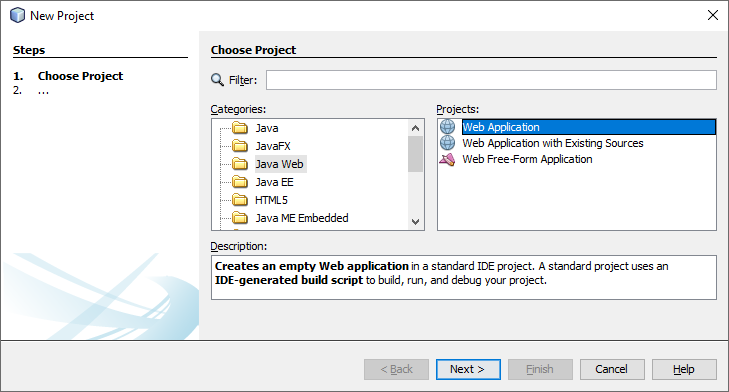
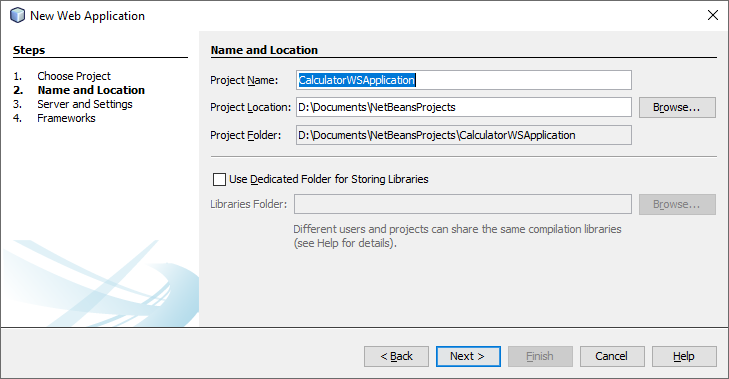
Creating a Web Service

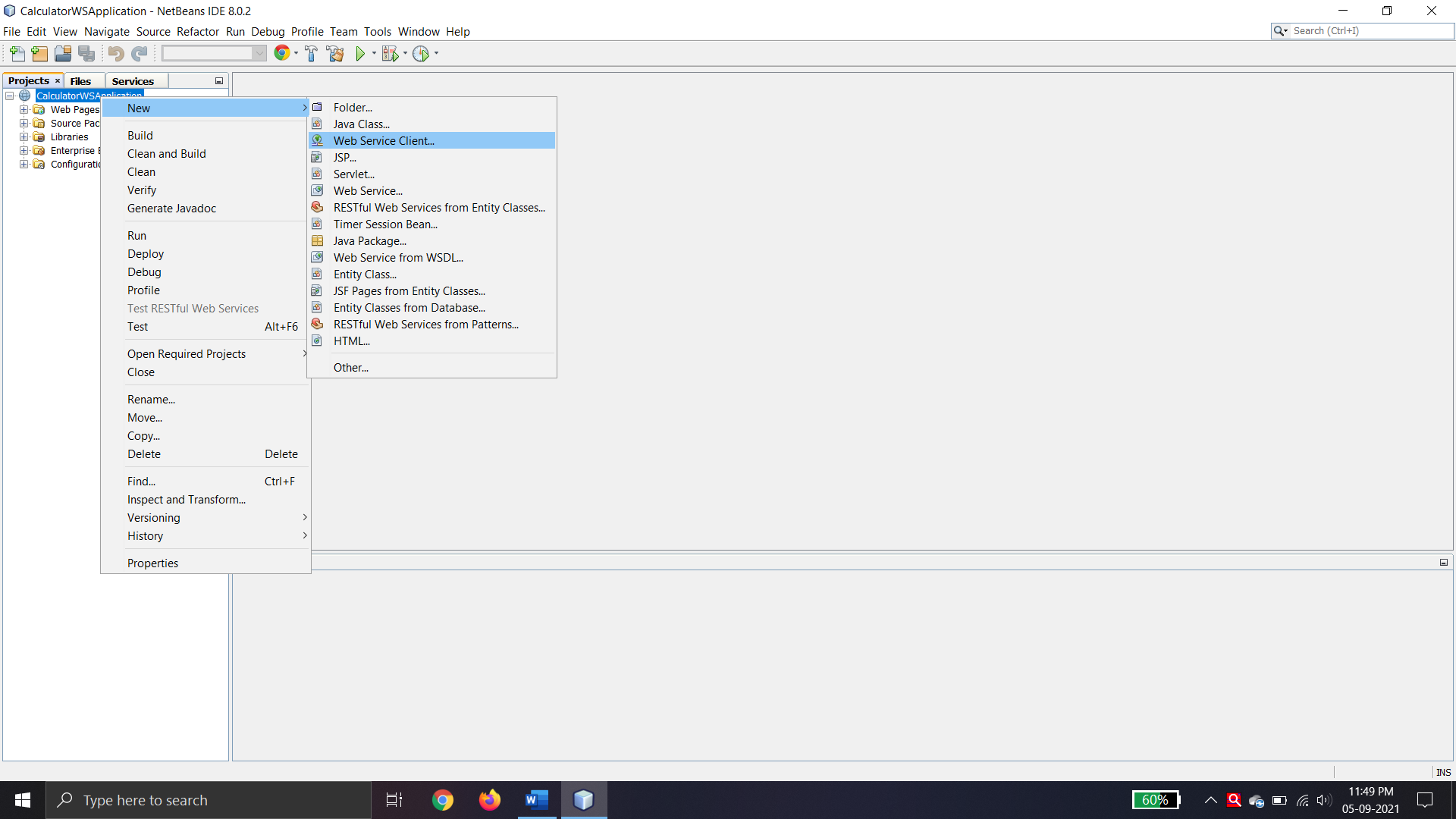
1. **Choosing a Container:**
2. Choose File > New Project. Select Web Application from the Java Web category.



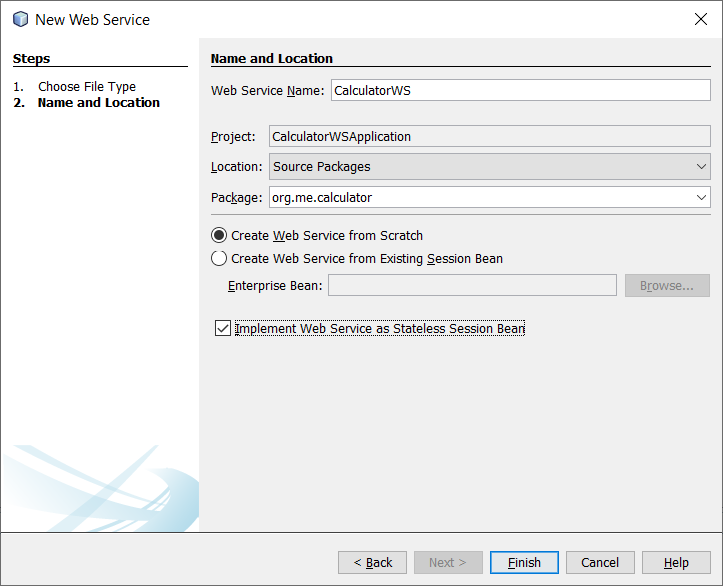
1. Name the project CalculatorWSApplication. Select a location for the project. Click Next.



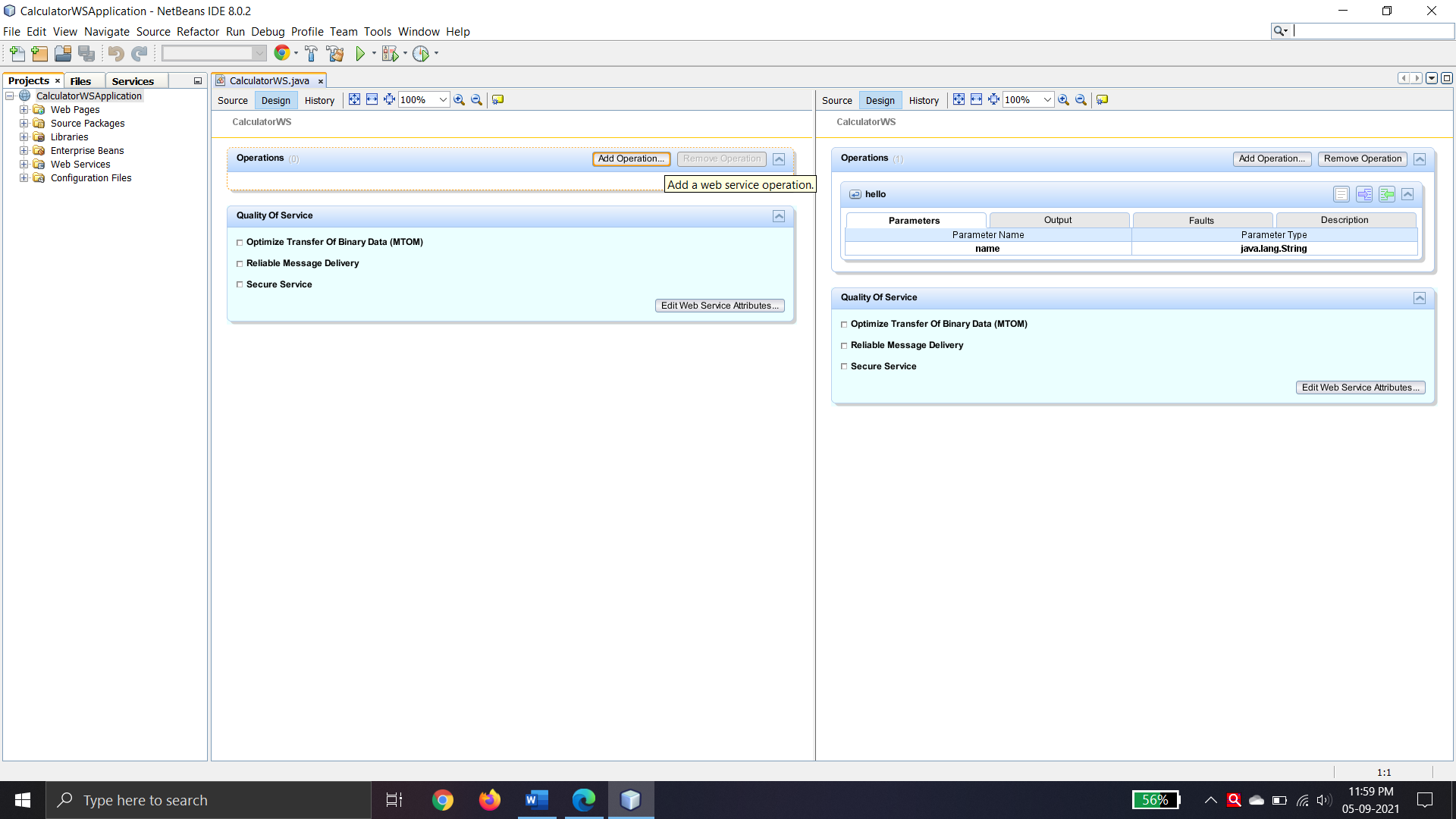
1. Select your server and Java EE version and click Finish.
2. **Creating a Web Service from a Java Class**
3. Right-click the CalculatorWSApplication node and choose New > Web Service.



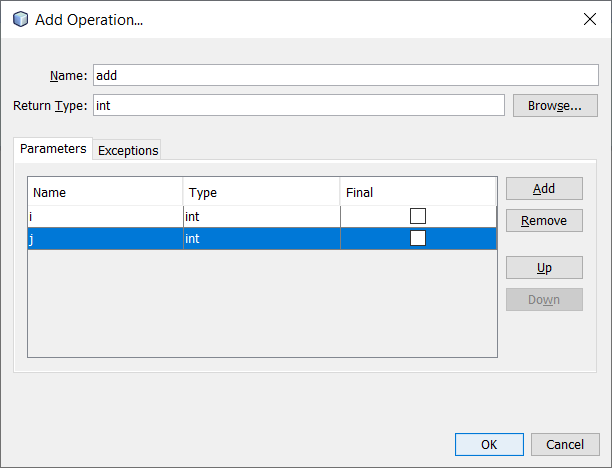
1. Name the web service CalculatorWS and type org.me.calculator in Package. Leave Create Web Service from Scratch selected.
2. If you are creating a Java EE project on GlassFish or WebLogic, select Implement Web Service as a Stateless Session Bean.



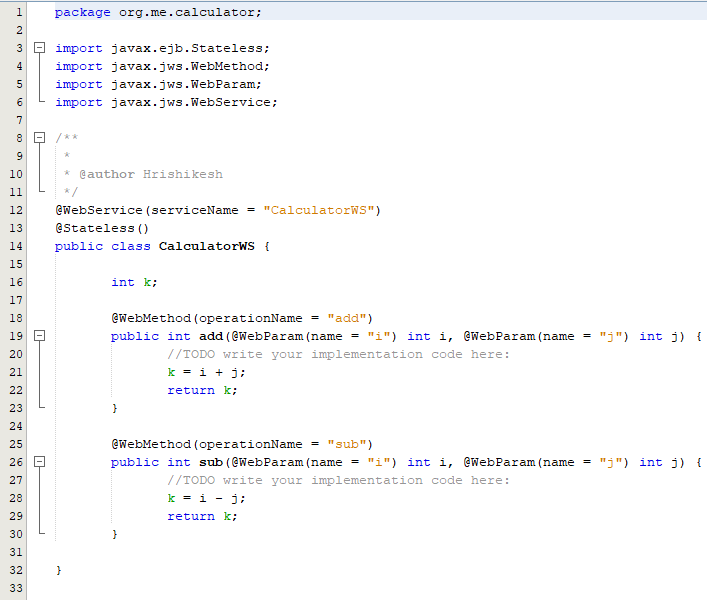
1. Click Finish. The Projects window displays the structure of the new web service and the source code is shown in the editor area.
2. **Adding an Operation to the Web Service**
3. Change to the Design view in the editor.



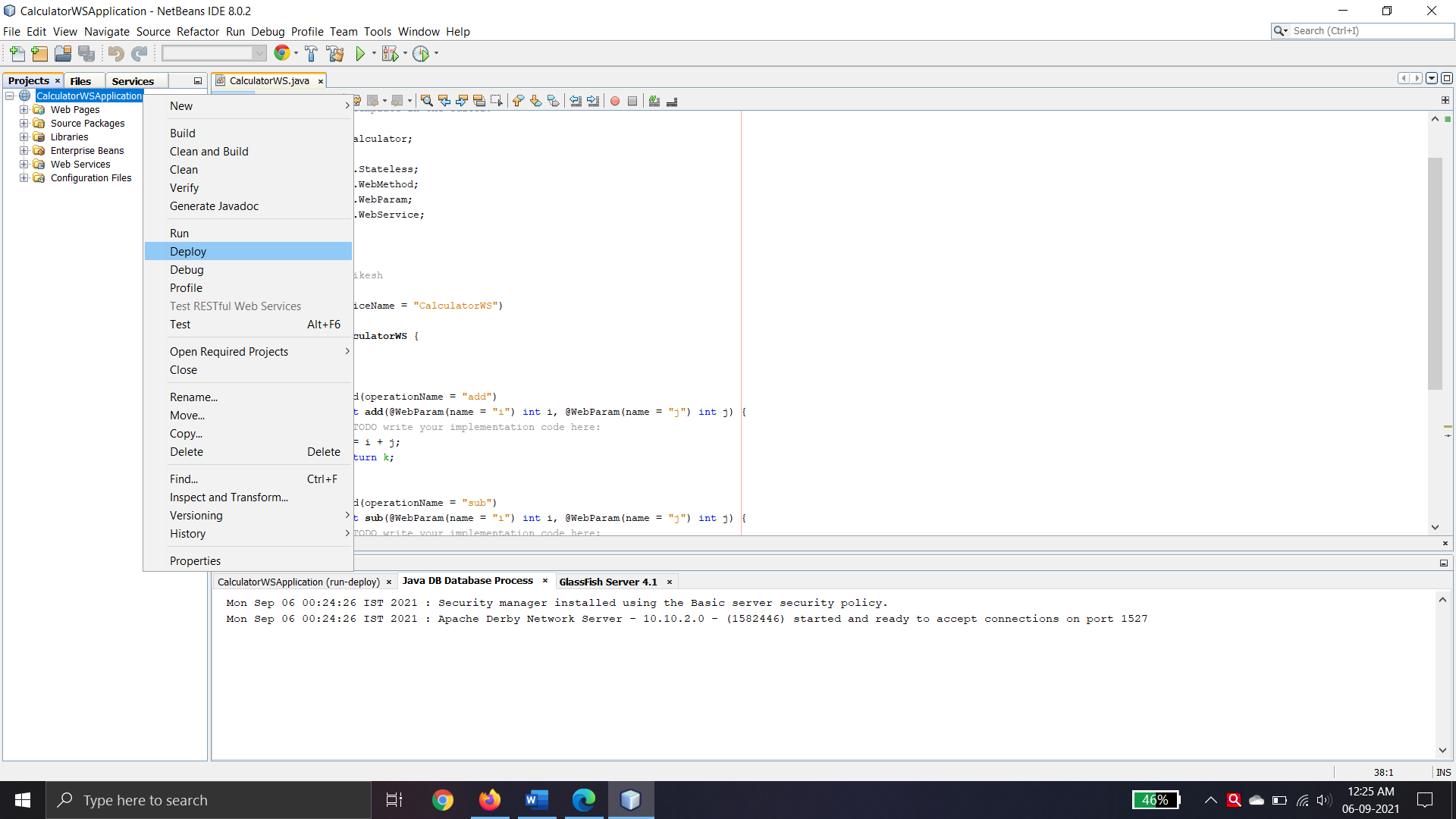
1. Click Add Operation in either the visual designer or the context menu. The Add Operation dialog opens.
2. In the upper part of the Add Operation dialog box, type add in Name and type int in the Return Type drop-down list.
3. In the lower part of the Add Operation dialog box, click Add and create a parameter of type int named i.
4. Click Add again and create a parameter of type int called j.



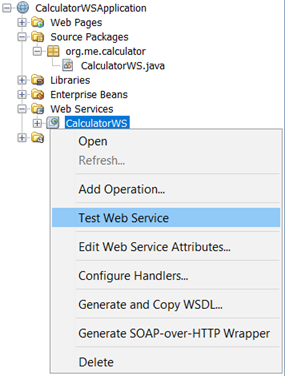
1. Click OK at the bottom of the Add Operation dialog box. You return to the editor.
2. Click Source. And code the following.



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

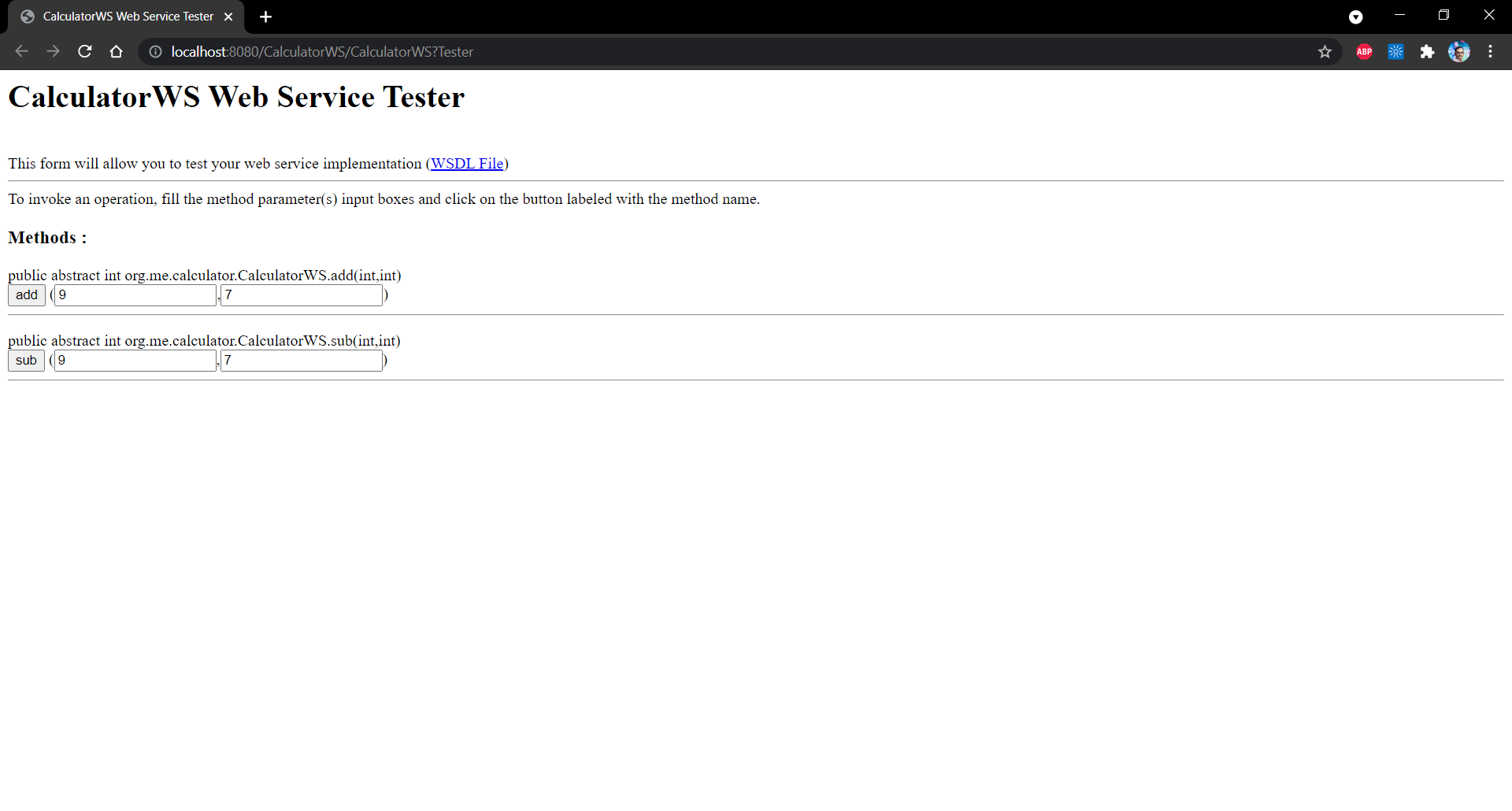


1. In the IDE’s Projects tab, expand the Web Services node of the CalculatorWSApplication project. Right-click the CalculatorWS node, and choose Test Web Service.

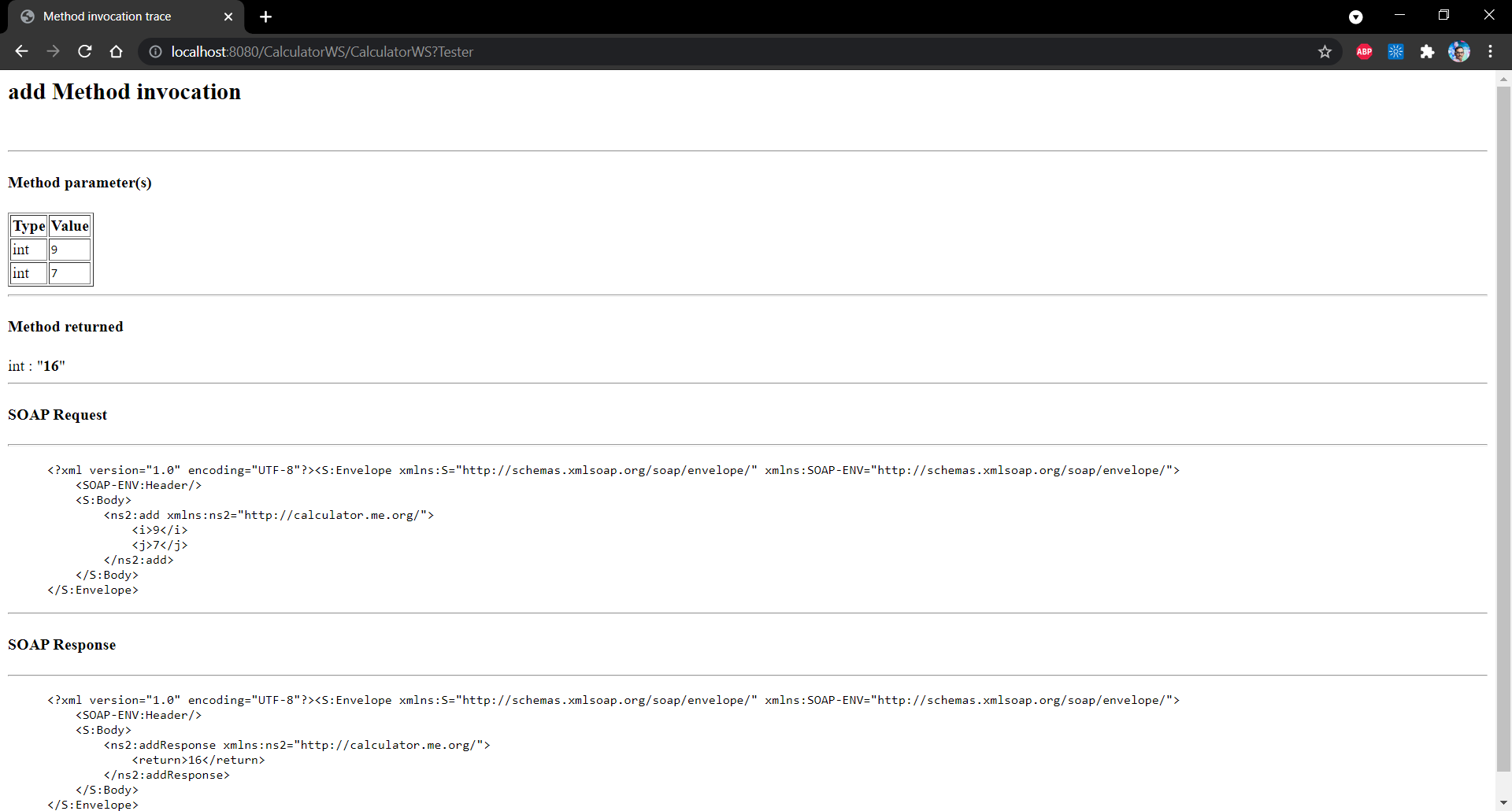


1. The IDE opens the tester page in your browser, if you deployed a web application to the GlassFish server. For the Tomcat Web Server and deployment of EJB modules, the situation is different:

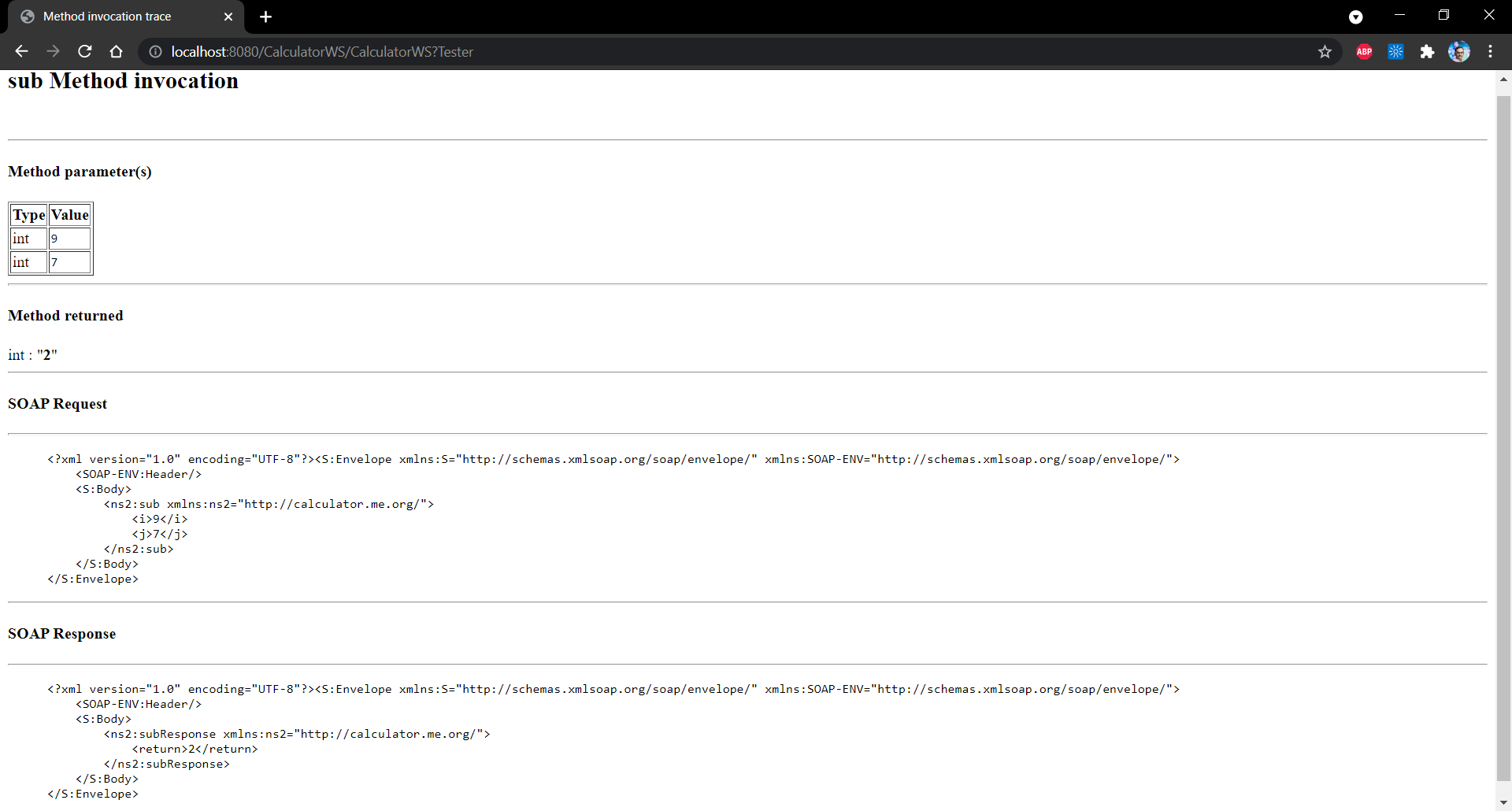
* If you deployed to the GlassFish server, type two numbers in the tester page, as shown below:



* The sum of the two numbers is displayed:



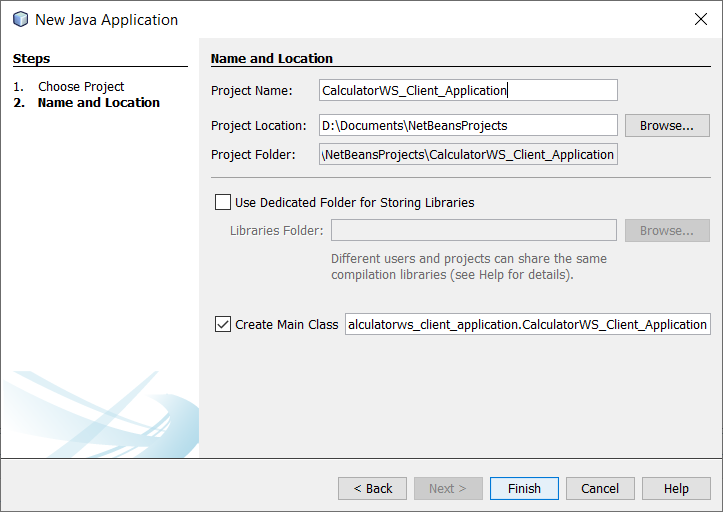
* The Subtraction of the two numbers is displayed:



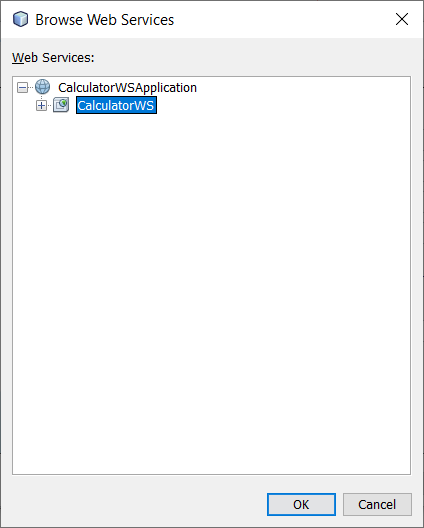
Consuming the Web Service

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

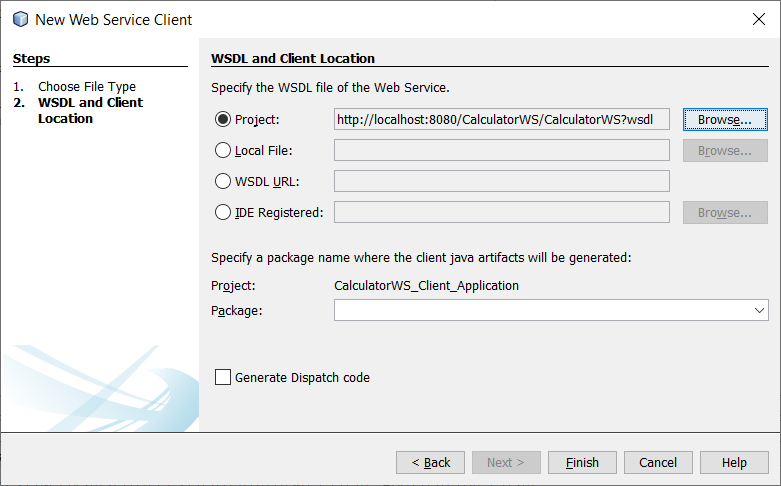
1. Choose File > New Project. Select Java Application from the Java category.
2. Name the project CalculatorWS\_Client\_Application.
3. Leave Create Main Class selected and accept all other default settings. Click Finish.



1. Right-click the CalculatorWS\_Client\_Application node and choose New > Web Service Client. The New Web Service Client wizard opens.
2. Select Project as the WSDL source. Click Browse. Browse to the CalculatorWS web service in the CalculatorWSApplication project. When you have selected the web service, click OK.

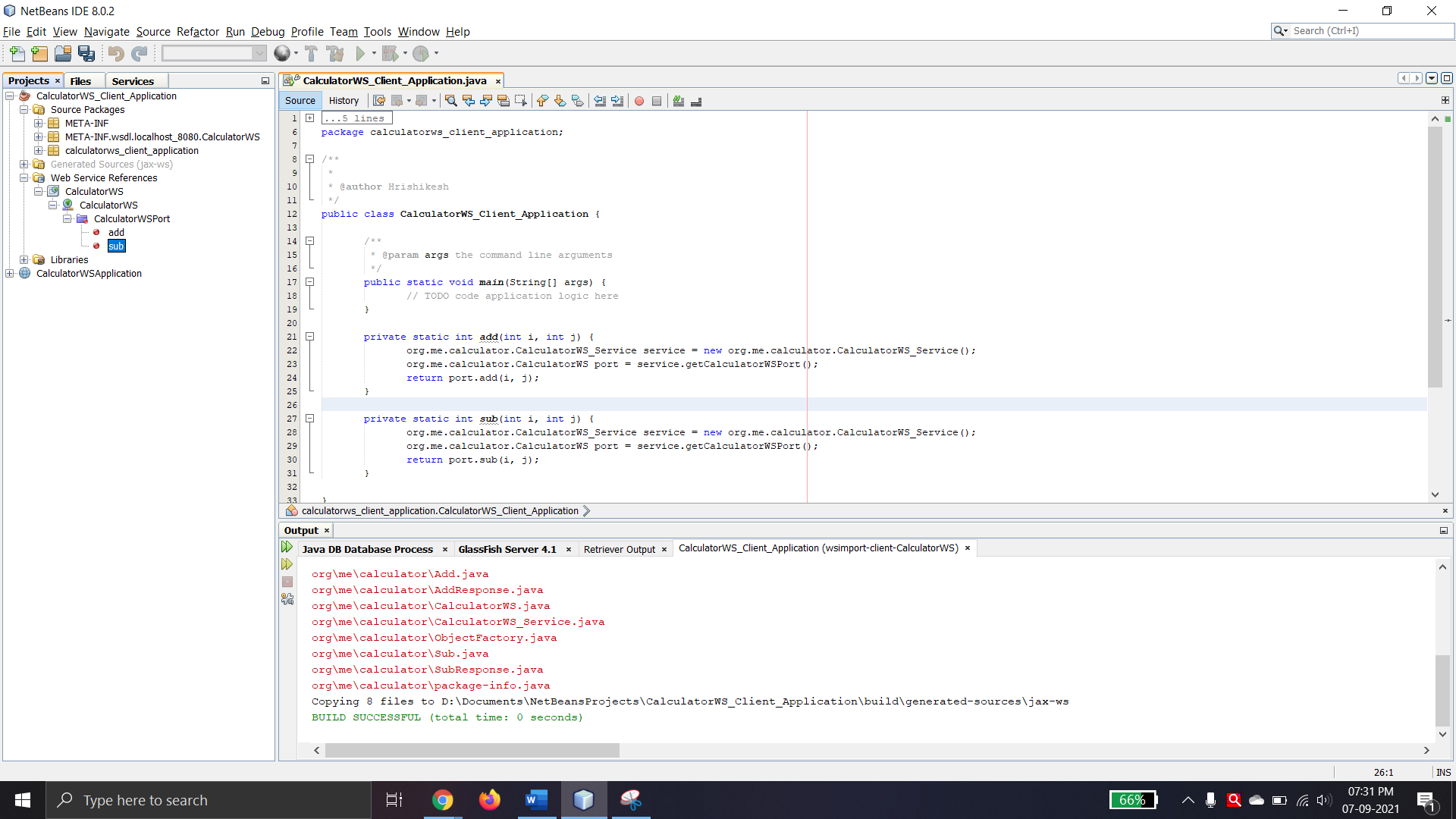


1. Do not select a package name. Leave this field empty.

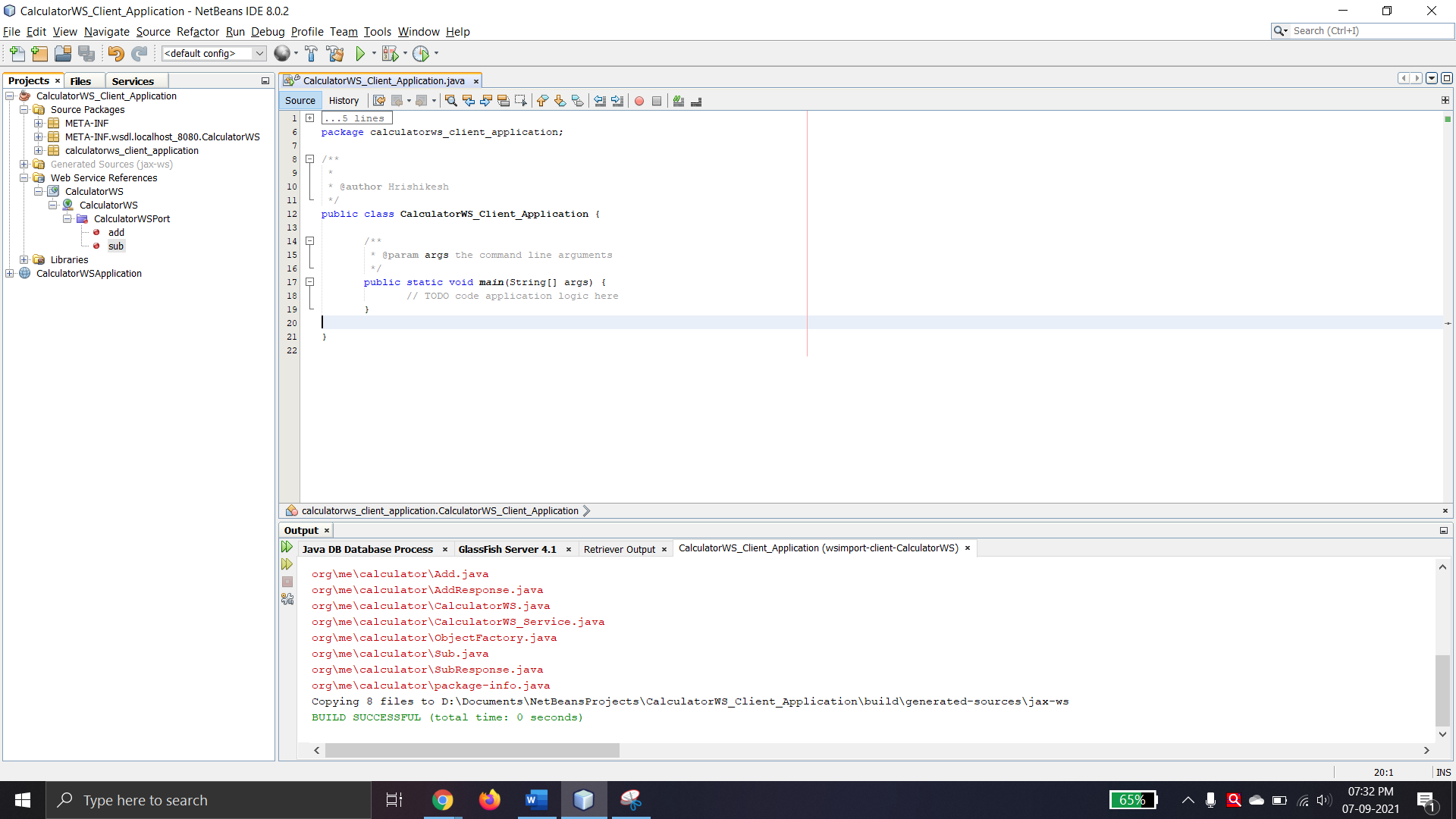


1. Leave the other settings at default and click Finish.

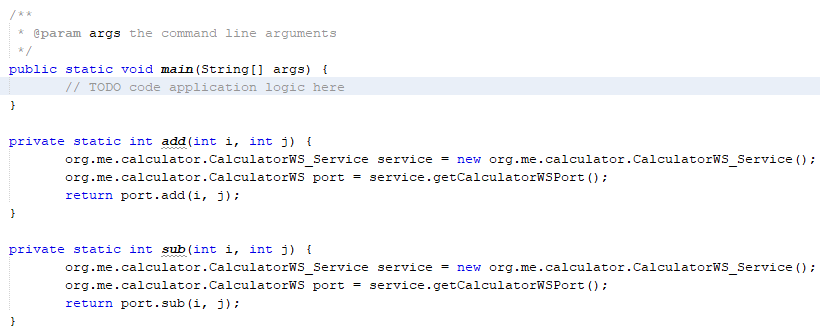
The Projects window displays the new web service client, with a node for the add method that you created:



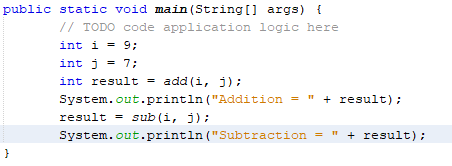
1. Double-click your main class so that it opens in the Source Editor. Drag the add node below the main() method.



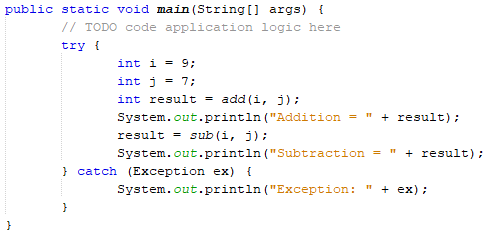
You now see the following:



1. In the main() method body, replace the TODO comment with code that initializes values for i and j , calls add() ,calls sub() and prints the results.

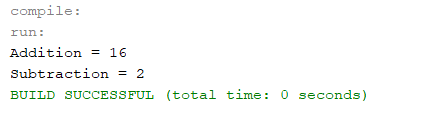


1. Surround the main() method code with a try/catch block that prints an exception.



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

The Output window now shows the sum:

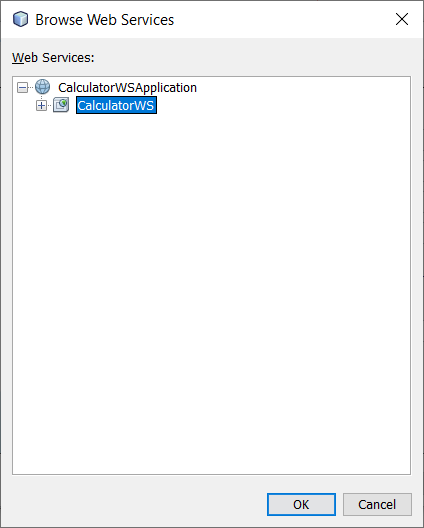


**Client 2:** **Servlet in Web Application**

1. Choose File > New Project. Select Web Application from the Java Web category. Name the project CalculatorWSServletClient . Click Next and then click Finish.
2. Right-click the CalculatorWSServletClient node and choose New > Web Service Client.

The New Web Service Client wizard opens.

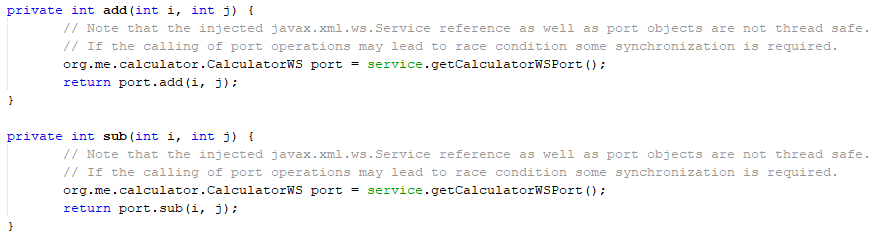
1. Select Project as the WSDL source and click Browse to open the Browse Web Services dialog box.
2. Select the CalculatorWS web service in the CalculatorWSApplication project. Click OK to close the Browse Web Services dialog box.



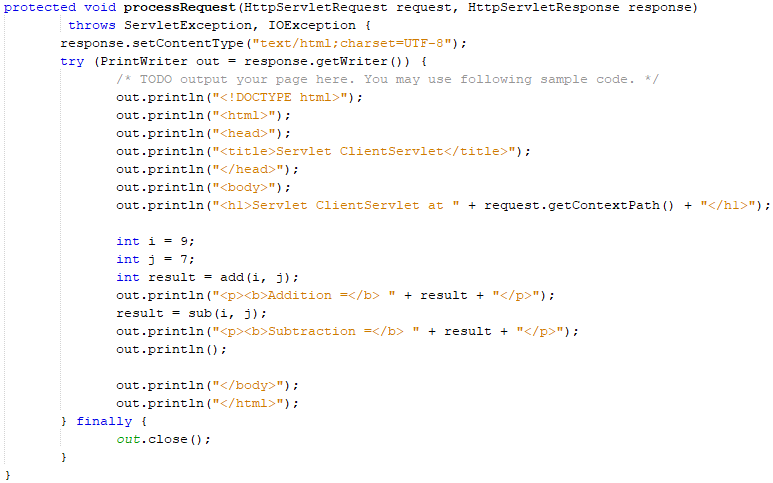
1. Confirm that the package name is empty in the New Web Service Client wizard and leave the other settings at the default value. Click Finish.
2. Right-click the CalculatorWSServletClient project node and choose New > Servlet. Name the servlet ClientServlet and place it in a package called org.me.calculator.client . Click Finish.
3. To make the servlet the entry point to your application, right-click the CalculatorWSServletClient project node and choose Properties. Open the Run properties and type /ClientServlet in the Relative URL field. Click OK.
4. If there are error icons for ClientServlet.java , right-click the project node and select Clean and Build.
5. In the processRequest() method, add some empty lines after this line:

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

1. In the Source Editor, drag the add operation anywhere in the body of the ClientServlet class. The add() method and sub() method appears at the end of the class code.



1. Add code that initializes values for i and j , calls add() , and prints the result. The added code is in **boldface**:

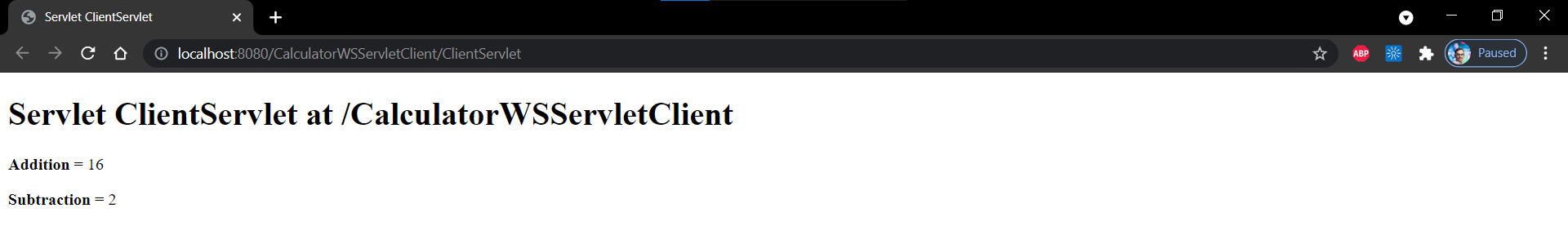


1. Surround the added code with a try/catch block that prints an exception.



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

The server starts, the application is built and deployed, and the browser opens, displaying the calculation result, as shown below:

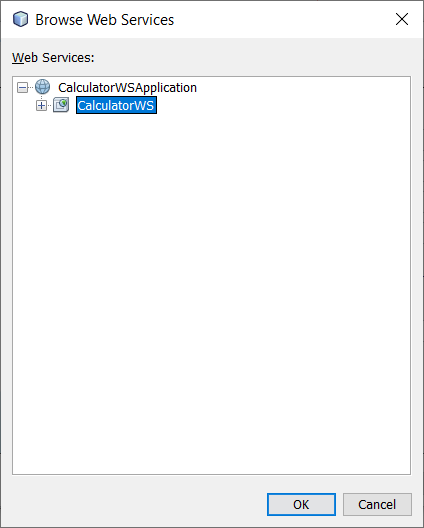


**Client 3:** **JSP Page in Web Application**

1. Choose File > New Project (Ctrl-Shift-N on Linux and Windows, ⌘-Shift-N on MacOS). Select Web Application from the Java Web category. Name the project CalculatorWSJSPClient . 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.

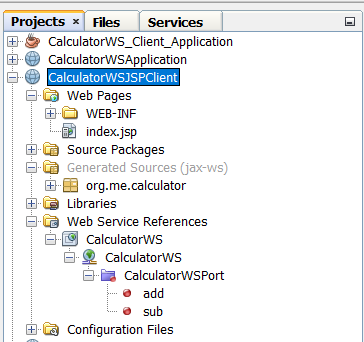
If JSP is not available in the popup menu, choose New > Other and select JSP in the Web category of the New File wizard.

1. Type index for the name of the JSP file in the New File wizard. Click Finish.
2. Right-click the CalculatorWSJSPClient node and choose New > Web Service Client.
3. Select Project as the WSDL source. Click Browse. Browse to the CalculatorWS web service in the CalculatorWSApplication project. When you have selected the web service, click OK.

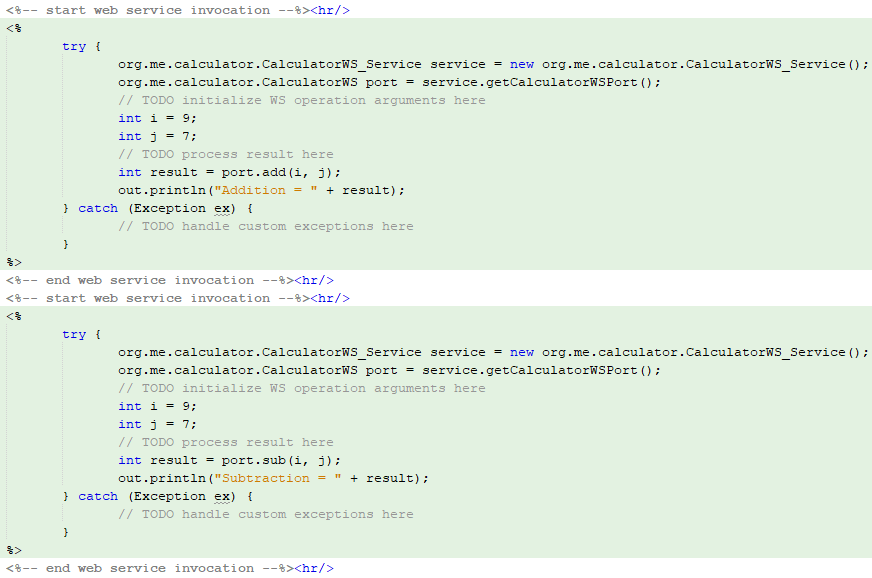


1. Do not select a package name. Leave this field empty.
2. Leave the other settings at default and click Finish.

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



1. 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.
2. Drag the add operation to the client’s index.jsp page, and drop it below the H1 tags. 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 i and j from 0 to other integers, such as 3 and 4. Replace the commented out TODO line in the catch block with out.println("exception" + ex); .

1. 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 result:

