**How to create and consume a simple Web Service using JAX WS**

One of the most exciting new features of the Java Platform, Standard Edition 6 (Java SE 6) is support for the Java API for XML Web Services (JAX-WS), version 2.0. JAX-WS 2.0 is the centre of a redesigned API stack for web services, which also includes Java Architecture for XML Binding (JAXB) 2.0 and SOAP with Attachments API for Java (SAAJ) 1.3. Even though JAX-WS is mainly part of Java EE Platform, we can use many of the functionalities without the need of Java EE Application Server. JAX-WS architecture is an easier-to-understand architecture for web services development.

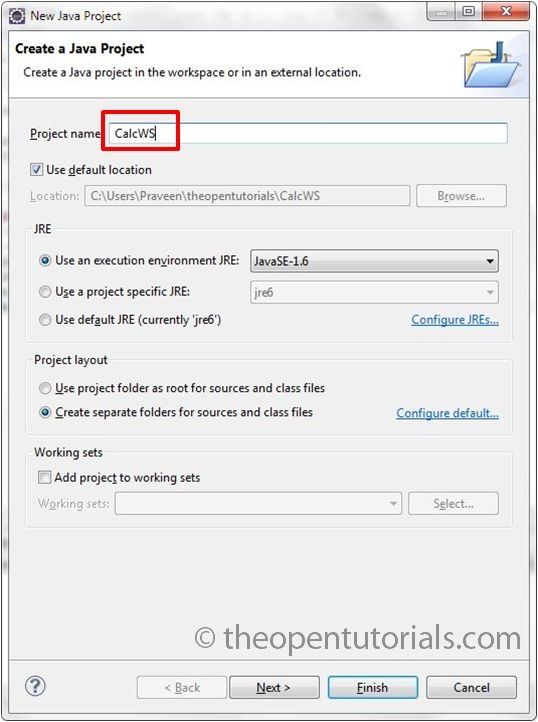
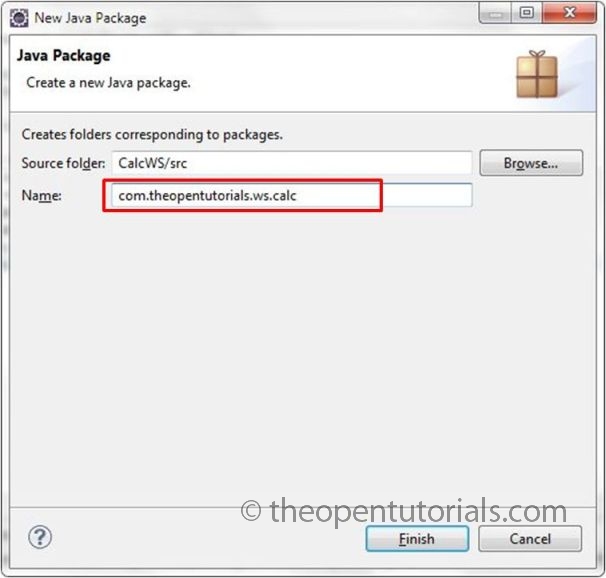
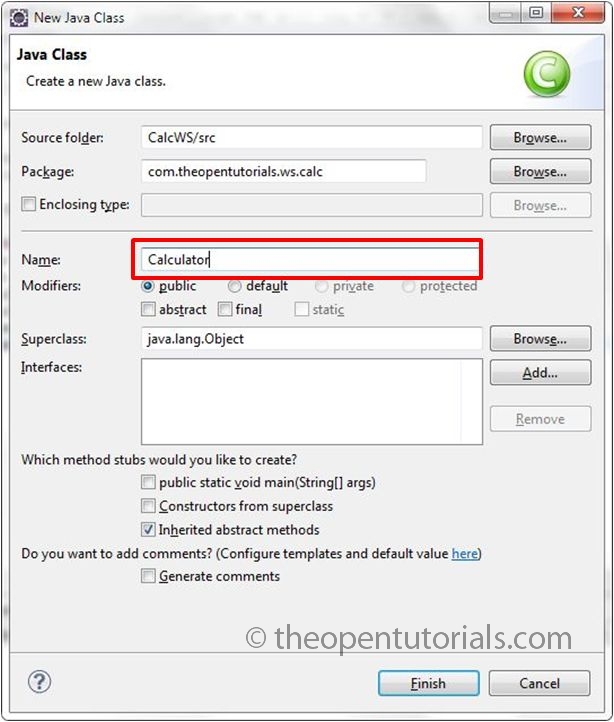
**Project Description**

* In this example, we create a SOAP based web service for a simple Java Calculator class with operations ‘add’ and ‘subtract’.
* We then create a web service client which then consumes the web service and displays the result of the invoked web service.

**Environment Used**

* Java SE 6
* JAX WS 2.x
* Eclipse IDE

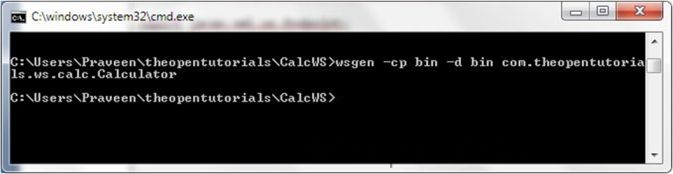
**Creating and Publishing Web Service**

1. Create a Java Project ‘CalcWS’.  
   [](http://theopentutorials.com/wp-content/uploads/how-to-create-and-consume-a-simple-web-service-using-jax-ws_2521/1-calcws-pjt.jpg?ef6f3c)
2. Create a package ‘com.theopentutorials.ws.calc’.  
   [](http://theopentutorials.com/wp-content/uploads/how-to-create-and-consume-a-simple-web-service-using-jax-ws_2521/2-calcws-calc-pkg.jpg?ef6f3c)
3. Create a Java class ‘Calculator’ and type the following code.  
   [](http://theopentutorials.com/wp-content/uploads/how-to-create-and-consume-a-simple-web-service-using-jax-ws_2521/3-calcws-calc-class.jpg?ef6f3c)

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14 | package com.theopentutorials.ws.calc;  import javax.jws.WebService;    @WebService  public class Calculator {      public int add(int a, int b) {          return (a + b);      }        public int sub(int a, int b) {          return (a - b);      }    } |

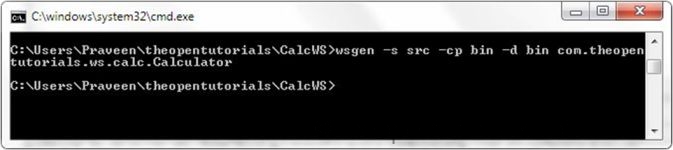
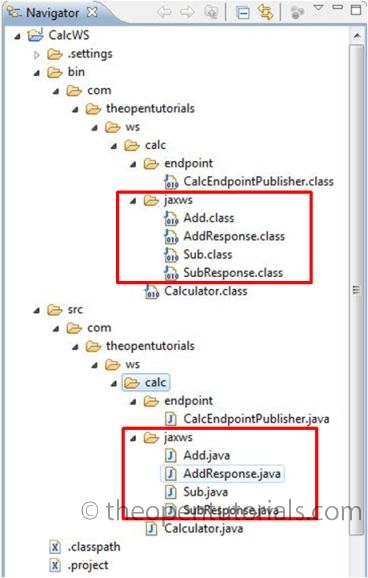
1. @WebService annotation at the beginning of the class definition tells the Java interpreter that we intend to publish ALL the methods of this class as a web service. If we want to publish only particular methods then we can use @WebMethod annotation before the method signature.
2. In order to publish our class and its methods as web service we need to crate appropriate stub files or artifacts for web service deployment and invocation. Fortunately Java provides a tool called ‘wsgen’ which generates JAX-WS portable artifacts used in JAX-WS web services.
3. Open command prompt or terminal in Linux and go to the project folder ‘CalcWS’.
4. Now issue the following command,

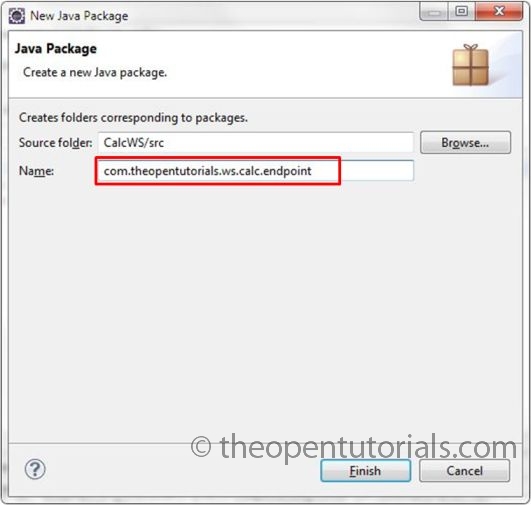
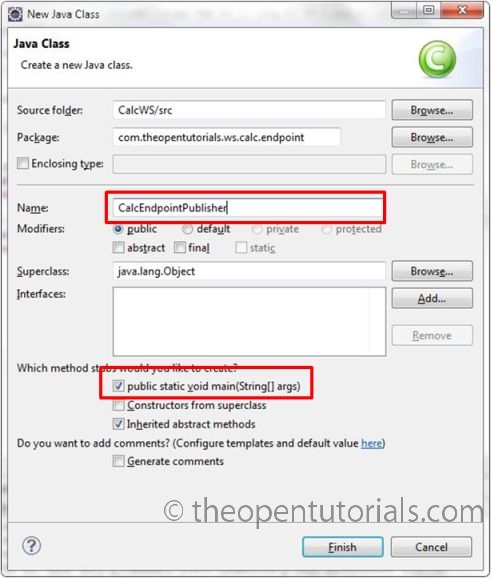
wsgen -cp bin -d bin com.theopentutorials.ws.calc.Calculator

the –cp option specifies the classpath for our Calculator class which is in ‘bin’ folder, the –d option specifies where to place generated output files which is also the ‘bin’ folder in our case.  
[](http://theopentutorials.com/wp-content/uploads/how-to-create-and-consume-a-simple-web-service-using-jax-ws_2521/4-wsgen.jpg?ef6f3c)

1. We can also have a look at the source of the generated files by using the –s option provided by ‘wsgen’.

wsgen -s src -cp bin -d bin com.theopentutorials.ws.calc.Calculator

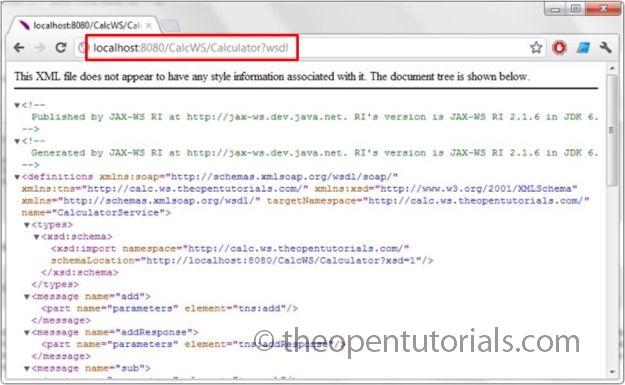
[](http://theopentutorials.com/wp-content/uploads/how-to-create-and-consume-a-simple-web-service-using-jax-ws_2521/5-wsgen-src.jpg?ef6f3c)  
[](http://theopentutorials.com/wp-content/uploads/how-to-create-and-consume-a-simple-web-service-using-jax-ws_2521/6-wsgen-output.jpg?ef6f3c)

1. Now we need to publish our class as a web service endpoint. For that we use the static publish() method of the javax.xml.ws.Endpoint class to publish our ‘Calculator’ class as a web service in the specified context root.
2. Create a package ‘com.theopentutorials.ws.calc.endpoint’.  
   [](http://theopentutorials.com/wp-content/uploads/how-to-create-and-consume-a-simple-web-service-using-jax-ws_2521/7-calcws-endpt-pkg.jpg?ef6f3c)
3. Create a class ‘CalcEndpointPublisher’ with main method and type the following code.  
   [](http://theopentutorials.com/wp-content/uploads/how-to-create-and-consume-a-simple-web-service-using-jax-ws_2521/8-calcws-publisher-class.jpg?ef6f3c)

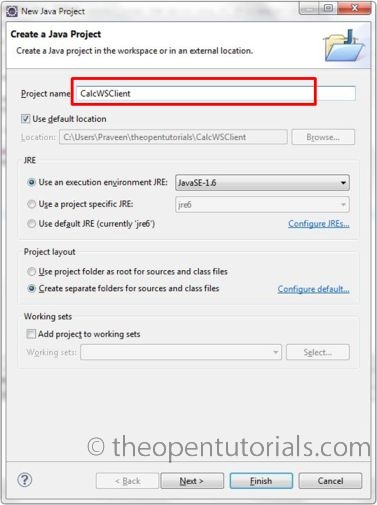
|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13 | package com.theopentutorials.ws.calc.endpoint;    import javax.xml.ws.Endpoint;  import com.theopentutorials.ws.calc.Calculator;    public class CalcEndpointPublisher {        public static void main(String[] args) {          Endpoint.publish("<http://localhost:8080/CalcWS/Calculator>",                          new Calculator());      }    } |

1. Run this class as ‘Java Application’.
2. You may not get output in the Console. To check whether our class is published as web service, open a browser and type the URL mentioned in the endpoint with a parameter ?wsdl appended.

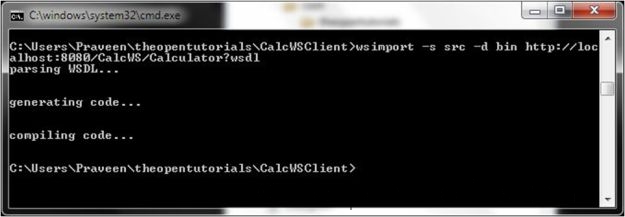
http://localhost:8080/CalcWS/Calculator?wsdl

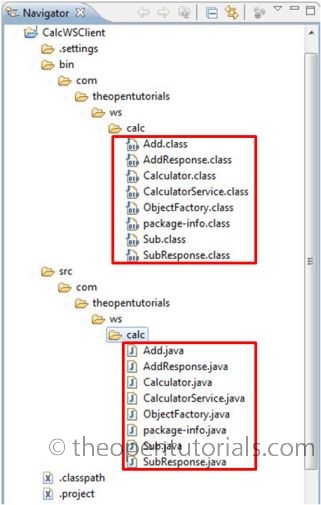
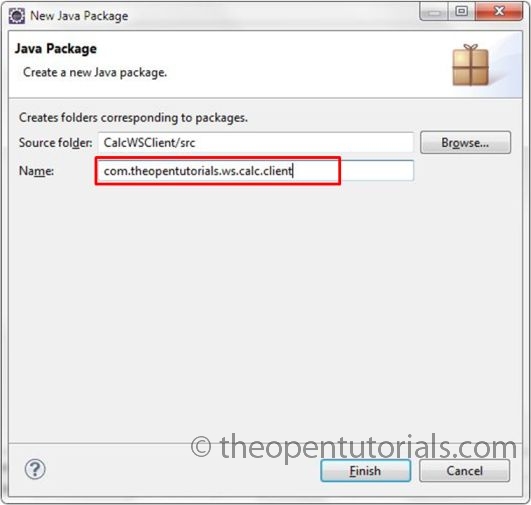
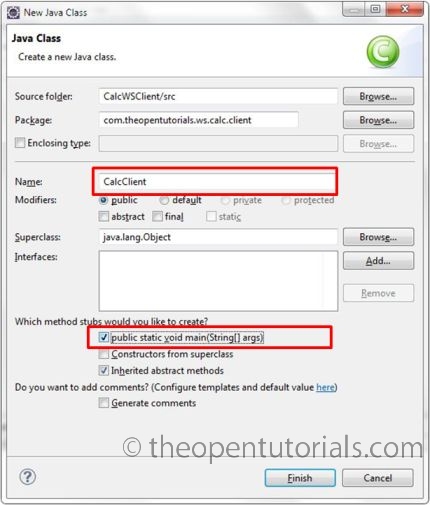
1. When you run the application, the Java SE 6 platform has a small web application server that will publish the web service at the address http://localhost:8080/CalcWS/Calculator while the JVM is running.
2. If you see a large amount of XML that describes the functionality behind the web service, then the deployment is successful.  
   [](http://theopentutorials.com/wp-content/uploads/how-to-create-and-consume-a-simple-web-service-using-jax-ws_2521/9-wsdl-browser-output.jpg?ef6f3c)

**Creating and consuming a Web Service Client**

1. Having published the web service, we now create a client which communicates with the service and displays the result.
2. Create a Java project ‘CalcWSClient’.  
   [](http://theopentutorials.com/wp-content/uploads/how-to-create-and-consume-a-simple-web-service-using-jax-ws_2521/10-calcws-client-pjt.jpg?ef6f3c)
3. Just like ‘wsgen’, JAX-WS also provides a tool called ‘wsimport’ for generating the artifacts required for creating and consuming a web service. ‘wsimport’ takes a wsdl file as input.
4. From the project folder in command prompt or terminal, issue the following command,

wsimport -s src -d bin http://localhost:8080/CalcWS/Calculator?wsdl

[](http://theopentutorials.com/wp-content/uploads/how-to-create-and-consume-a-simple-web-service-using-jax-ws_2521/11-wsimport.jpg?ef6f3c)

1. This will generate many files as shown in the below file hierarchy tree.  
   [](http://theopentutorials.com/wp-content/uploads/how-to-create-and-consume-a-simple-web-service-using-jax-ws_2521/12-wsimport-output.jpg?ef6f3c)
2. Let us now create a Java class with main method to run this client. Create a package ‘com.theopentutorials.ws.calc.client’  
   [](http://theopentutorials.com/wp-content/uploads/how-to-create-and-consume-a-simple-web-service-using-jax-ws_2521/13-calcws-client-pkg.jpg?ef6f3c)
3. In that package, create a class ‘CalcClient’ with main method and type the following code.  
   [](http://theopentutorials.com/wp-content/uploads/how-to-create-and-consume-a-simple-web-service-using-jax-ws_2521/14-calcws-client-class.jpg?ef6f3c)

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16 | package com.theopentutorials.ws.calc.client;    import com.theopentutorials.ws.calc.Calculator;  import com.theopentutorials.ws.calc.CalculatorService;    public class CalcClient {        public static void main(String[] args) {          int a = 10;          int b = 12;          CalculatorService calcService = new CalculatorService();          Calculator calc = calcService.getCalculatorPort();          System.out.println(a + " + " + b + " = " + calc.add(a, b));          System.out.println(a + " - " + b + " = " + calc.sub(a, b));      }  } |

1. Run this class as Java Application.
2. You will get the following output in the console.

10 + 12 = 22  
10 – 12 = -2

1. Make sure that the ‘CalcEndpointPublisher’ class is run before running this client. Because we need to publish the web service before running the client. Otherwise you may get the following exception on running the client.

Exception in thread “main” javax.xml.ws.WebServiceException: Failed to access the WSDL at: http://localhost:8080/CalcWS/Calculator?wsdl. It failed with:  
Connection refused: connect.

1. To stop the ‘CalcEndpointPublisher’ class, select that particular console and use the terminate button (red button) in the eclipse console tab bar.  
   [](http://theopentutorials.com/wp-content/uploads/how-to-create-and-consume-a-simple-web-service-using-jax-ws_2521/15-calcws-terminate.jpg?ef6f3c)