Experiment 4

Aim:

To study and implement Web Services in SOAP for Java applications.

Theory:

Overview of Web Services:

Web services are application components that are designed to support interoperable machine-to-machine interaction over a network. This interoperability is gained through a set of XML-based open standards such as the Web Services Description Language (WSDL), the Simple Object Access Protocol (SOAP), and Universal Description Discovery and Integration (UDDI). These standards provide a common and interoperable approach for defining, publishing and using Web Services.

Choosing a Container:

You can either deploy your web service in a web container or in an EJB container. This depends on your choice of implementation. 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. For example, if you plan to deploy to the Tomcat Web Server, which inly has a web container, create a web application not an EJB module.

Choose File → New Project → Select Web Application from the Java Web category → Name the project Calculator WS Application → Select a location for the project → Click Next → Select your server and Java EE version → click Finish

Creating a Web Service from a Java Class:

Right-click the Calculator WS Application node and choose New → Web Service → Name the Web Service Calculator WS and type org.me.calculator in Package → Leave Create Web Service from scratch selected → If you are creating a Java EE Project on GlassFish or WebLogic, select Implement Web Service as a Stateless Session Bean → Click Finish → The project window displays the structure of the new web service and 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 wither in the web service visual designer or in the web service context menu.

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 box opens.
3. In the upper part of 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. click Add again and create a parameter of type int called j.
5. Click OK at the bottom of the Add Operation dialog box. You return to the editor.
6. 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.
7. Click Source and view the code that you generated in the previous steps. It differs whether you created the service as a Java EE stateless bean or not.

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. The GlassFish and WebLogic servers provide test clients.

If you are using the Tomcat Web Server, there is no test client. You can only run the project and see if the Tomcat Web Services page opens. In this case before you run the project you need to make the web service the entry point to your application. 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. Toi run the project right-click the project node again and select Run.

To test successful deployment to a GlassFish or WebLogic server:

Right-click the project and choose Deploy. The IDE starts the application server builds the application and deploys the app0lication to the server. You can follow the progress of these operations in the CalculatorWSApplication and the GlassFish server or Tomcat tabs in the Output view. In the IDE’s project tab expand the Web Services node of the CalculatorWSApplication project. Right-click the CalculatorWS node and choose Test Web Service. The IDE opens the tester page in your browser of you deploy a web application to the GlassFish server. For the Tomcat Web Server and deployment of EJB modules the situation if different. If you deploy to the GlassFish server type two numbers in the tester page.

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 app method. Here you create three clients- a Java class in a Java SE application, a servlet and a JSP page in a web application.

Conclusion:

Thus we have studied use of web services using SOAP for a java application.