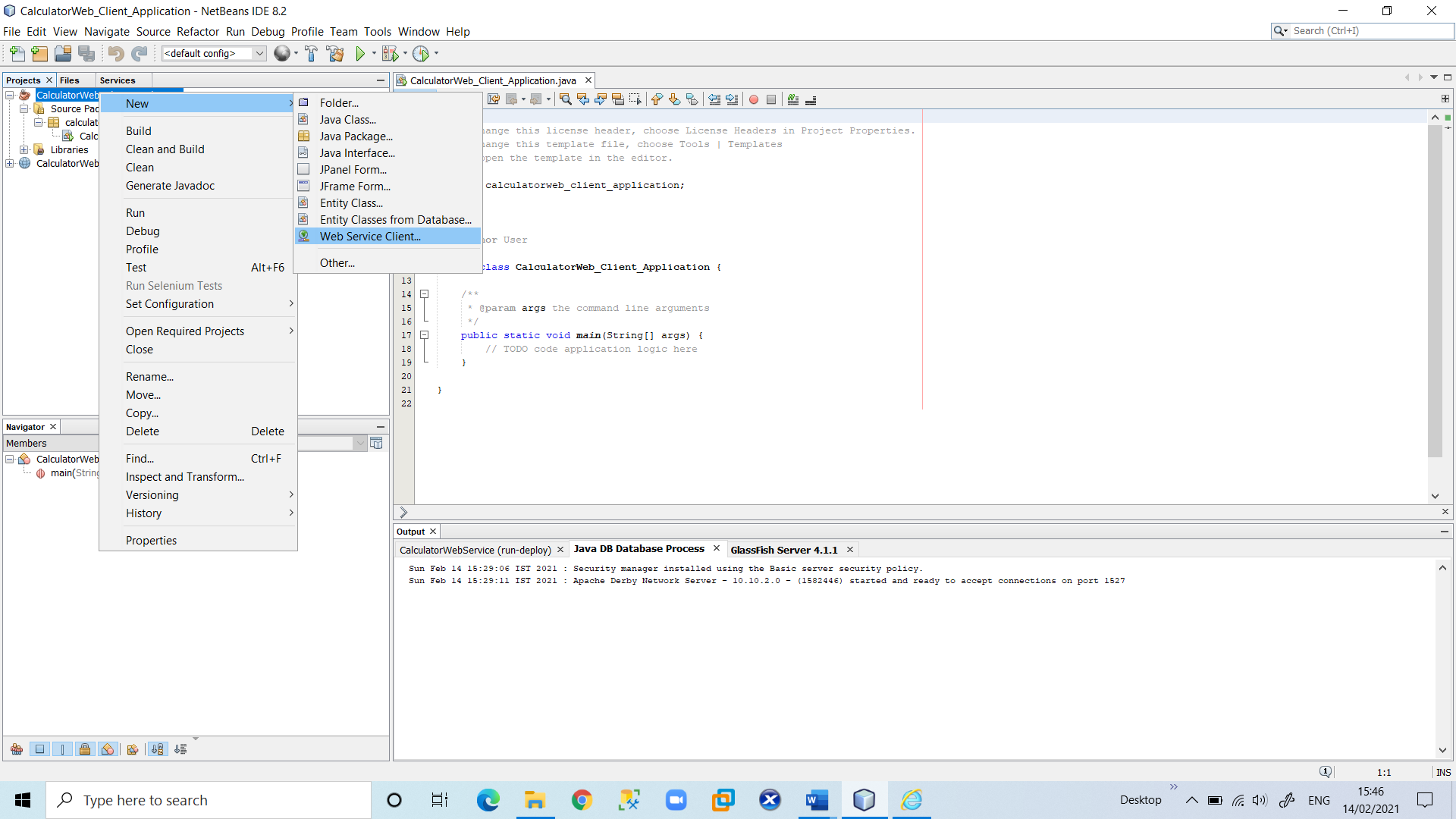
1. Choose **File > New Project**. Select Java Application from the Java category.

Name the project “**CalculatorWeb\_Client\_Application**”. Leave Create Main Class selected and accept all other default settings. Click **Finish**.

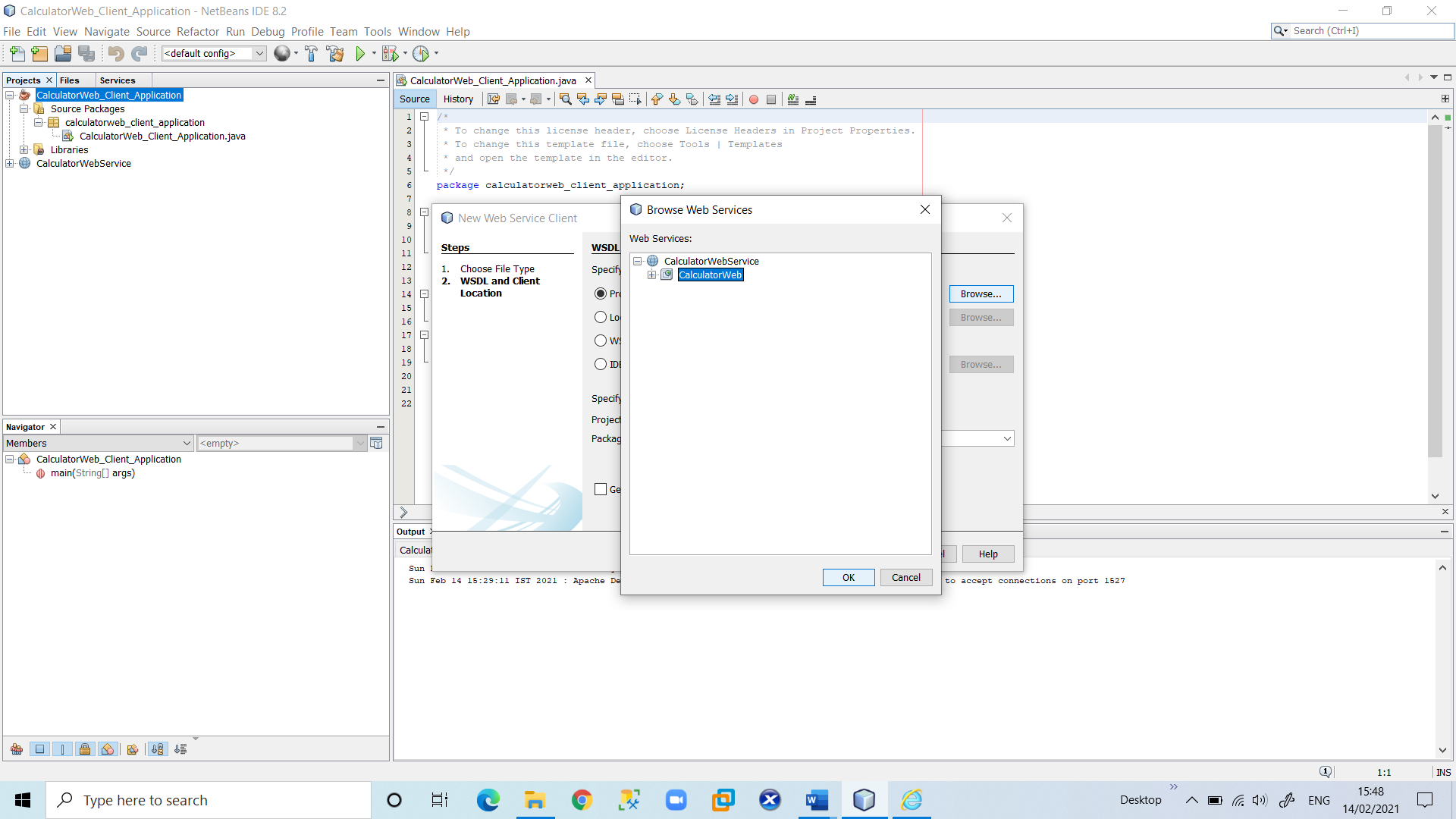




2. Right-click the “CalculatorWeb\_Client\_Application” node and choose New > Web Service Client. The New Web Service Client wizard opens.



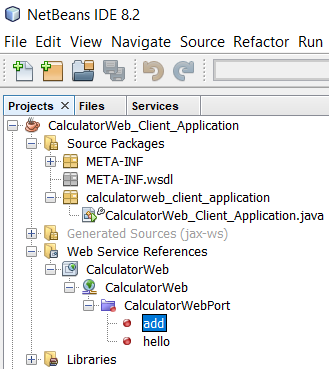
3. Select Project as the **WSDL source**. Click **Browse**. Browse to the “**CalculatorWeb**” web service in the “**CalculatorWebService**” project. When you have selected the web service, click **OK**.



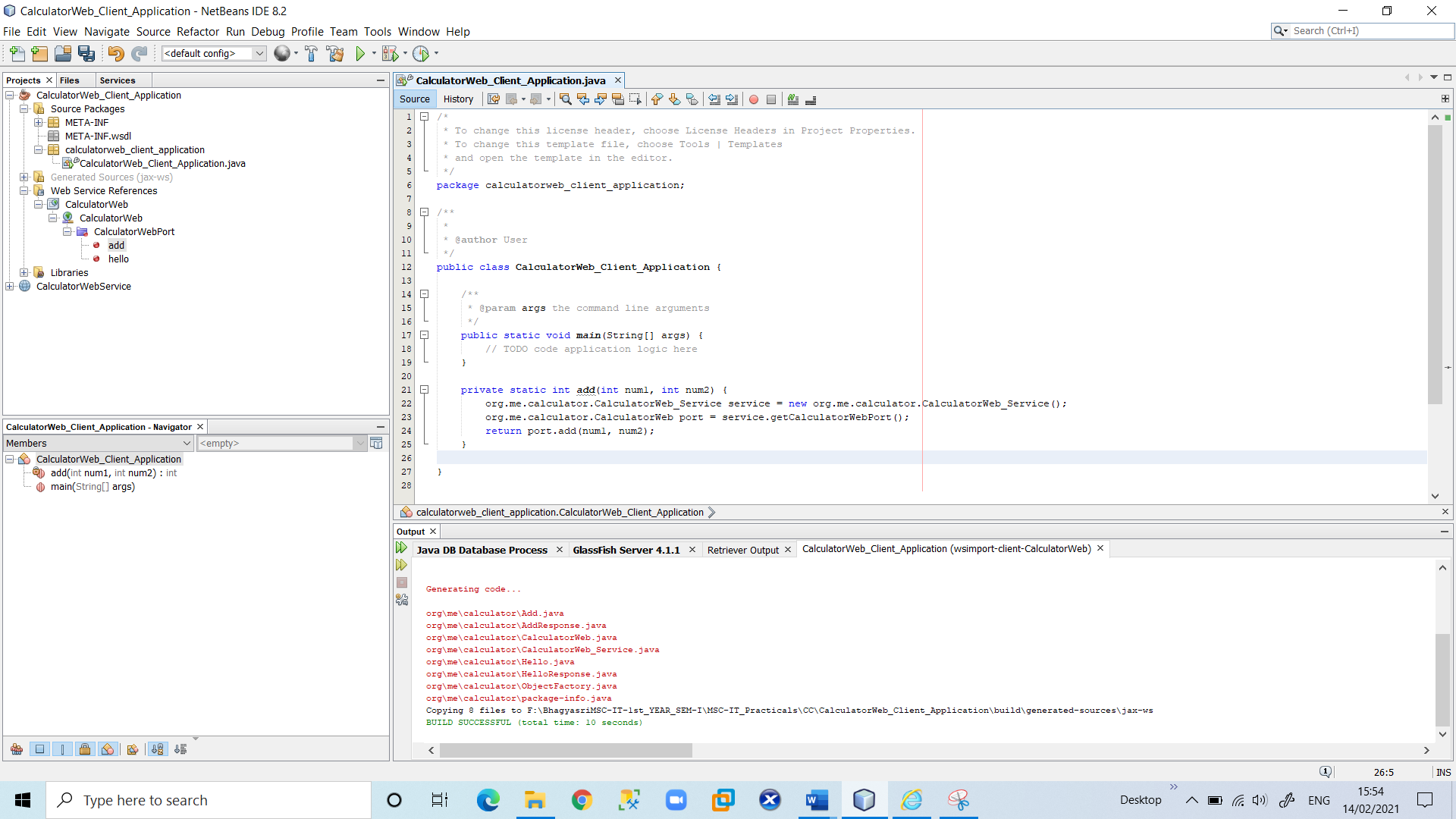
4. Do not select a package name. Leave this field empty. Leave the other settings at default and click Finish.



5. The Projects window displays the new web service client, with a node for the add method that you created:



6. Double-click your main class so that it opens in the Source Editor. Drag the add node below the main() method.



7. In the main() method body, replace the TODO comment with code that initializes values for num1 and num2, calls add(), and prints the result.

try

{

int num1 = 33;

int num2 = 42;

int result = add(num1, num2);

System.out.println("Result of Add Operation = " + result);

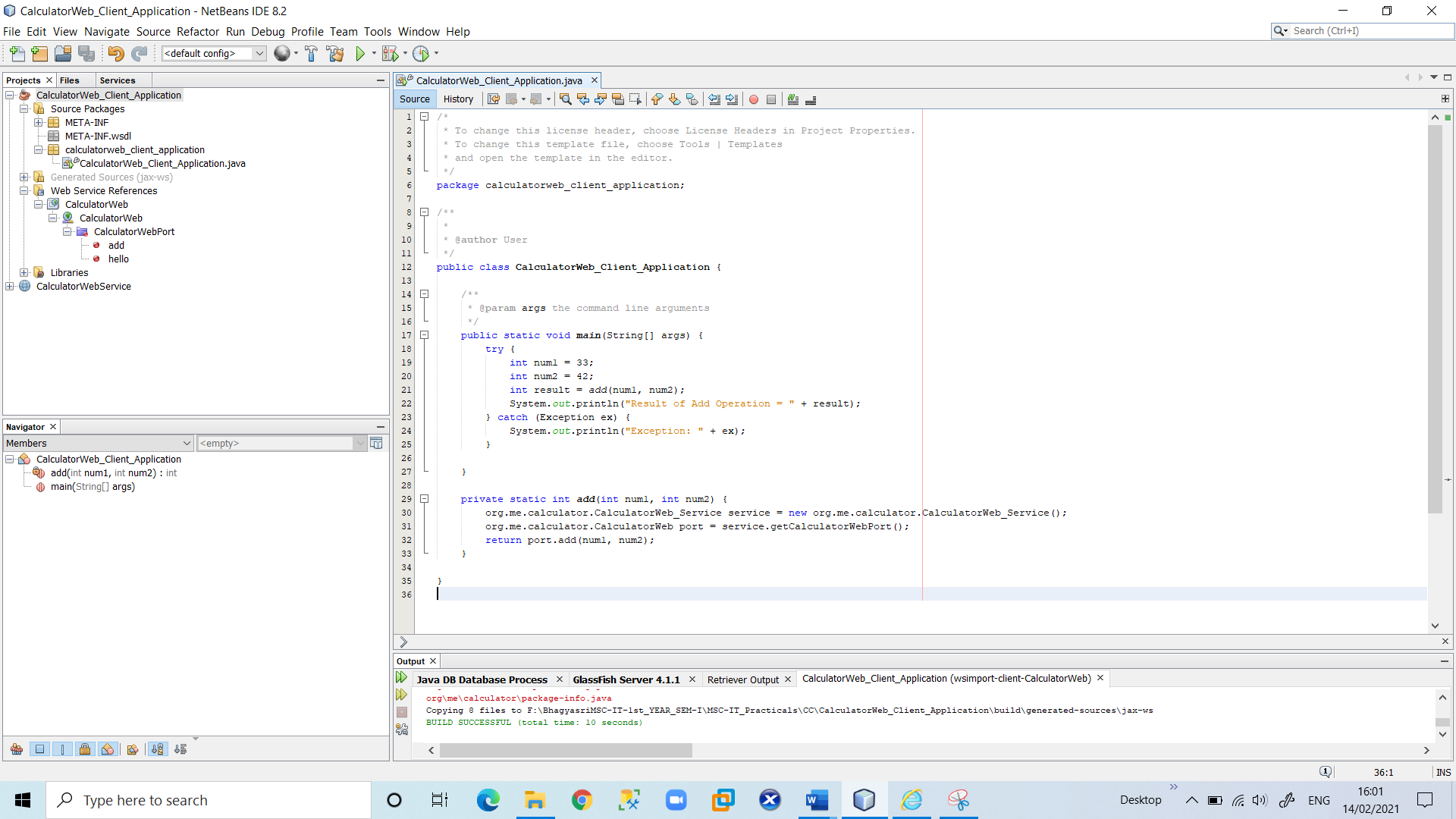
}

catch (Exception ex)

{

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

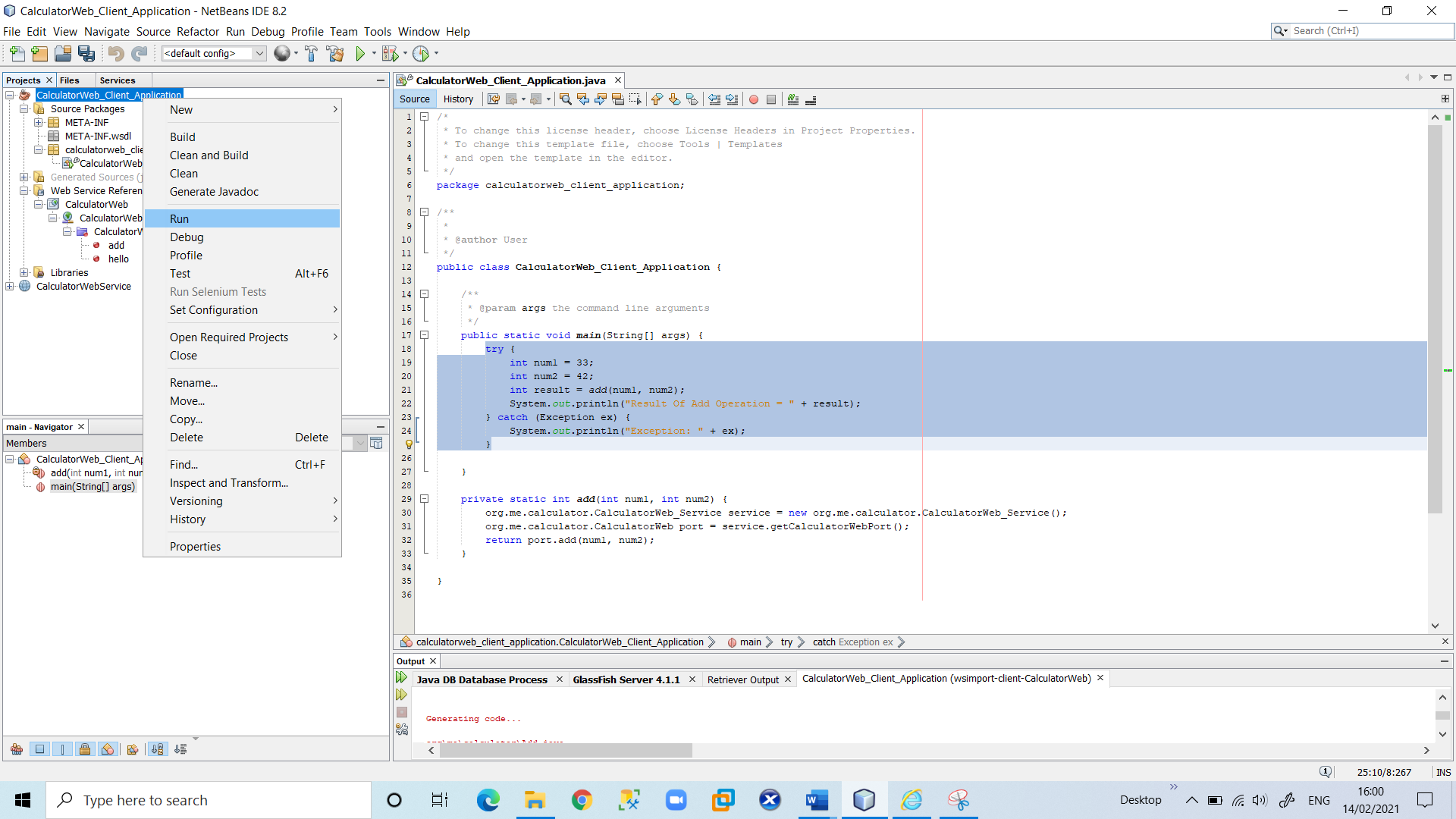
}

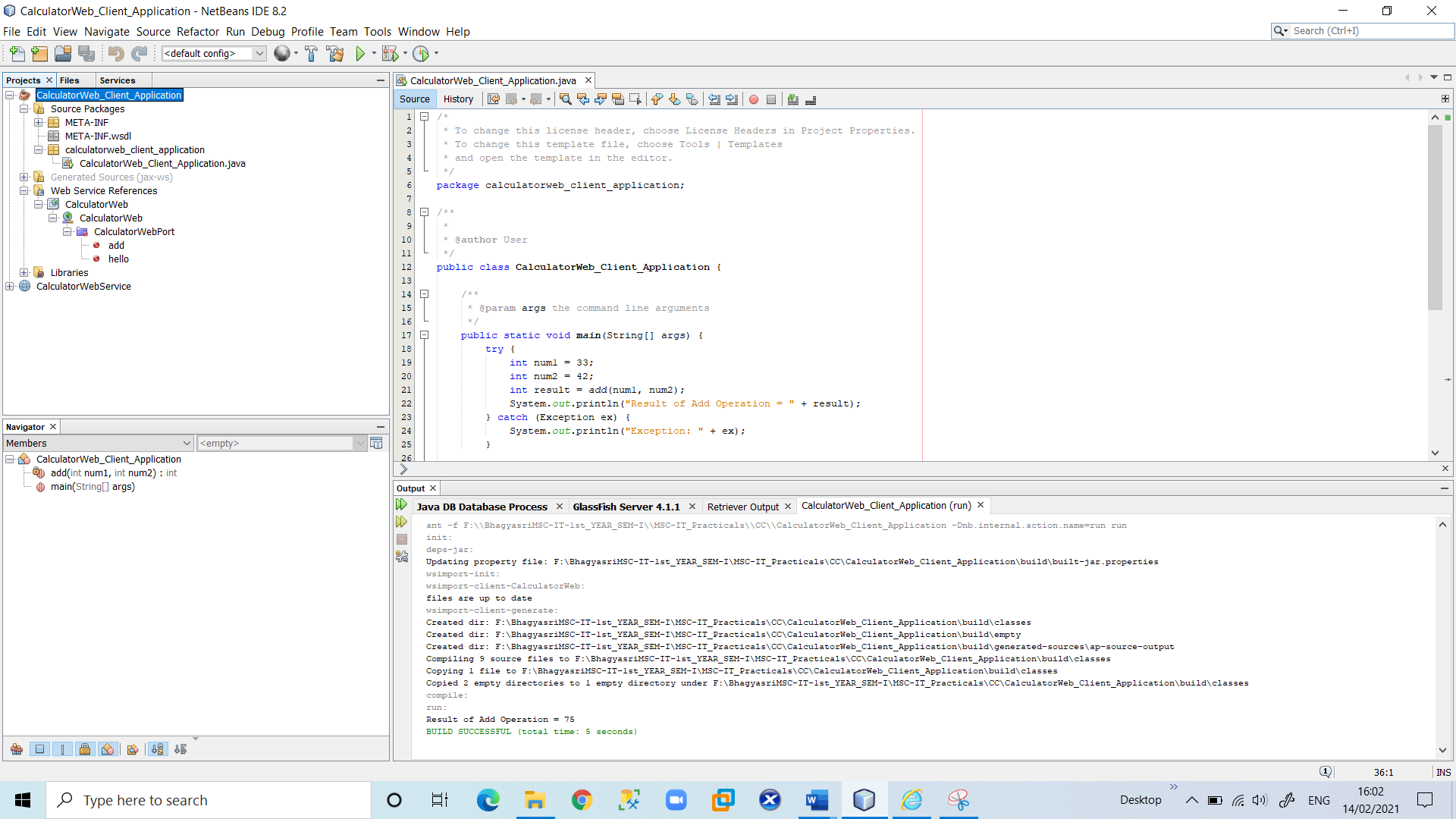


8. Right-click the project node and choose Run.

The Output window now shows the sum:

compile: run: Result of Add Operation = 75 BUILD SUCCESSFUL (total time: 1 second)

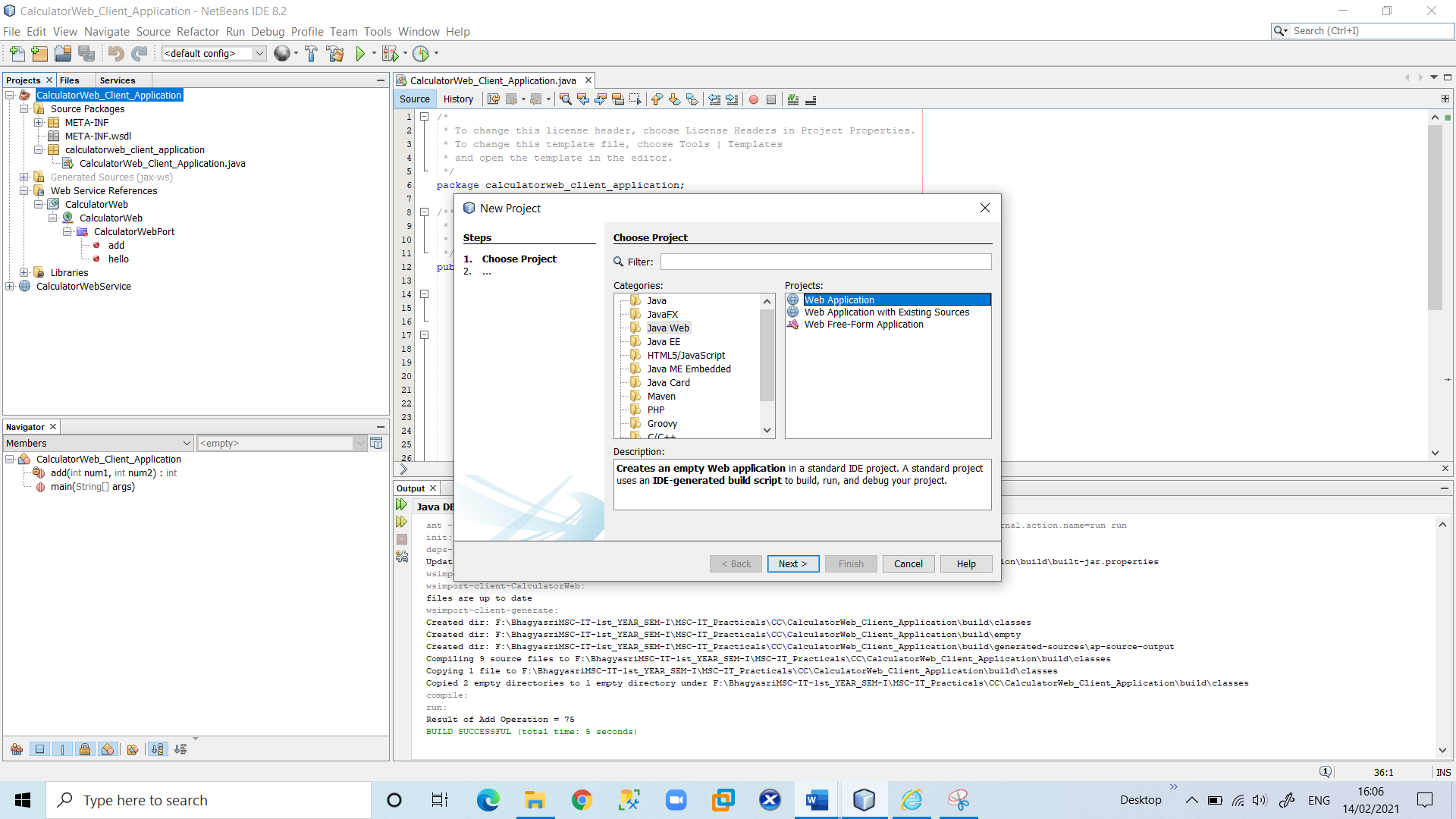


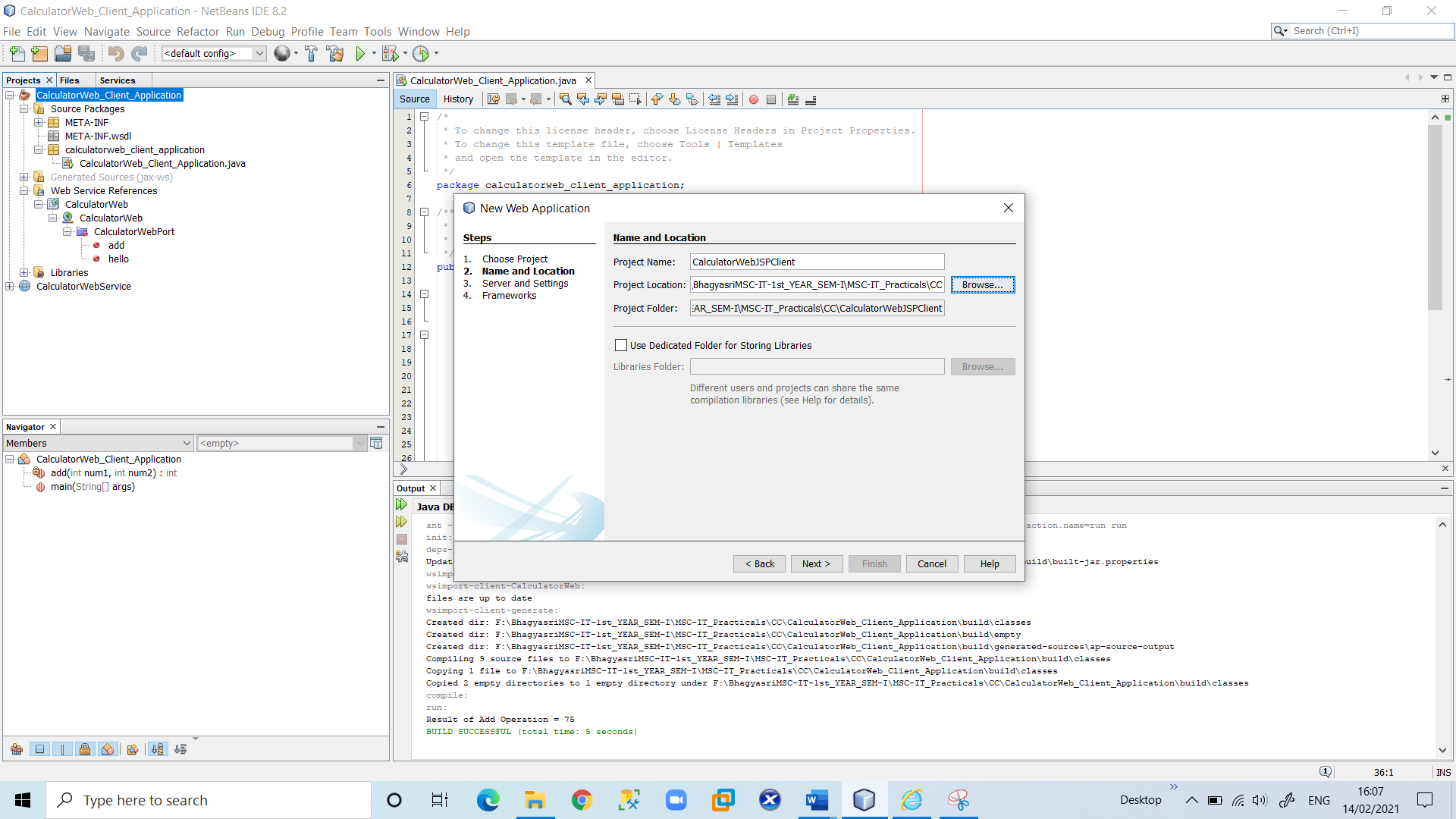


**2.Client : JSP Page in Web Application**

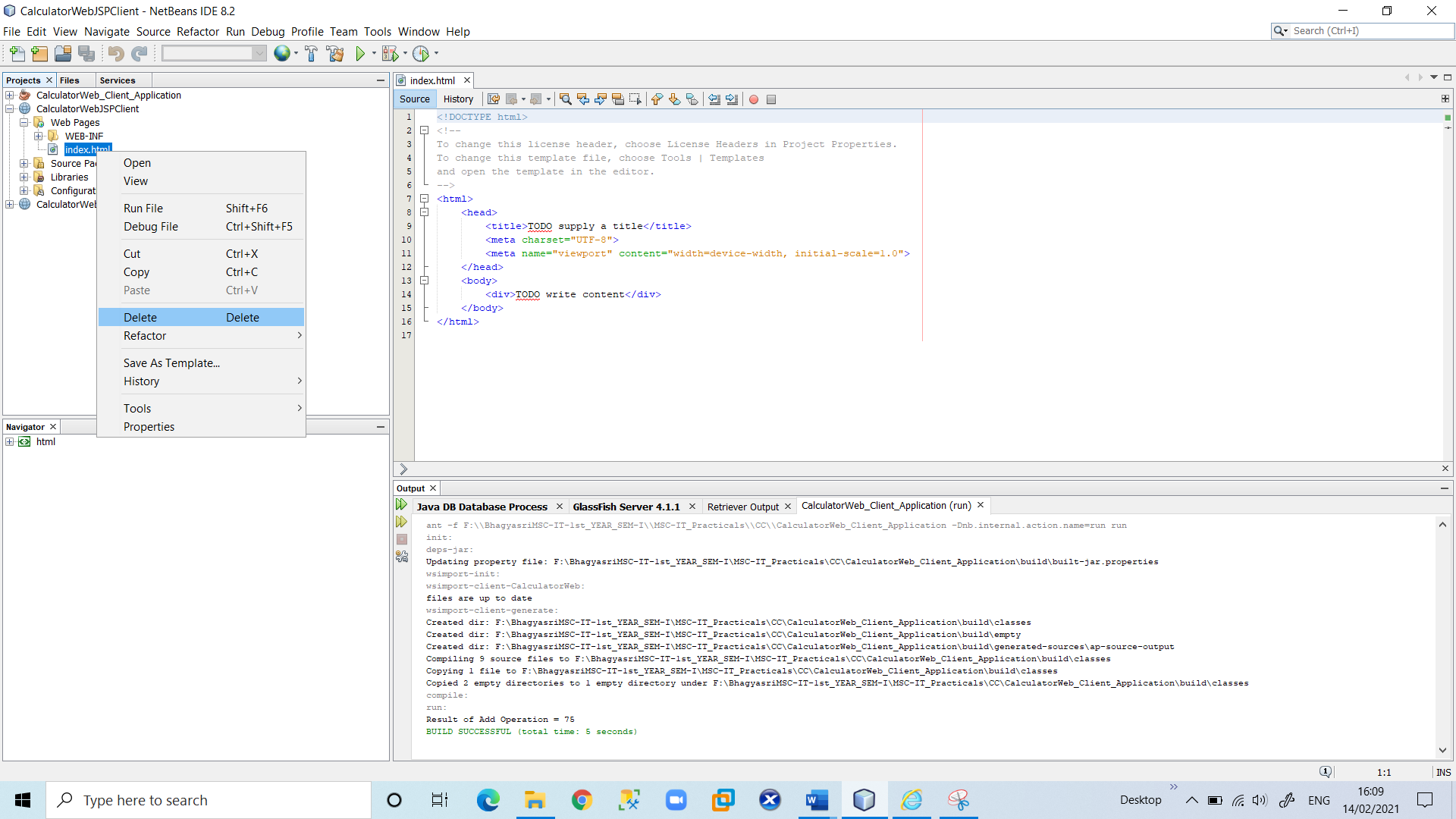
In this section, you create a new web application and then consume the web service in the default JSP page that the Web Application wizard creates.

1.Choose **File > New Project**. Select Web Application from the Java Web category. Name the project “**CalculatorWebJSPClient”**. Click **Next** and then click **Finish**.



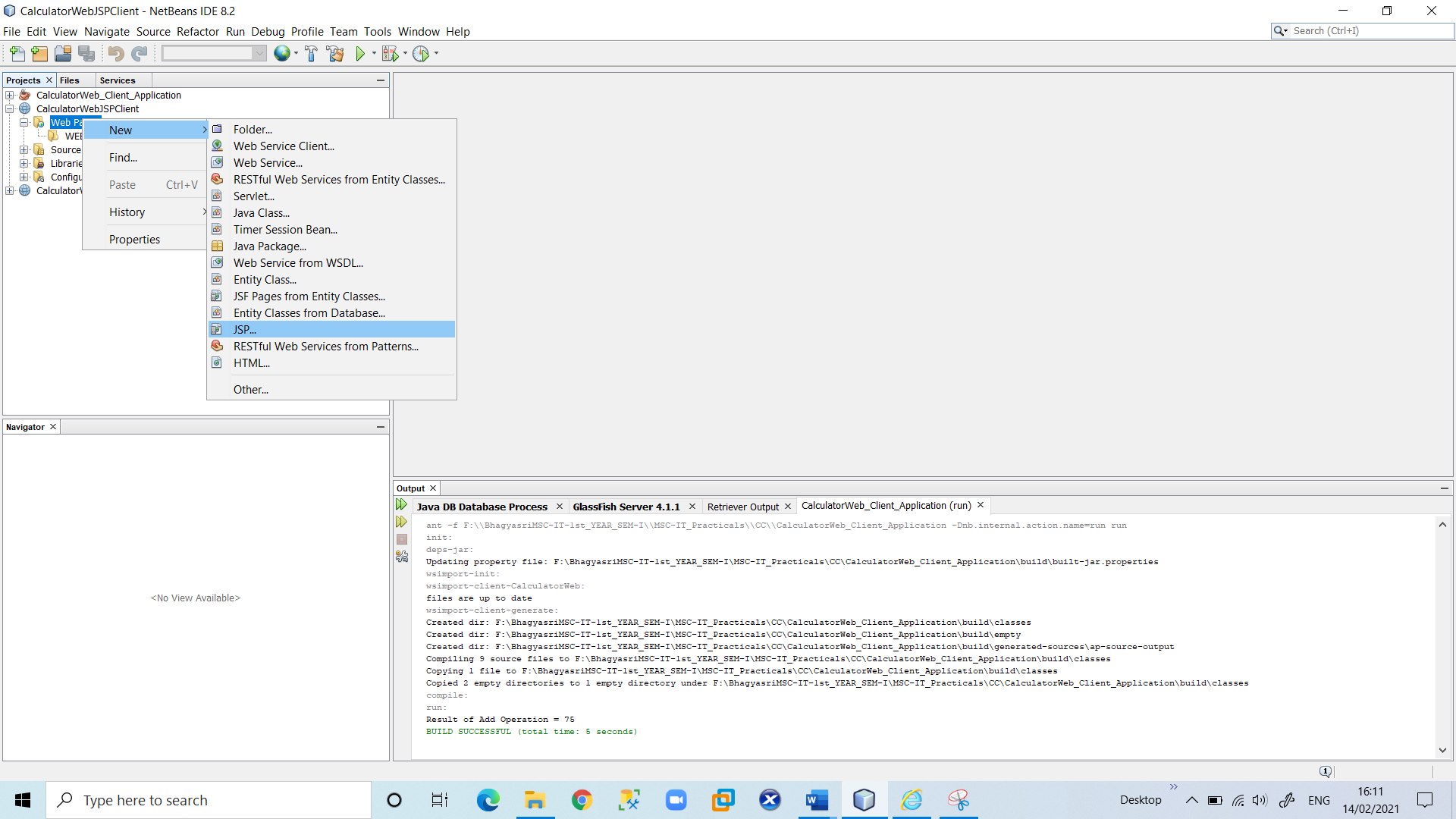


2. Expand the Web Pages node under the project node and delete **index.html**.

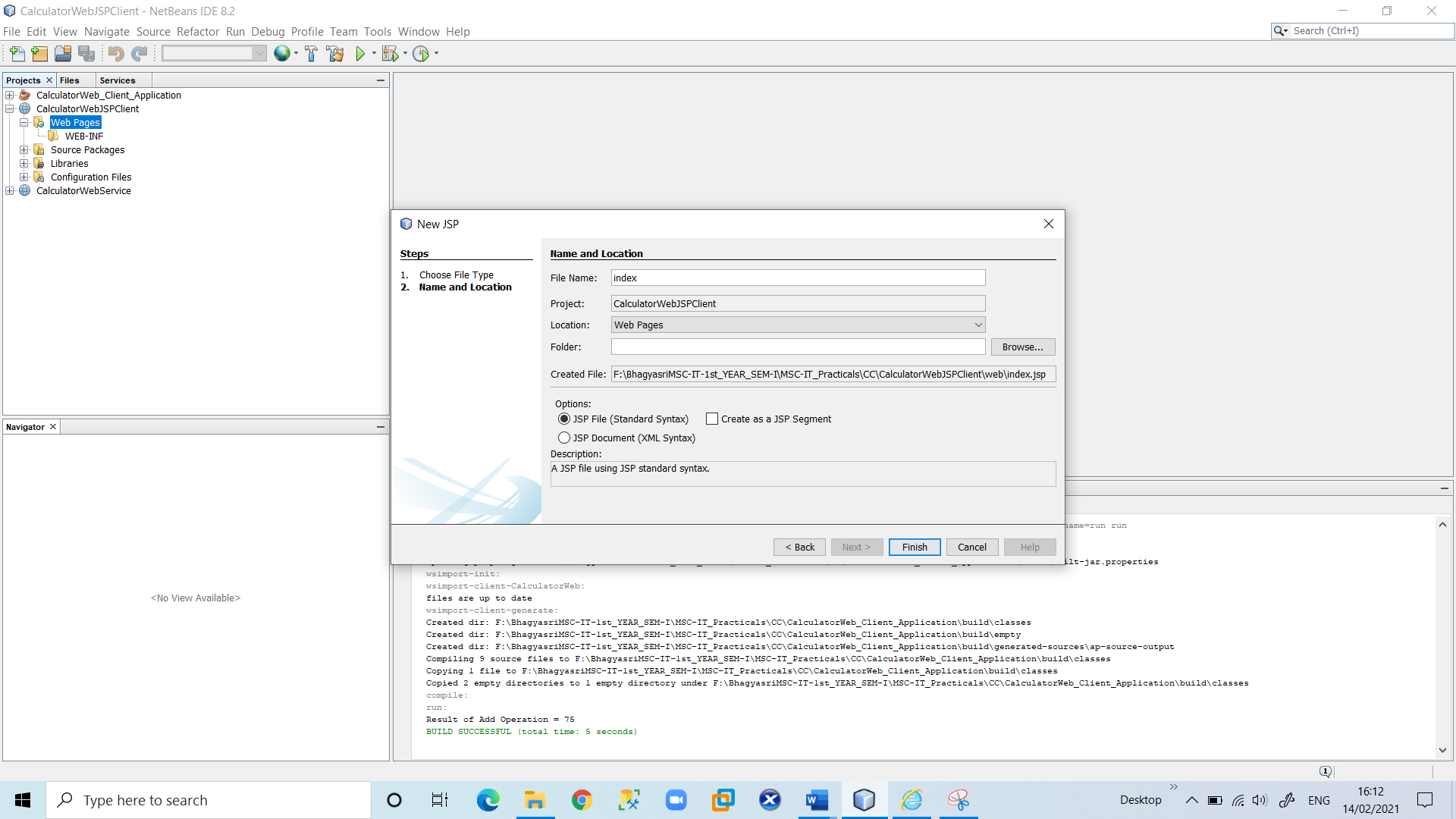


3.Right-click the Web Pages node and choose **New > JSP** in the popup menu.

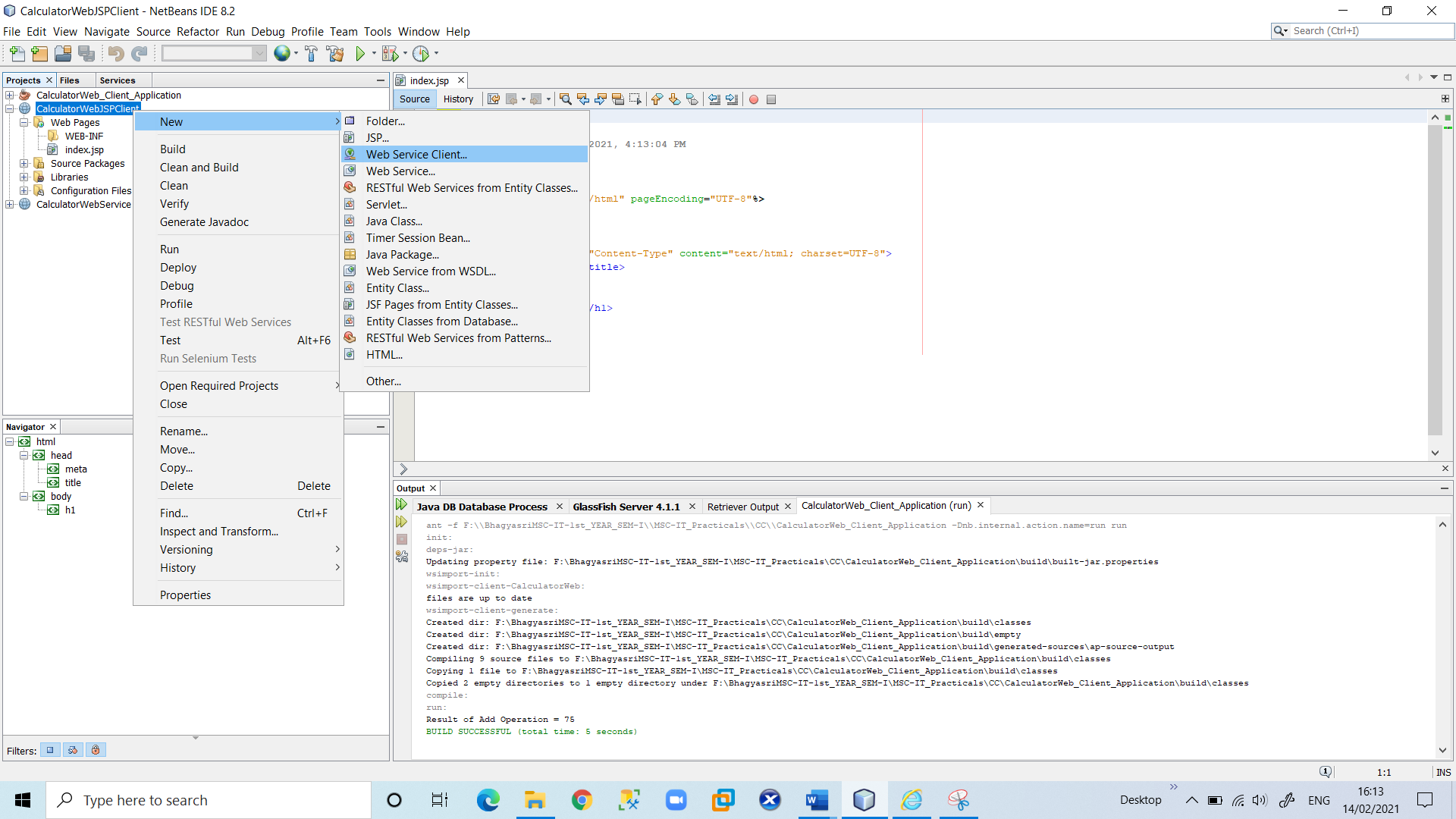
If JSP is not available in the popup menu, choose **New > Other** and select **JSP** in the Web category of the **New File** wizard.



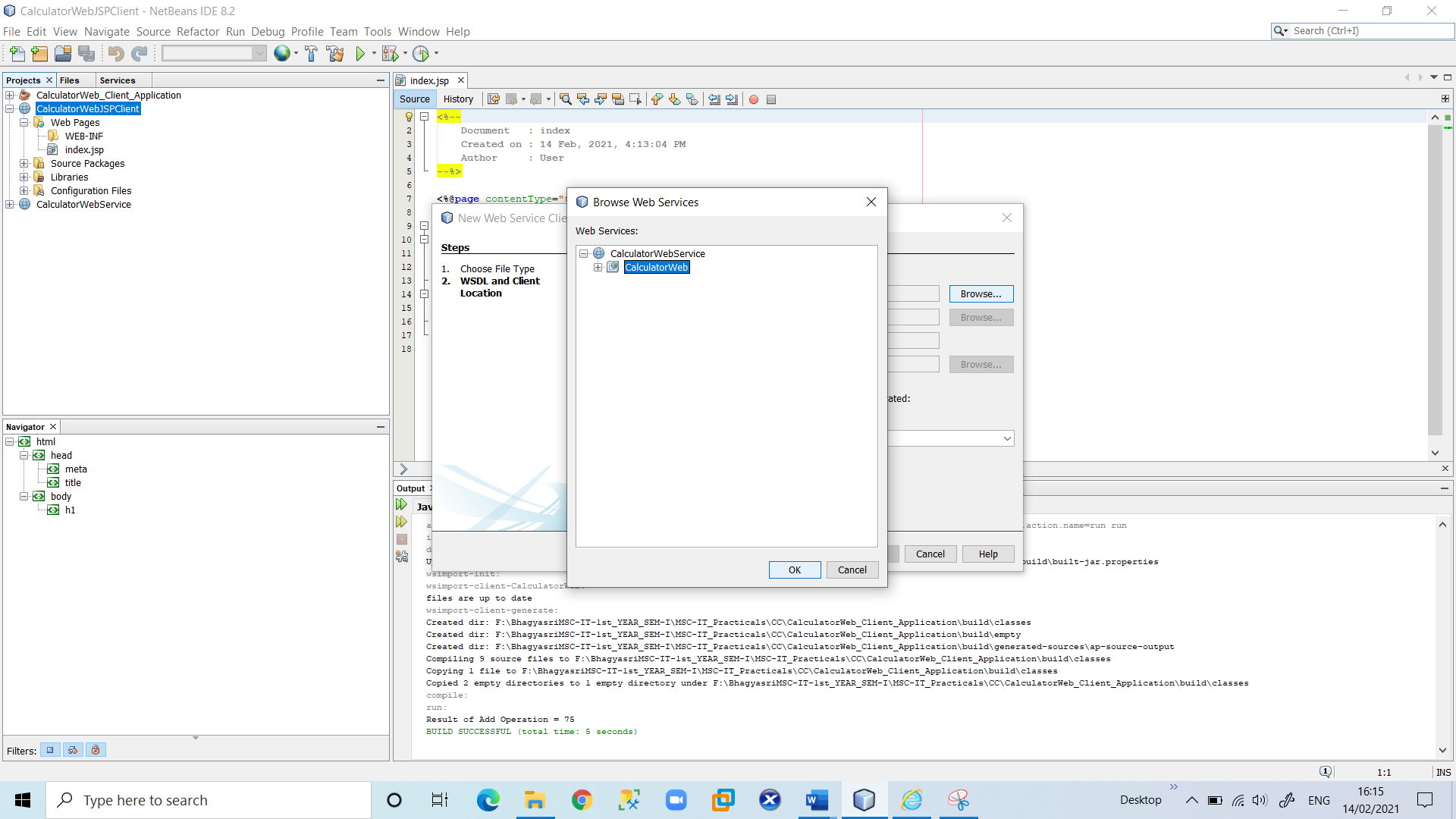
4.Type **index** for the name of the JSP file in the **New File** wizard. Click **Finish**.

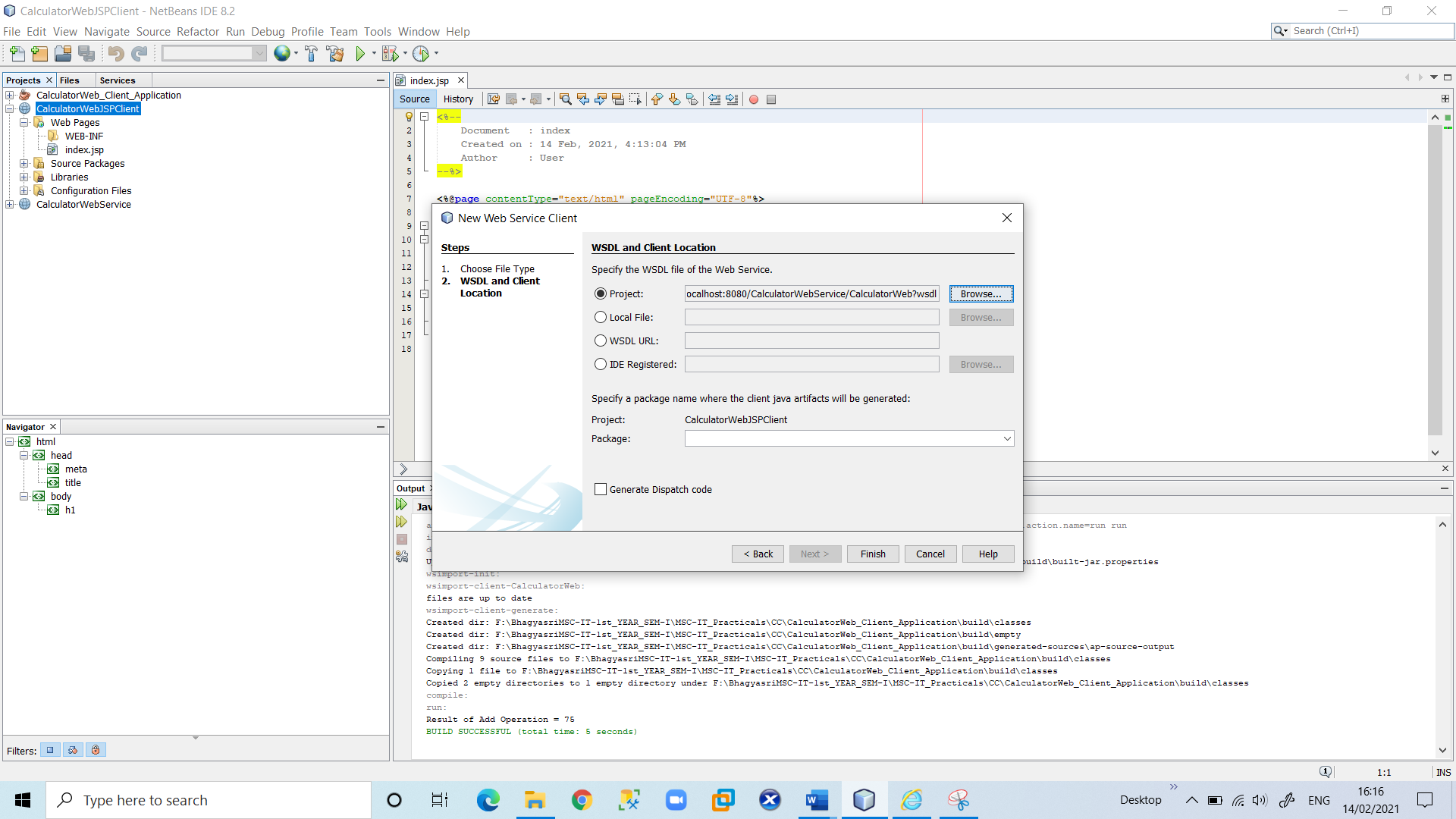


5.Right-click the **CalculatorWebJSPClient** node and choose **New > Web Service Client.**



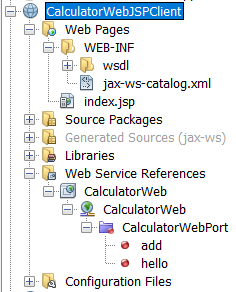
6.Select Project as the WSDL source. Click **Browse**. Browse to the **CalculatorWeb** web service in the **CalculatorWebService** project. When you have selected the web service, click **OK**.





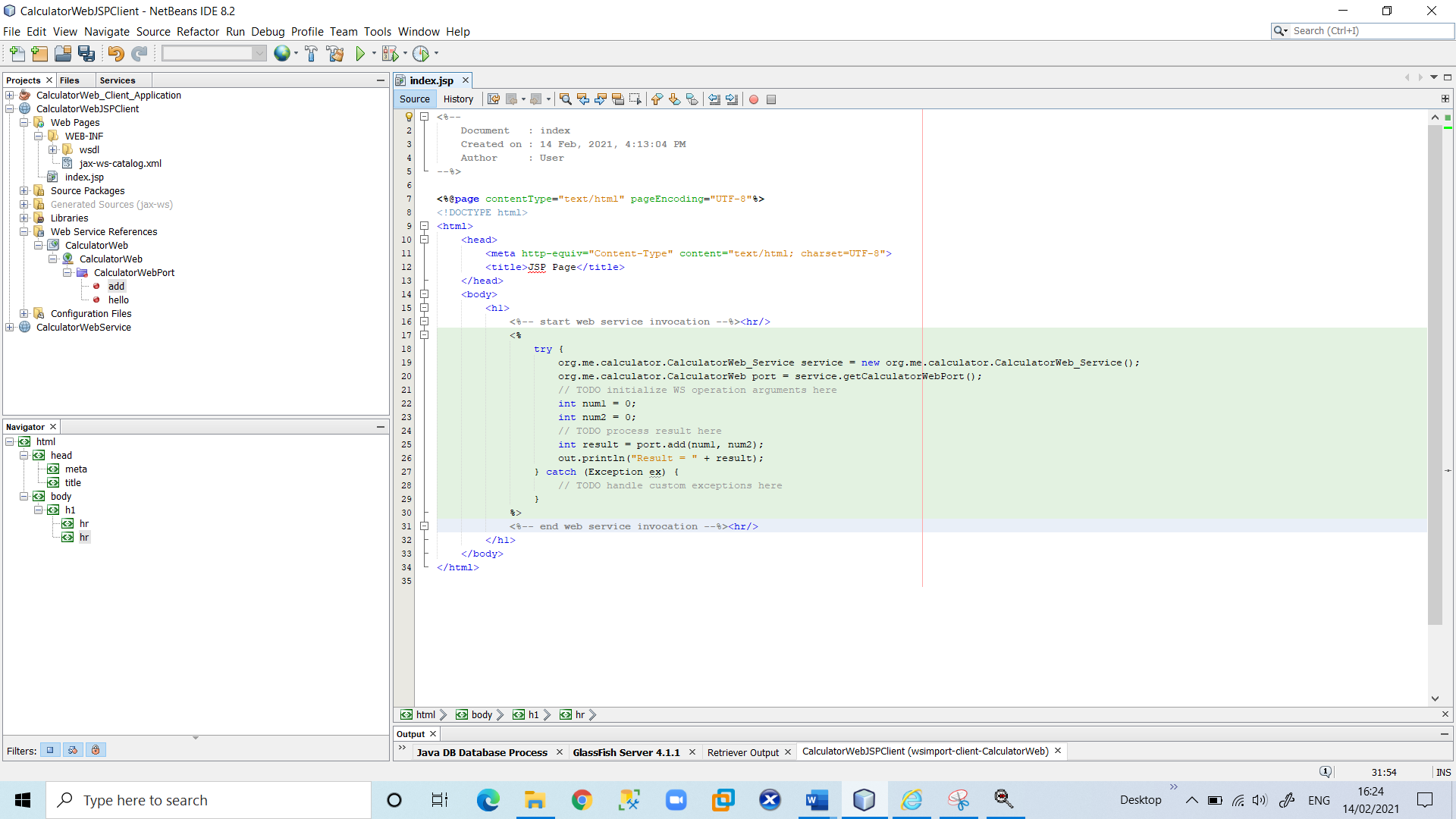
Do not select a package name. Leave this field empty. Leave the other settings at default and click Finish.

The Projects window displays the new web service client, as shown below:

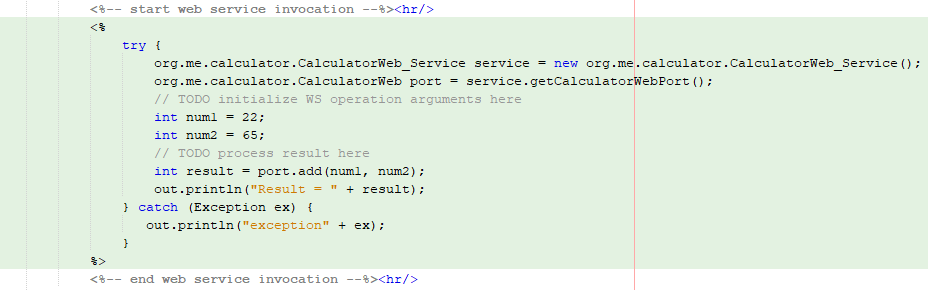


7. In the Web Service References node, expand the node that represents the web service. The add operation, which you will invoke from the client, is now exposed.

Drag the add operation to the client's **index.jsp** page, and drop it below the H1 tag starting tag. The code for invoking the service's operation is now generated in the index.jsp page, as you can see here:



Change the value for num1 and num2 from 0 to other integers, such as 22 and 65. Replace the commented out TODO line in the catch block with out. println ("exception" + ex);



8.Right-click the project node and choose Run.

The server starts, if it wasn't running already. The application is built and deployed, and the browser opens, displaying the calculation of Result:



