University of Westminster
Faculty of Science and Technology

Module code and title: 5COSC004W-Client Service Architecture Tutorial Manual

Tutorial title	**User-Defined Types and Exceptions in Web Services**
Tutorial type	**Guided and indepenent and non-marked**
Week 01	**18/03/2021**

Contents

Learning Goals 1

TASKS to be Performed under the instruction of the Tutor (from Task 1 to Task 19) 1

TASKS to BE PERFORMED Independently be the student (from Task 20 to Task 23) (Formative Assessment)..**Error! Bookmark not defined.**

Learning Goals

This tutorial focuses on preparing for the Second In Class Test (ICT2)
You can refer to Tutorials 3 to 5 to review the implementation steps.

TASKS to be Performed

1) Start Netbeans (8.0.2 EE or *.2 EE) in your system.
2) Question 1 – Create the Server (15 points)
   a. Create a new Web Application project in Netbeans (Figure 1 to 3). You are free to select any reasonable name, in this document, we will call this ICT2WebApplication.
   b. Create a new Web Application project in Netbeans (Figure 1 to 3). You are free to select any reasonable name, in this document, we will call this ICT2WebApplication.
   c. Create a new Java package in the Web Application where we will put our server code. You are free to select any reasonable name, , in this document, we will call this server.
   d. Now, we can create a proper Web Service in the Project of Step (2.a). You are free to select any reasonable name, You are free to select any reasonable name, , in this document, we will call this ICT2WebService.

   e. If you want, you can remove the WebMethod hello. We will not use it in the tutorial.
   f. Add a WebMethod called isConnected that takes no parameters and return the String “Connected”.
   g. Test the Web Service with the Test Web Service tool of NetBeans
   h. Export the Project as a Zip File.

3) Question 2 – Create the Client (15 points)
   a. Create a new Web Application project in Netbeans (Figure 1 to 3). You are free to select any reasonable name, in this document, we will call this ICT2Client.
   b. Create the client stub which connects to the Web Service you have created in step (2.d)
   c. Write the Client code to create an instance of the ICT2Client class, define and execute the execute() method.
   d. Drag and drop the client stub for the isConnected Web Method into the client code.
   e. Inside the execute() method, invoke the isConnected client stubs to check if the Server is connected.
   f. Export the Project as a Zip File.

4) Question 3 – Simple Method (15 points)
   a. On the Server, implement a Web Method called getLongestStringBetween(String a, String b) which returns the longest of the two strings passed as Web Parameters.
   b. Test the Web Service with the Test Web Service tool of NetBeans
   c. Export the Server Project as a Zip File.
   d. On the Client, refresh the Web Service reference.
   e. Drag and Drop the client stub for the getLongestStringBetween Web Method into the client code.
   f. Test the getLongestStringBetween Web Method with strings of equal and different length.
   g. Export the Client Project as a Zip File.
5) Question 4 – Exceptions (15 points)
   a. On the Server, implement a Web Method called getLongestStringBetweenWithException(String a, String b) which returns the longest of the two strings passed as Web Parameters. The Web Method throws an Exception when one or both the passed parameters are null.
   b. Test the Web Service with the Test Web Service tool of NetBeans

c. Export the Server Project as a Zip File.
d. On the Client, refresh the Web Service reference.
e. Drag and Drop the client stub for the getLongestStringBetweenWithException Web Method into the client code.
f. Test the getLongestStringBetweenWithException Web Method with strings of equal and different length and one or both null parameters.
g. Export the Client Project as a Zip File.

6) Question 5 – Related Methods (20 points)

a. On the Server, implement a Web Method called addSampleOnServer(String s) which adds the passed String to the Server. You are free to use any type to store the Strings but ArrayList is a good choice.
b. Test the Web Service with the Test Web Service tool of NetBeans.
c. On the Server, implement a Web Method called getLongestStringOnServer() which returns the longest string added to the Server.
d. Export the Server Project as a Zip File.
e. On the Client, refresh the Web Service reference.
f. Drag and Drop the client stub for the addSampleOnServer and getLongestStringOnServer() Web Methods into the client code.
g. Test the two Web Method with strings of different lengths.
h. Export the Client Project as a Zip File.