**Web services are open standard ( XML, SOAP, HTTP etc.) based Web applications that interact with other web applications for the purpose of exchanging data**

**Web Services can convert your existing applications into Web-applications.**

Web Services is a Distributed Technology achieves Interoperability (portability) across different technologies such as a JAVA, .NET, PHP, C++ etc.

The other distributed technologies like CORBA, RMI, EJB, DCOM etc uses Binary format, hence the objects are serialized into stream of bytes.

Web Services uses XML format instead of Binary Format. Hence in web services the data is exchanged as XML message, which is validated against XML Schema (XSD) but nor DTD, since XSD additionally supports data types.

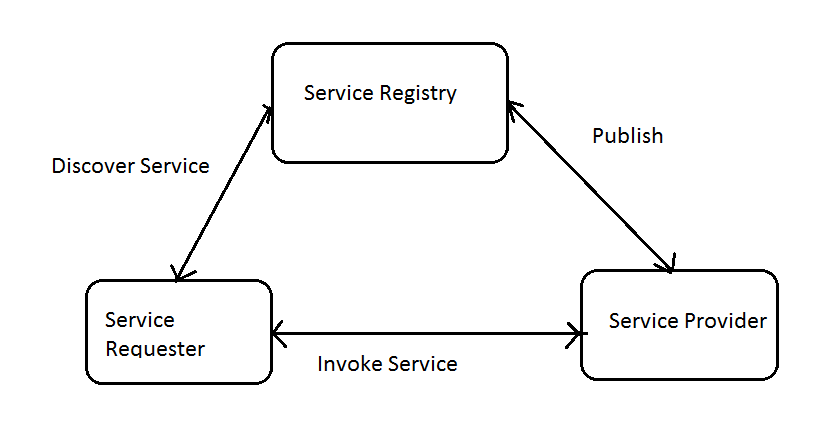
In Web services, the objects are serialized into XML messages.

**Components of Web Services?**

The basic Web services platform is XML + HTTP. All the standard Web Services works using following components

* SOAP (Simple Object Access Protocol)
* UDDI (Universal Description, Discovery and Integration)
* WSDL (Web Services Description Language)

**Architecture**



There are three major roles within the web service architecture:

* **Service provider:**

This is the provider of the web service. The service provider implements the service and makes it available on the Internet.

* **Service requestor**

This is any consumer of the web service. The requestor utilizes an existing web service by opening a network connection and sending an XML request.

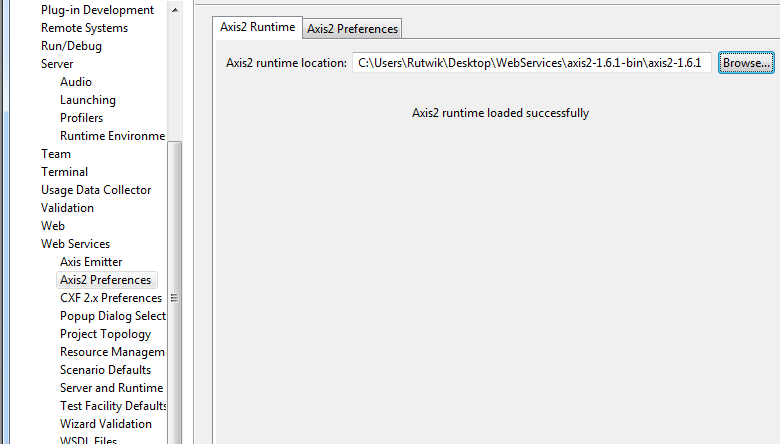
* **Service registry**

This is a logically centralized directory of services. The registry provides a central place where developers can publish new services or find existing ones. It therefore serves as a centralized clearinghouse for companies and their services.

**Sample Application using AXIS-2 , Tomcat**

Configure the Tomcat 7 Server in the work space.

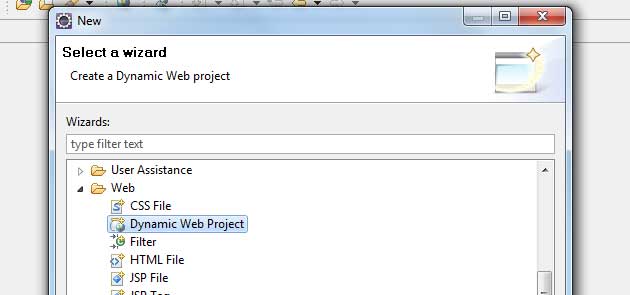
In the Window-> Preferences window select Web Service –-> Axis2 Preferences and browse the top level directory of Apache Axis2 Binary Distribution



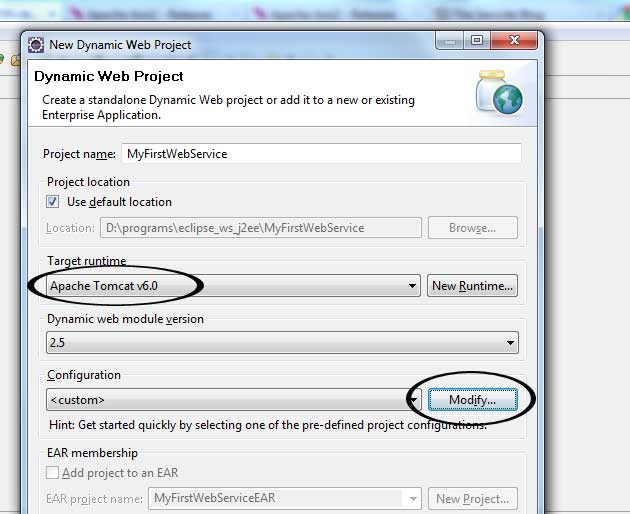
Click OK.

### Creating the Web Service Using Bottom-Up Approach

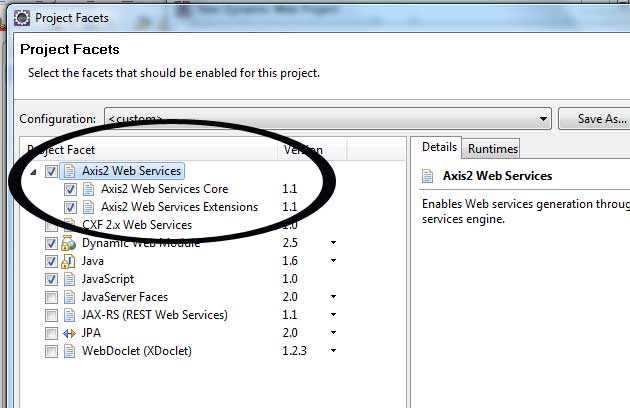
2.1 First create a new Dynamic Web Project (File --> New –-> Other…) and choose Web --> Dynamic Web Project.



2.2 Set Apache Tomcat as the Target Runtime and click Modify to install Axis2 Web Services project facet.



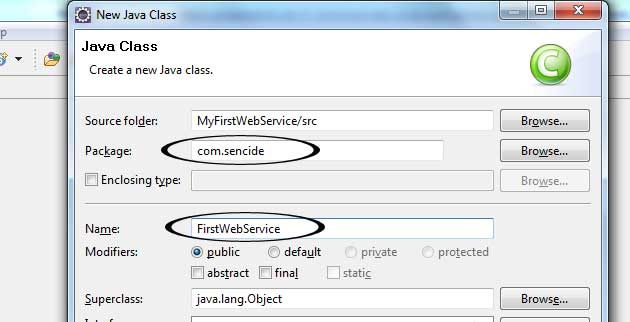
2.3 Select Axis2 Web Services



2.4 Click OK and then Next. There you can choose folders and click Finish when you are done.

### 3. Create Web Service Class

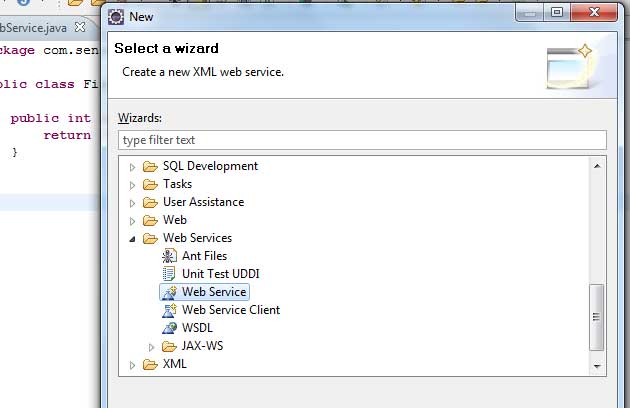
Now you can create a Java class that you would want to expose as a Web Service. I’m going to create new class called FirstWebService and create public method called addTwoNumbers which takes two integers as input and return the addition of them.  
  
3.1 Right Click on MyFirstWebService in Project Explorer and select New –-> Class and give suitable package name and class name. I have given com.sencide as package name and FirstWebService as class name.



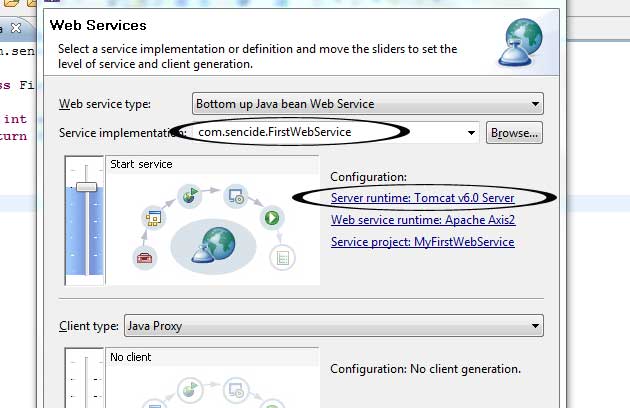
[?](http://blog.sencide.com/2011/06/create-web-service-using-apache-axis2.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | package com.sencide;  public class FirstWebService {   public int addTwoNumbers(int firstNumber, int secondNumber){    return firstNumber + secondNumber;   }  } |

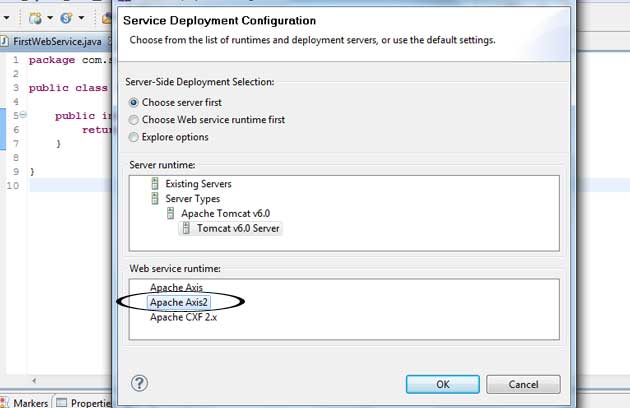
3.2 Then, select File --> New –-> Other and choose Web Service.



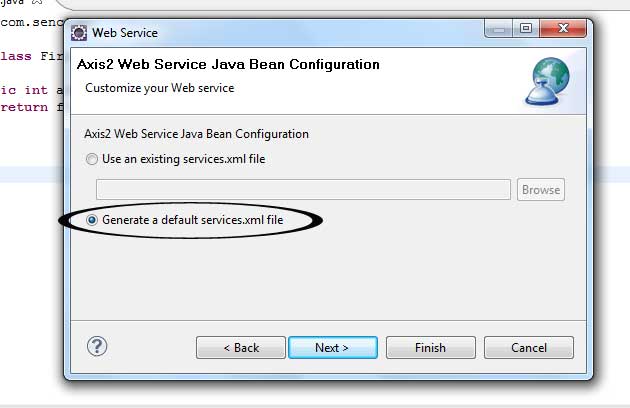
3.3 Select the FirstWebService class as service implementation and to make sure that the Configuration is setup correctly click on Server runtime.



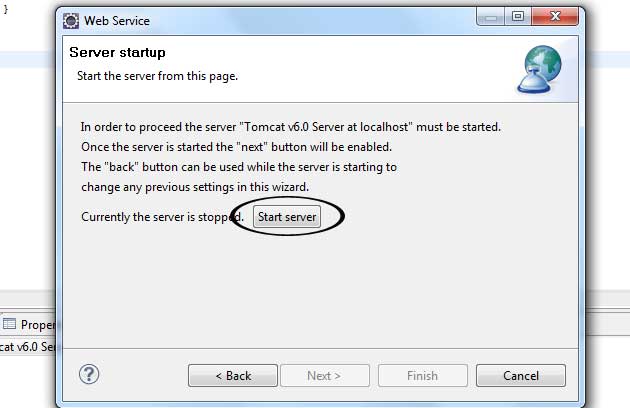
3.4 There set the Web Service runtime as Axis2 (Default one is Axis) and click Ok.



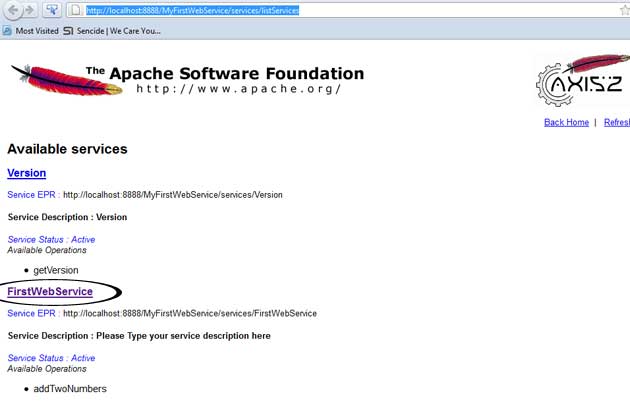
3.5 Click Next and make sure Generate a default service.xml file is selected.



3.6 Click Next and Start the Server and after server is started you can Finish if you do not want to publish the Web service to a test UDDI repository.



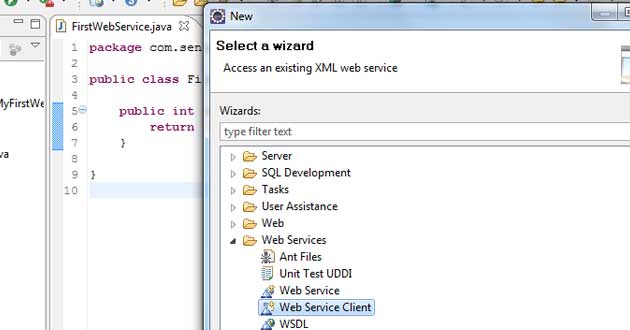
You can go to http://localhost:8888/MyFirstWebService/services/listServices to see your running service which is deployed by Axis2. You can see the WSDL by clicking the link FirstWebService.



We have to use Eclipse every time when we want to run the service if we do not create .aar (Axis Archive) file and deploy it to the server. So let’s see how we can create it.

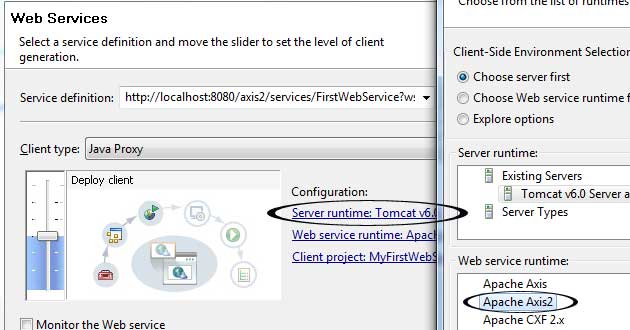
### 4. Creating a Web service client

4.1 Select File --> New --> Other… and choose Web Service Client



4.2 Set he newly created Axis2 Web service (http://localhost:8080/axis2/services/FirstWebService?wsdl) as the Service definition. Then configure the Server runtime as previously and click finish.

1



4.3 This will generate two new classes called FirstWebServiceStub.java and FirstWebServiceCallbackHandler.java. Now we can create test class for client and use our web service. Create new class called TestClient.java and paste following code.

|  |  |
| --- | --- |
|  | package com.sencide;   import java.rmi.RemoteException;  import com.sencide.FirstWebServiceStub.AddTwoNumbers;  import com.sencide.FirstWebServiceStub.AddTwoNumbersResponse;   public class TestClient {    public static void main(String[] args) throws RemoteException {    FirstWebServiceStub stub = new FirstWebServiceStub();    AddTwoNumbers atn = new AddTwoNumbers();    atn.setFirstNumber(5);    atn.setSecondNumber(7);    AddTwoNumbersResponse res = stub.addTwoNumbers(atn);    System.out.println(res.get\_return());     }  } |