

# Web Services using Tomcat and Eclipse

Nauman

recluze@gmail.com

Security Engineering Research Group

Institute of Management Sciences

Peshawar, Pakistan

<http://recluze.wordpress.com>

<http://serg.imsciences.edu.pk>

Assumptions:

1. You're using Windows. If you're using Linux, you don't need this tutorial. You can find enough stuff on the Internet to get going.
2. You have downloaded Apache Tomcat (<http://jakarta.apache.org>), Axis (<http://ws.apache.org/axis>) and Eclipse Europa (<http://www.eclipse.org>)
3. You've installed the java development kit (<http://java.sun.com>)
4. You're familiar with Java language and the Windows operating system.

## Part I: Setting up the server

1. To use web services with apache, first you need to install Tomcat. We'll assume you installed it in c:\tomcat. We'll use <tomcat\_home> for this directory.
2. Extract the Axis binary archive to <tomcat\_home>. You should get axis folder in <tomcat\_home>\webapps\axis
3. Copy tools.jar from <jdk\_home>\lib to <tomcat\_home>/common/lib
4. Set the following System Environment Variables (from Control Panel -> System -> Advanced -> Environment Variables):

```
AXIS_HOME: c:\tomcat
AXIS_LIB: %AXIS_HOME%\lib
AXISCLASSPATH: %AXIS_LIB%\axis.jar;%AXIS_LIB%\commons-
discovery.jar;%AXIS_LIB%\commons-
logging.jar;%AXIS_LIB%\jaxrpc.jar;%AXIS_LIB%\saa.jar;%AXIS_LIB%\log4j-
1.2.8.jar;%AXIS_LIB%\xml-apis.jar;%AXIS_LIB%\xercesImpl.jar
```

(Also make sure your CLASSPATH variable is set to include java bin and lib directories)

```
CLASSPATH: .;c:\Program Files\Java\jdk1.6.0_02\bin;c:\Program
Files\Java\jdk1.6.0_02\lib;%AXISCLASSPATH%;C:\Program
Files\Java\jre1.6.0_02\lib\ext\QTJava.zip
```

You may find it useful to copy this text to notepad and set it according to your own installation directories.

5. You're all set up to use web services! Now, let's create a simple Java Web Service.
6. Create a new java file called BasicMath.java with the following code:

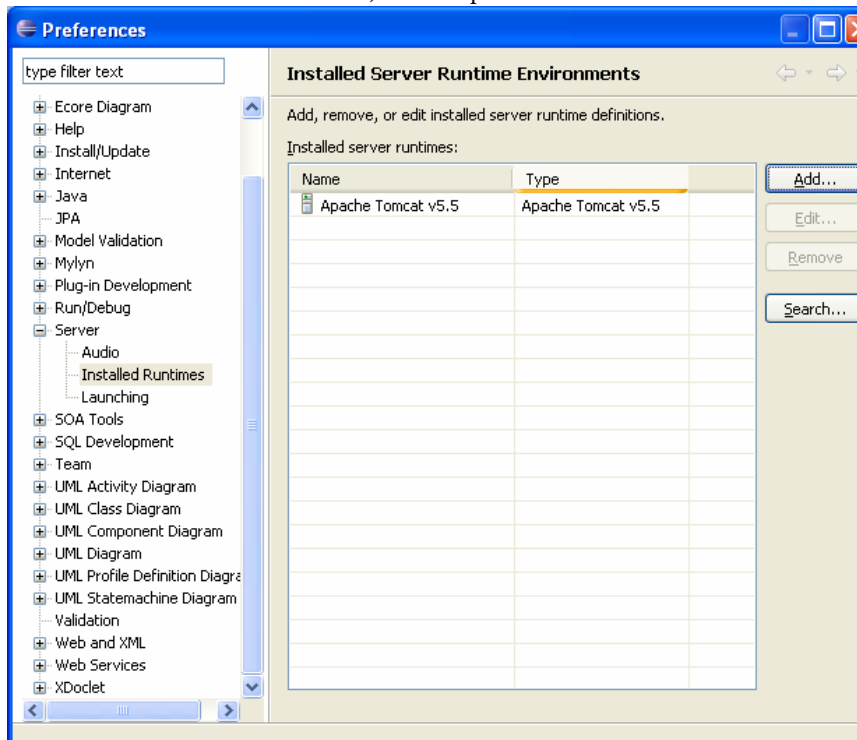
```
public class BasicMath {
    public int add(int n1, int n2){
        return n1+n2;
    }
}
```

```
}  
}
```

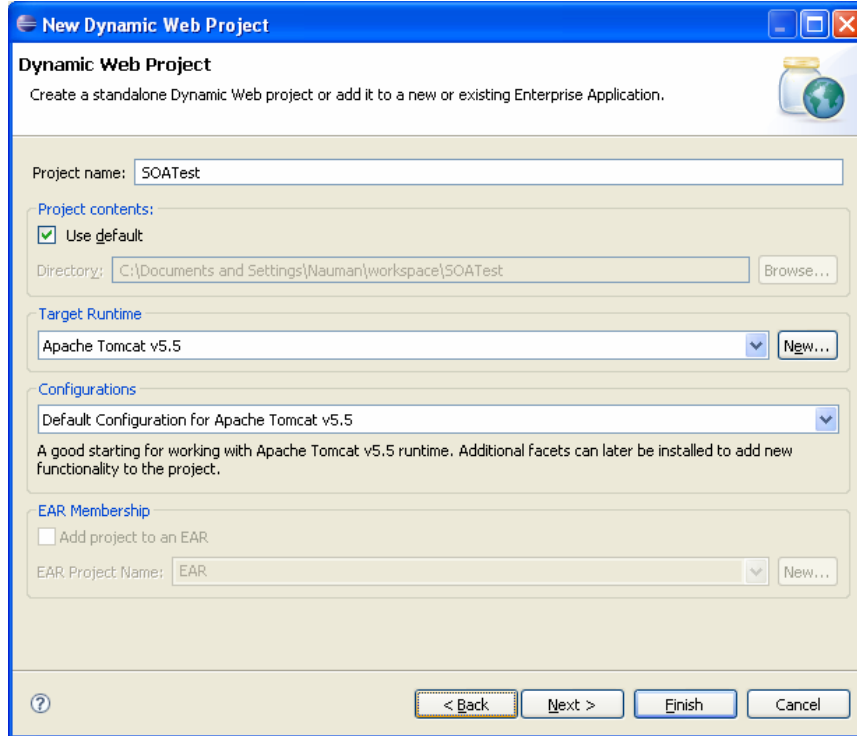
7. Save it as a JWS file in <tomcat\_home>/axis/BasicMath.jws
8. Start apache server using Apache Monitor from the program menu.
9. Call the service from <http://localhost:8080/axis/BasicMath.jws>
10. And look at the wsdl on:  
<http://localhost:8080/axis/BasicMath.jws?wsdl>
11. You can also call the method using:  
<http://localhost:8080/axis/BasicMath.jws?method=add&n1=4&n2=3>

## Part II: Using Eclipse

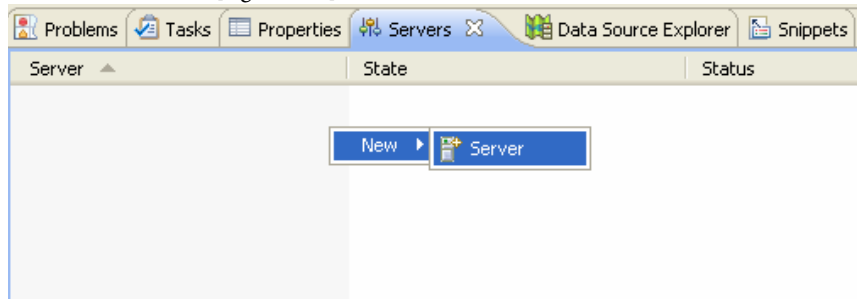
1. Install Eclipse
2. If you downloaded the default Eclipse Europa package, it does not include Web Tools Platform (WTP). You need to Help->Software Updates->Find and Install. Alternatively, you can download the MDT and WTP bundle provided by the Security Engineering Research Group at <http://serg.imsciences.edu.pk/pub/software/eclipse3.3-win32-wtp-mdt.rar>
3. Start Eclipse and setup a new installed runtime (Windows->Preferences->Server->Installed Runtimes->Add). Select Apache Tomcat 5.5 as the new runtime.



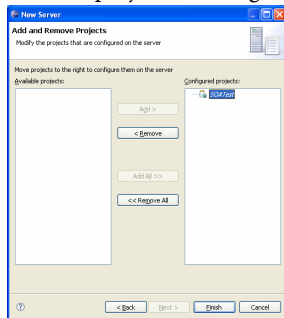
4. Create a new “Dynamic Web Project” from the “Web” node with the following settings.



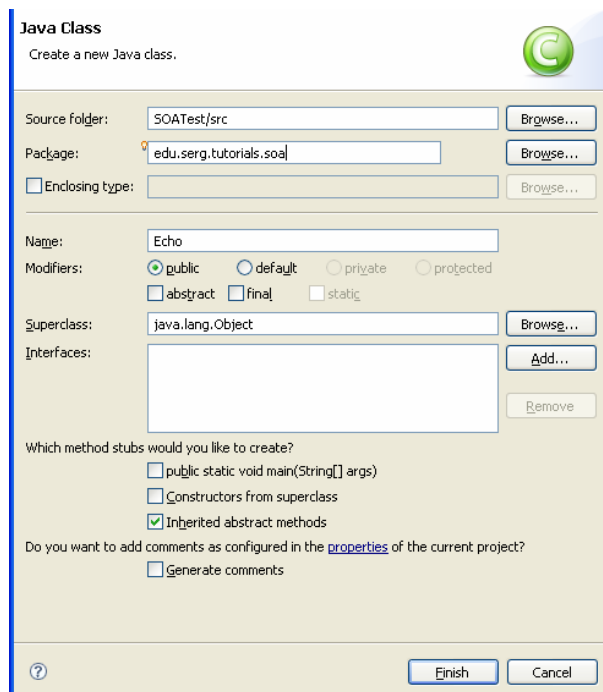
5. In the Servers View, [right-click] New->Server.



6. Add the project to configured projects. Then, right click the server and start it.



7. In the project explorer, add a new class file called Echo in package `edu.serg.tutorials.soa`:



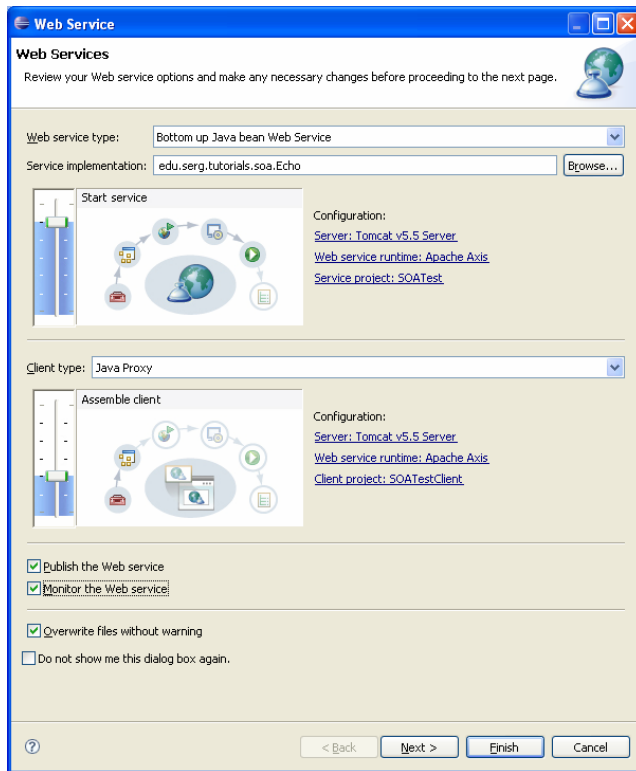
8. Enter the following code for the class:  
`package edu.serg.tutorials.soa;`

```
public class Echo {
    public int[] getNums(int n, int m) {
        int a[] = new int[10];

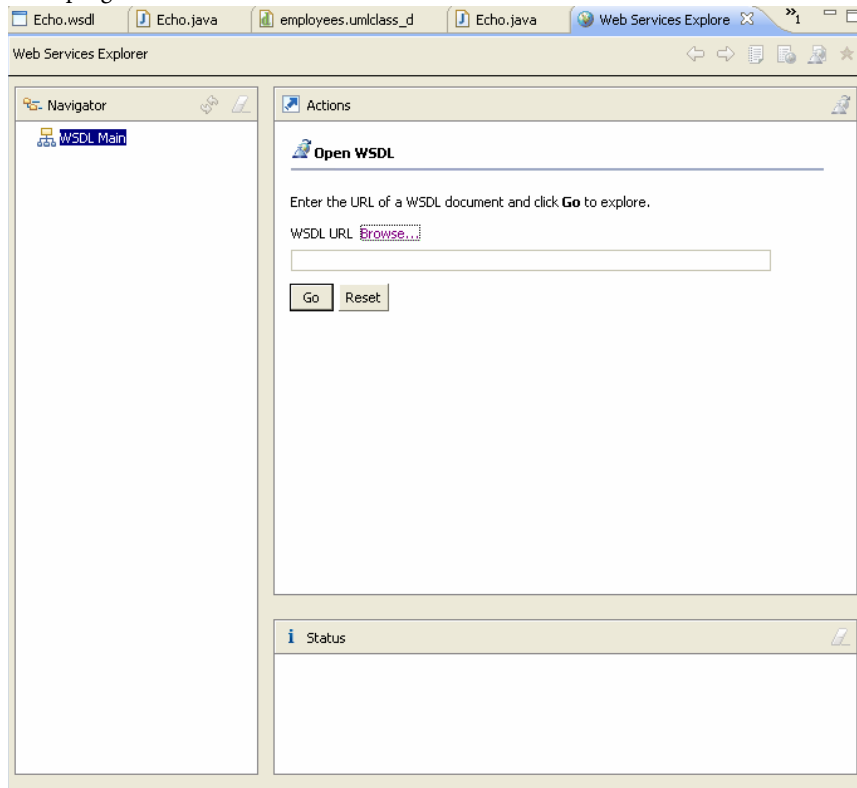
        for (int i=0; i < 10; i++){
            a[i] = m * n * i;
        }
        return a;
    }
    public int[] getNums2(int n, int m) {
        int a[] = new int[10];

        for (int i=0; i < 10; i++){
            a[i] = m * n * i * 2;
        }
        return a;
    }
}
```

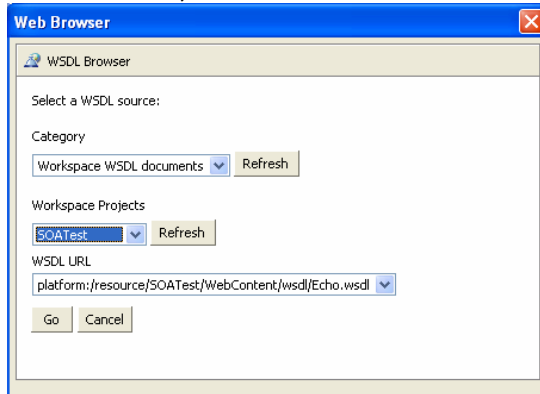
9. Build the Project. Then Right-Click Echo.java in project explorer and select Web Services -> Create Web Service. Select the following options (Make sure you set the sliders as in the screenshot.) You can select the default settings in the following dialogs.



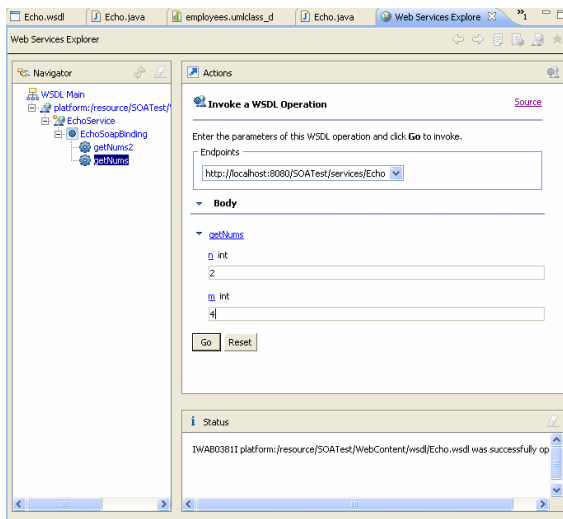
10. Run-> Launch Web Services Explorer. Select **WSDL Page** from the buttons on the top right of the view.



11. Select WSDL Main from the Navigator. Then click Browse. Select Category “Workspace Documents”, Project “SOATest” and hit Go.



12. Hit Go in the WSDL Main page and you get a list of operations. Click on getNums ( ) . Enter values for variables n and m. Hit Go.



13. In the Status pane, see the result. Hit Source and take a look at SOAP request and response.
14. Notice also that the web service creation procedure has created a SOATestClient project for you. If you don't understand why, see the client slider in Step 9 above. You asked eclipse to Assemble a client for you. Anyway, let's open this client project and write a java client.
15. In SOATestClient project, create a file EchoClient.java in the src folder. Enter the following code for it:

```
package edu.serg.tutorials.soa;
import org.apache.axis.AxisFault;
import java.net.URL;

public class EchoClient {
    public static void main(String[] args) throws AxisFault ,Exception {
        EchoSoapBindingStub srv = new EchoSoapBindingStub(
            new URL("http://localhost:8080/SOATest/services/Echo"),
            new EchoServiceLocator());
        int a[] = srv.getNums(3, 4);
        for(int x:a)
            System.out.println(x);
    }
}
```

16. Run the client and see the results.

You may also test the client using the browser at:

<http://localhost:8080/SOATest/services/Echo>

Assuming, of course, that your server is running at port 8080.

Feel free to contact me if you need help with this stuff.