

Lab – Introduction to Web Services

This lab will take you through the basics of using the Netbeans IDE to develop a JAX-WS Web service and to consume it with multiple clients. The clients that you will create are separate applications but all consume the same Web service.

Prerequisites

Before you begin, you will need to have the following installed on your computer:

- 1) NetBeans IDE version 6.0 or later
- 2) Java Standard Development Kit 6.0 or later (if not bundled with the IDE)
- 3) Sun Java System Application Server 9.0 or later (if not bundled with the IDE)

Configuring the Environment

If you have not registered an instance of the Sun Java System Application Server, you must do so before you can begin developing Java EE applications.

- 1) Choose Tools → Servers

Note: If a default server instance has already been created then you can skip the remaining steps. The server instances are listed in the *Servers* list.

- 1) Click Add Server. Select Sun Java System Application Server and give a name to the instance. Click Next.
- 2) Specify the server information, the location of the local instance of the application server, and the domain to which you want to deploy.
- 3) Click Finish.

Creating the Project

- 1) Choose File → New Project. Select Web Application from the Web category.
- 2) Name the project *TempConverterWebApp*.
- 3) Click Finish.

Creating the Web Service

- 1) Right-click on the TempConverterWebApp none in the projects window and choose New → Web Service...
- 2) Name the service TempConverterWS and specify any package name (i.e org.temp).

The Projects window displays the new Web service. For example, for web applications the Projects window now looks as follows:

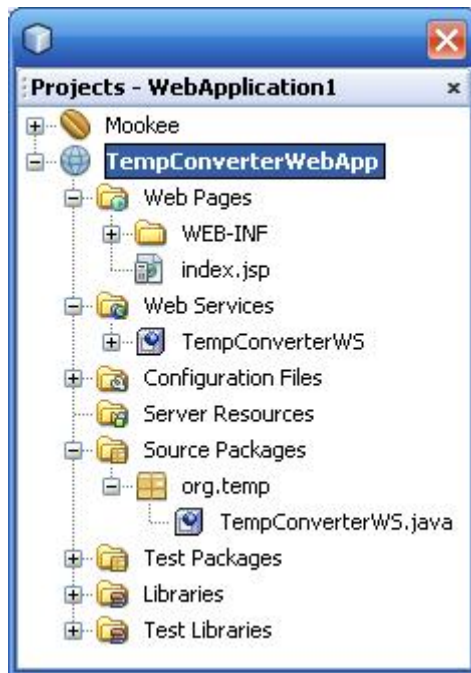


Figure 1 - Projects Window

Coding the Web Service

A Web service can have multiple methods. Add the following methods to the Web service:

- C2K (Celsius to Kelvin)
- C2F (Celsius to Fahrenheit)
- F2C (Fahrenheit to Celsius)
- F2K (Fahrenheit to Kelvin)
- K2C (Kelvin to Celsius)
- K2F (Kelvin to Fahrenheit)

Note: Use the *Design* view in the NetBeans IDE to add operations to the Web service.

With the methods created you must now complete the implementation (business logic) of the operations. Find the necessary algorithms needed to calculate the temperature conversions declared and implement the methods.

Go to the *Source* view of the *TempConverterWS* file and study the structure of a java Web service class and method. You'll notice that it differs from a typical java class and method with the inclusion of the `@WebService()`, `@WebMethod`, and `@WebParam()` identifiers. These identifiers tell the Java Runtime to interpret these components as a Web service and aid in describing their behaviour to the public.

Deploy and Test Web service

When you deploy a java Web service, the NetBeans IDE lets you test the Web service to see if it functions as you expect. The Tester application, provided by the Sun Java System Application Server, is integrated into the IDE for this purpose.

- 1) Right-click the *TempConverterWebApp* project node and click *Properties*.
- 2) Click on the *Run* option and uncheck *Display Browser on Run*.

By default, the browser is set to open a default resource. We will disable this behaviour.

- 3) Click OK
- 4) Right-click the *TempConverterWebApp* project node and click *Run*.

This will start the Java server and your Web application. You may notice that the default *index.jsp* is displayed in the Web browser.

- 5) Right-click *TempConverterWS* and choose *Test Web Service*.

This will load your Web service in the browser and will build a default GUI to enter parameters in order to test the service. Test your service and guarantee that it works as expected. Also, try providing values that may fail the service to make sure your algorithms are robust.

Note: If you attempt to test the Web service without starting the server first then the Web service will not run and you will be given an error stating so.

Consuming the Web Service through Java Application Class

- 1) Choose File → New Project. Select Java Application from the General category. Name the project *TempConverterAppClient*.

- 2) Right-click the *TempConverterClient1* and click New → Web Service Client.
- 3) Within the project, click Browse. Browse to the Web service that you want to consume. Click OK.

Note: You'll need to start the Web service server in order for the client to be able to discover the service.

- 4) Go to Main.java. Right-click the first line of the main method and click Web Service Client Resources → Call Web Service Operation
- 5) Browse to one of the temperature conversion operations under the TempConverterAPPCClient and click OK.

Note: NetBeans automatically generates the basic code needed to invoke the Web service.

- 6) Change the value of the input parameter in the code and run Main.java.

Consuming the Web Service through JSP

- 1) Choose File → New Project. Select Web Application from the Web category. Name the project *TempConverterJSPClient*.
- 2) Right-click the TempConverterJSPClient node and choose New → Web Service Client.
- 3) Within the project, click Browse. Browse to the Web service that you want to consume. Click OK.
- 4) Go to index.jsp. Right-click after the line “<h2>Hello World!</h2>” of the main method and click Web Service Client Resources → Call Web Service Operation
- 5) Browse to one of the temperature conversion operations under the TempConverterJSPClient node and click OK.

Note: NetBeans automatically generates the basic code needed to invoke the Web service.

- 6) Change the value of the input parameter in the code and run index.jsp.

Bonus

For extra marks, create a Web service client in a language other than a Java Application class. You could create a JSP (Java Server Page) or even an ASP.net client. What do you discover or learn when creating alternative clients.