

Web Services (using Java EE)

This tutorial will repeat last week's exercise of creating web services and web service clients, but this time we will use the Java EE "Metro stack" for Web Service development.

The tutorial relies on NetBeans. If you have your own laptop, you may wish to install NetBeans prior to the lab session. NetBeans can be downloaded at <http://www.netbeans.org/downloads/>. Preferably, you should download the Web & Java EE version of Netbeans. During installation, select the option to install the bundled Glassfish server together with the IDE.

You can also complete the tutorial using another IDE (e.g. Eclipse), but the instructions below are given for NetBeans.

Part A – Creating a Web Service Client to a Remote Web Service

- Use NetBeans to create a project of type "Java Application". Once a project is created, you can add various types of items to the project by right-clicking on the project's icon.
 - Add a new "Web Service Client ..." to the project. Set the Package to "net.terraservice" and the WSDL URL to `http://terraservice.net/TerraService2.asmx?WSDL`
 - In your main method, create an instance of the TerraService proxy class.
 - Call the `getTerraServiceSoap` method to access the TerraServiceSoap interface.
 - Create a Place object called NY and set its properties to "New York", "New York", "United States".
 - Invoke the `GetPlaceFacts` method of the service to retrieve a PlaceFacts object about New York.
 - Use `System.out.println` to write out the type of place that it is and the longitude and latitude its centre.
 - Compile and test your application.

- Invoke the `GetTileMetaFromLonLatPt` method of the service, passing the centre of New York, a theme of 1 and a scale of 8 metres.
 - If the tile metadata indicates that the tile exists, invoke the `GetTile` method of the service to retrieve the `tileId` contained in the metadata.
 - Create a new `java.io.FileOutputStream` to write to file name "tile.bmp".
 - Invoke the stream's `write` method to write the byte array to the file and then close the stream.
 - Compile and test your application.

Part B – Creating a Web Service (using WSDL first approach)

- Use NetBeans to create a project of type "Enterprise EJB Module" called `TemperatureConverter`
 - Add a new "Web Service from WSDL" with name `Converter` in Package `UT` using the WSDL file [ITempConverter.wsdl](#). Hint: When you right-click on the name of the project and then select "New", it may happen that you are not able to see the option "Web Service from WSDL". In this case, you need to select "Other", look for the menu "Web Services" and then find the option "Web Service from WSDL".
 - Implement the `CtoF` and `FtoC` methods:
 - $F = C * 9/5 + 32$, $C = (F - 32) * 5/9$
 - Compile/build your web service.

Part C – Deploying the Web Service to a Remote Server

- A Glassfish server is available for your use in the ATS server. For security reasons, the URL to the admin console, the login and the password are not made available in this tutorial handout. They will be given to you by the lab assistant. Also, this server can only be accessed within the UT network (you can use SSH tunnelling if outside the UT network). If you want to work at home, you should consider installing a Glassfish server locally in your laptop or machine at home.
- Once you have logged onto the admin console, follow the link “EJB Modules” and click Deploy...
 - Find the jar file in your local machine that contains the EJB module. Usually, this is located in the `dist` directory of the project. When the project is compiled, NetBeans will display the location of this directory.
 - In the “Application name”, please use the following naming convention: *yourLogin-TemperatureConverter*, where *yourLogin* is your UT user name. Note that the Glassfish server is being used by all ESI students. So it is important that you follow this naming convention when deploying Web services to this shared Glassfish server. Otherwise there will probably be name clashes with other students.
 - Once you have uploaded the jar file, browse to the “Web Services” link in the Glassfish admin console and click on your “Converter” service.
- Click on the Test button and test both of methods.