TP 2:

Service web Calculatrice avec différents clients

Creating a Web Service

The goal of this exercise is to create a project appropriate to the deployment container that you decide to use. Once you have a project, you will create a web service in it.

Choosing a Container

You can either deploy your web service in a web container or in an EJB container.

If you are creating a Java EE application, use a web container in any case, because you can put EJBs directly in a web application.

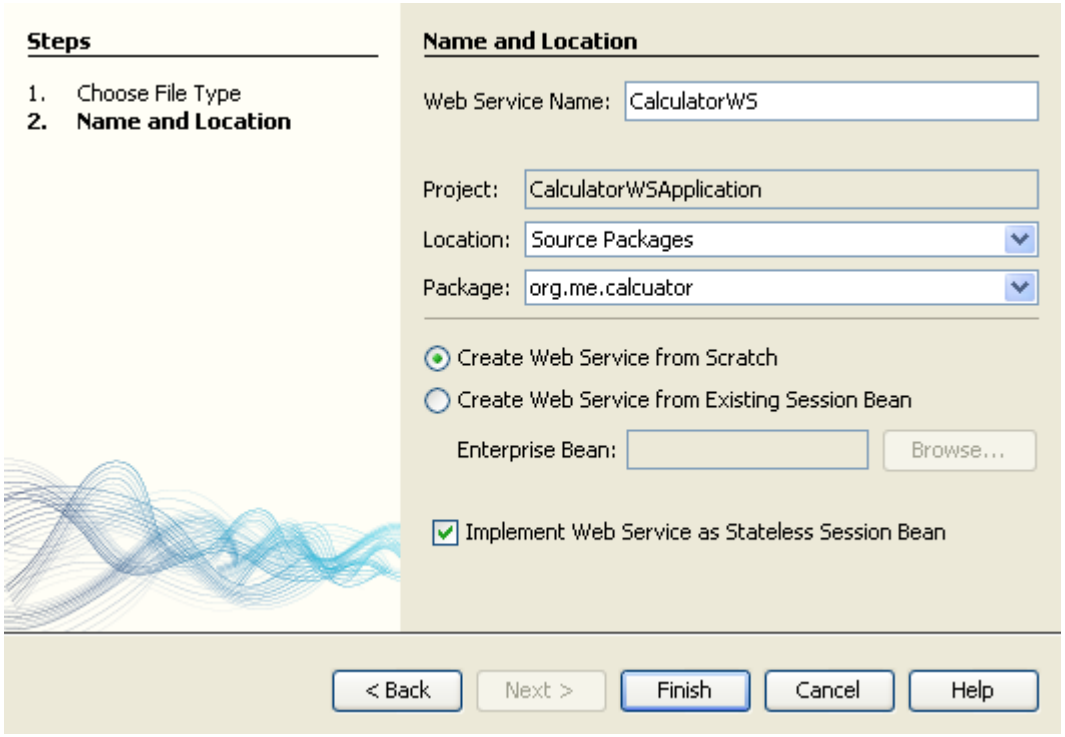
Choose File > New Project

Select Web Application from the Java Web category

1. Name the project `CalculatorWSApplication`.
2. Select a location for the project.
3. Click Next.
4. Select your server and Java EE version and click Finish.

Creating a Web Service from a Java Class

1. Right-click the `CalculatorWSApplication` node
2. and choose New > Web Service.
3. Name the web service `CalculatorWS`
4. and type `packCal` in Package.
5. Leave Create Web Service from Scratch selected.



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

Adding an Operation to the Web Service

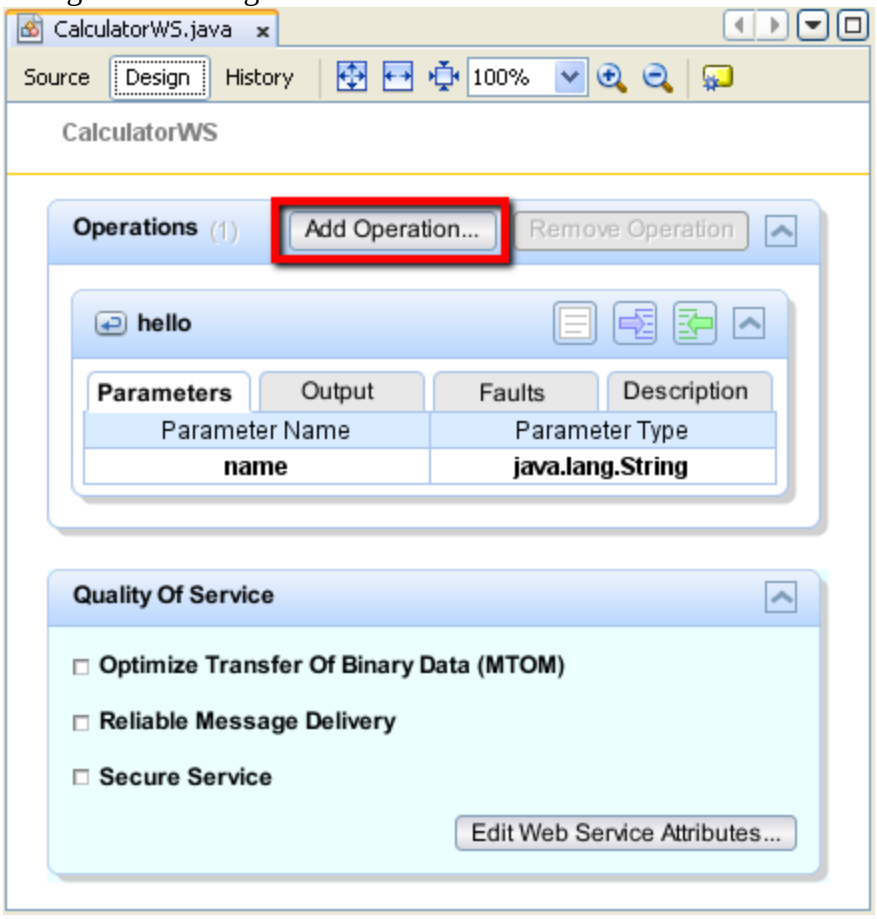
The goal of this exercise is to add to the web service an operation that adds two numbers received from a client.

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.

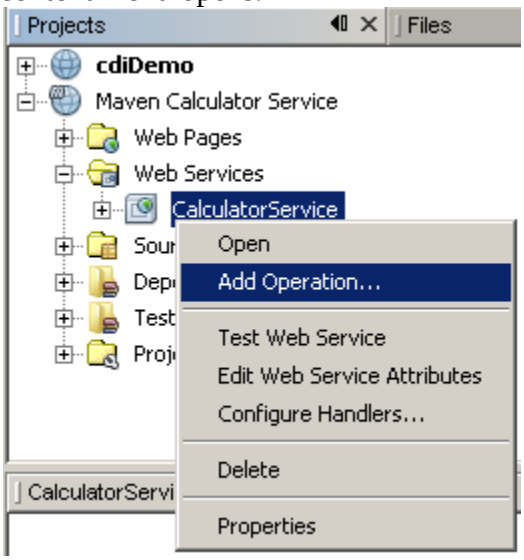
To add an operation to the web service:

1. Either:
    - Change to the Design view in the editor.



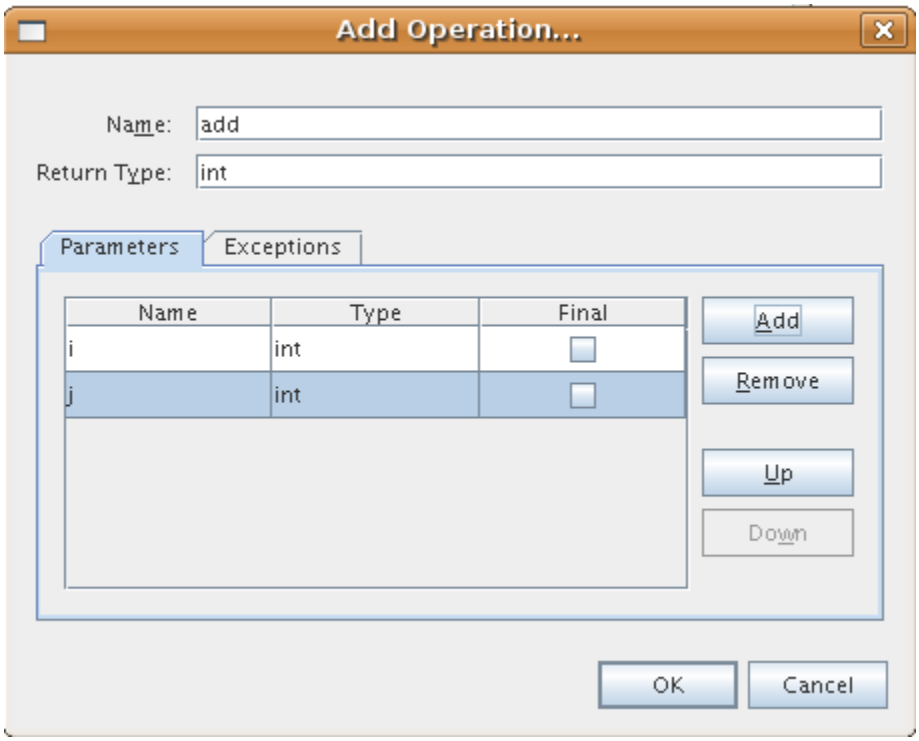
Or:

- Find the web service's node in the Projects window. Right-click that node. A context menu opens.



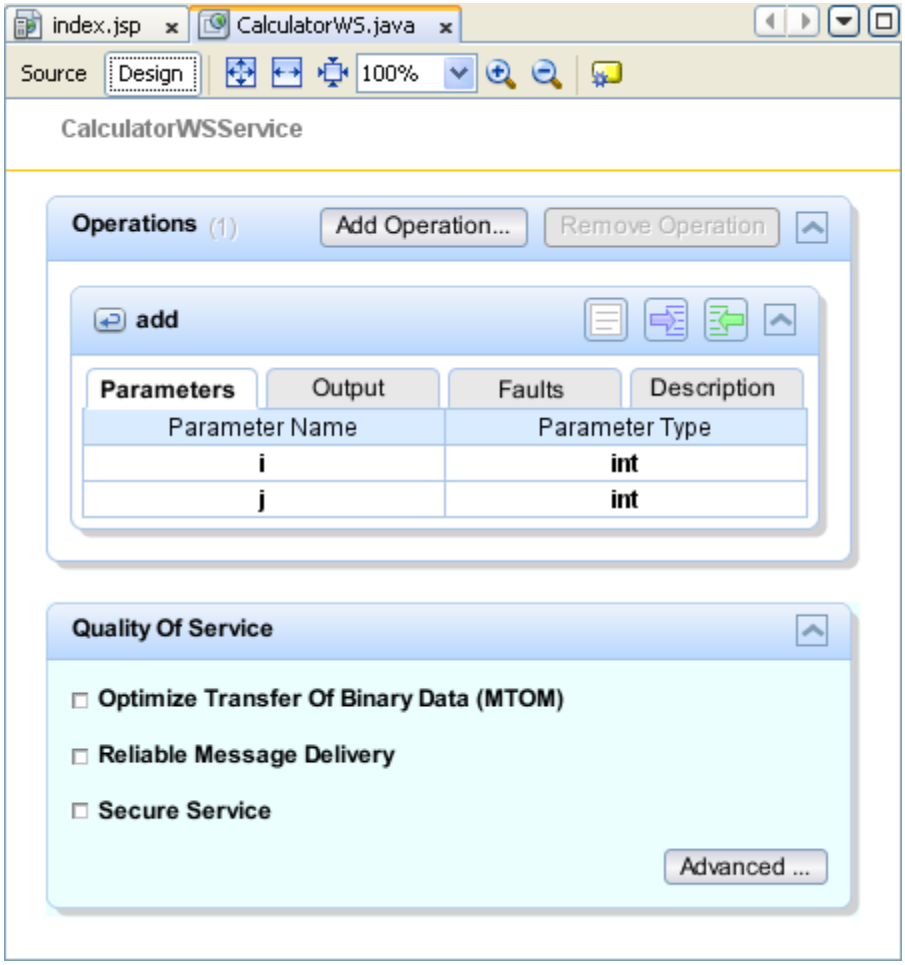
2. Click Add Operation in either the visual designer or the context menu.
3. The Add Operation dialog opens.
4. In the upper part of the Add Operation dialog box, type `add` in Name and type `int` in the Return Type drop-down list.
5. In the lower part of the Add Operation dialog box, click Add and create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

   You now see the following:



7. Click OK at the bottom of the Add Operation dialog box.
8. You return to the editor.
9. Remove the default `hello` operation, either by deleting the `hello()` method in the source code or by selecting the `hello` operation in the visual designer and clicking Remove Operation.

   The visual designer now displays the following:



10. Click Source and view the code that you generated in the previous steps.

```
package org.me.calculator;

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

/**
 *
 * @author gw152771
 */
@WebService()
public class CalculatorWS {

/**
     * Web service operation
     */
    @WebMethod(operationName = "add")
    public int add(@WebParam(name = "i")
    int i, @WebParam(name = "j")
    int j) {
//TODO write your implementation code here:
        return 0;
    }

}
```

11.

```
package org.me.calcuator;

import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;
import javax.ejb.Stateless;

/**
 *
 * @author jeff
 */
@WebService()
@Stateless()
public class CalculatorWS {

    /**
     * Web service operation
     */
    @WebMethod(operationName = "add")
    public int add(@WebParam(name = "i")
    int i, @WebParam(name = "j")
    int j) {
        //TODO write your implementation code
        return 0;
    }
}
```

12. In the editor, extend the skeleton `add` operation to the following (changes are in bold):

```
13.     @WebMethod
14.     public int add(@WebParam(name = "i") int i, @WebParam(name =
    "j") int j) {
15.         int k = i + j;
16.         return k;
        }
```

As you can see from the preceding code, the web service simply receives two numbers and then returns their sum. In the next section, you use the IDE to test the web service.

Deploying and Testing the Web Service

After you deploy a web service to a server,

To make the web service the entry point to your application, right-click the CalculatorWSApplication project node and choose Properties. Open the Run properties and type `/CalculatorWS` in the Relative URL field. Click OK. To run the project, right-click the project node again and select Run.

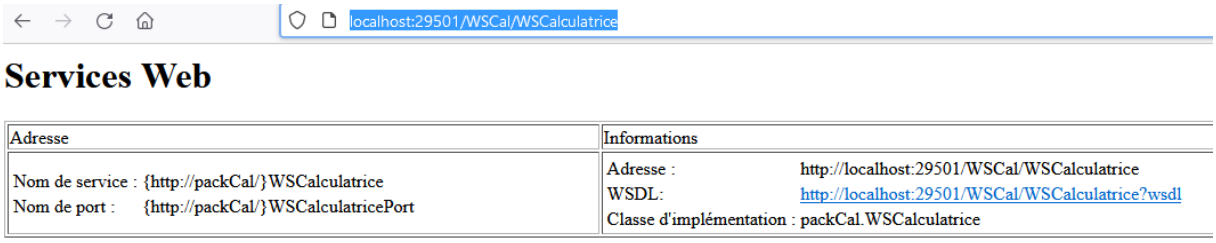
To test successful deployment to a GlassFish

Right-click the project and choose Deploy.

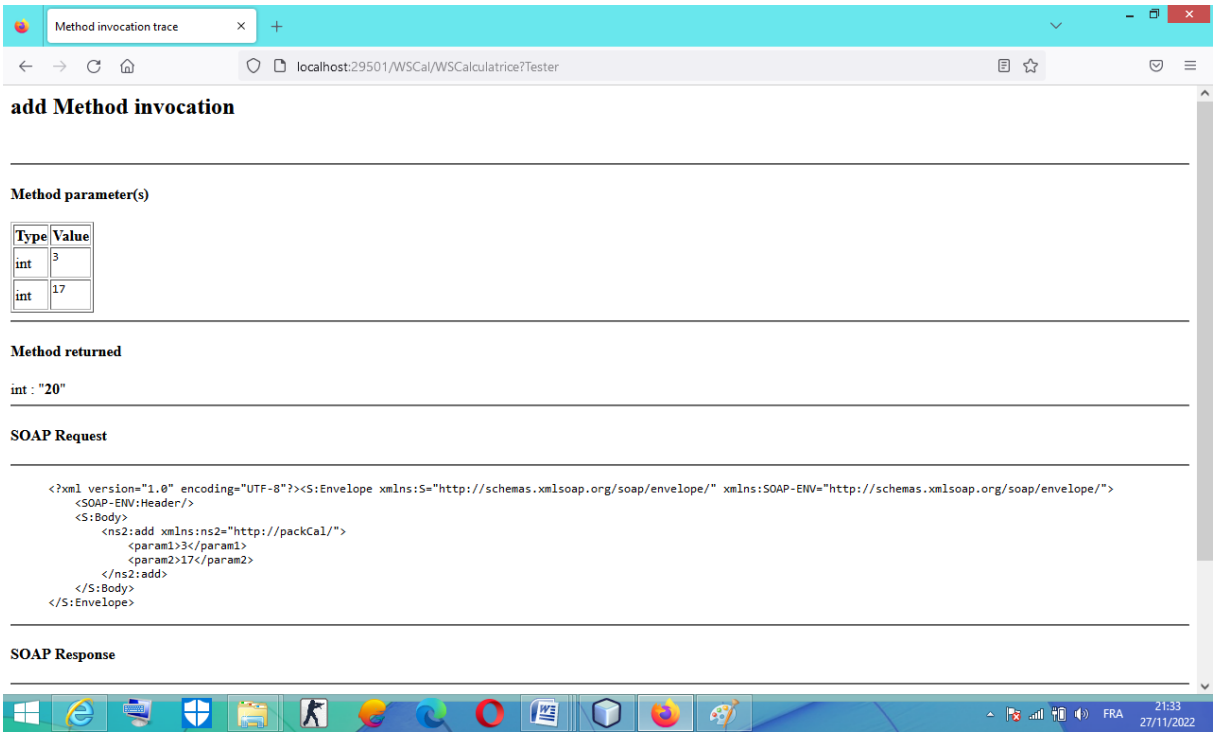
The IDE starts the application server, builds the application, and deploys the application to the server. You can follow the progress of these operations in the CalculatorWSApplication (run-deploy) and the GlassFish server or Tomcat tabs in the Output view.

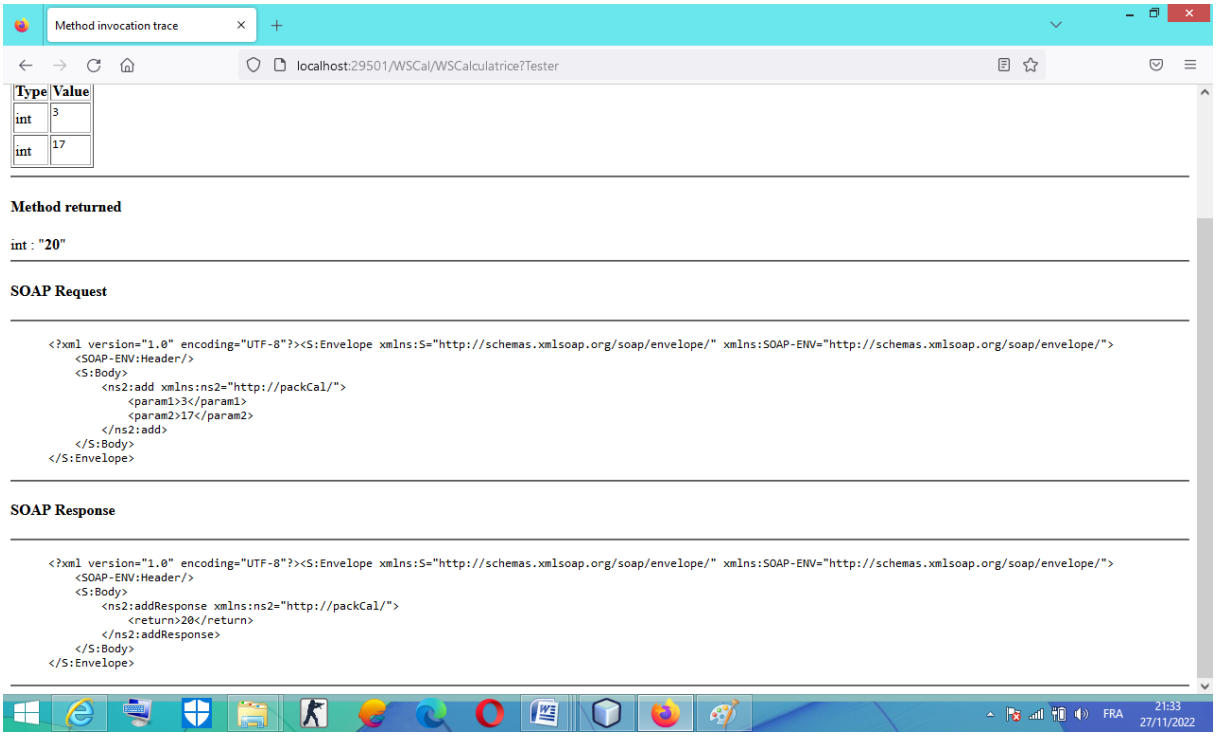
1. In the IDE's Projects tab, expand the Web Services node of the CalculatorWSApplication project.

http://localhost:29501/WSCal/WSCalculatrice

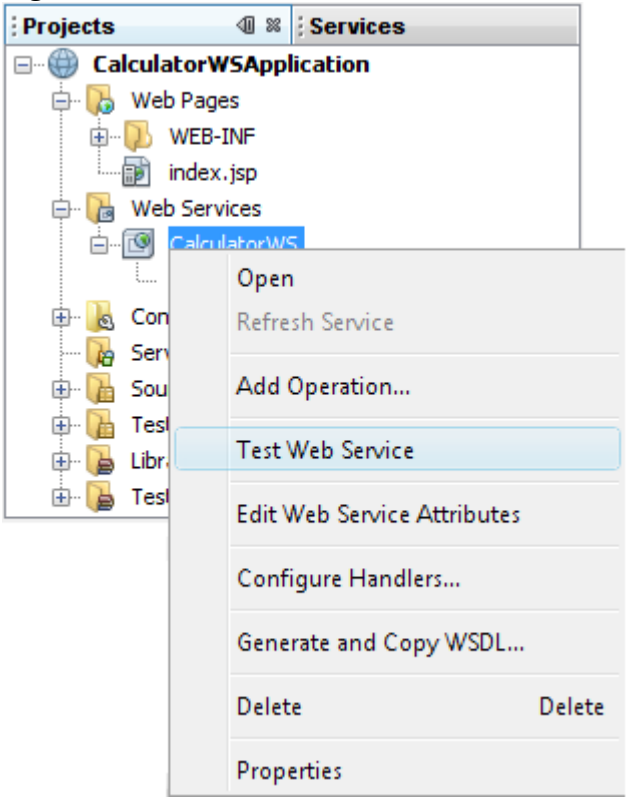


http://localhost:29501/WSCal/WSCalculatrice?Tester



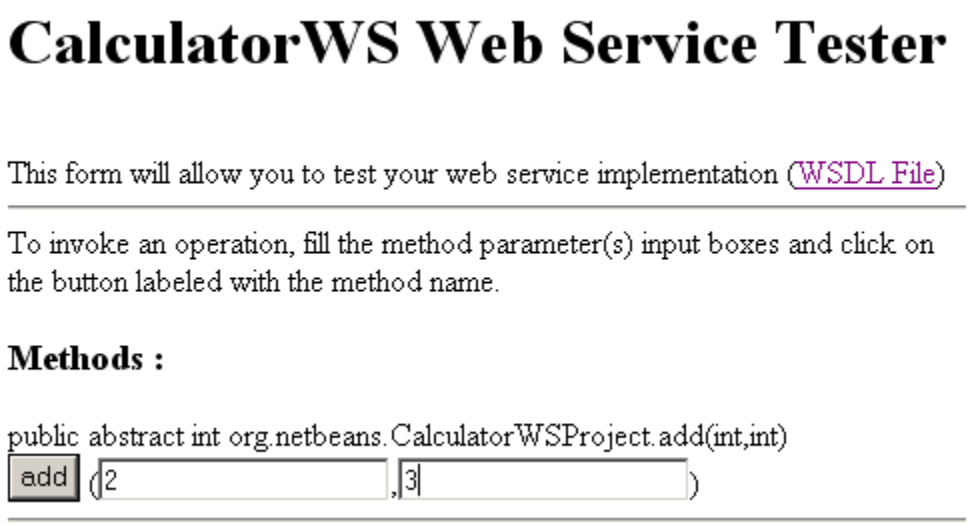


2. Right-click the CalculatorWS node, and choose Test Web Service.



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