

# Developing Client for RESTful Web Service

This tutorial will take you through the steps required in developing, deploying and testing a RESTful Web Service Client in Apache Geronimo for a web services which are already deployed on the server

We will be creating a Web based Client which can access the RESTful web service via GET and POST methods. You can easily create this client without any external tools except that of a server environment.

To run this tutorial, as a minimum you will be required to have installed the following prerequisite software.

- Sun JDK 5.0+ (J2SE 1.5)
- Apache Geronimo 2.x
- Eclipse IDE for Java EE Developers - Europa release
- Geronimo Eclipse Plug-in 2.x

Details on installing Eclipse are provided in the [Development environment](#) section.

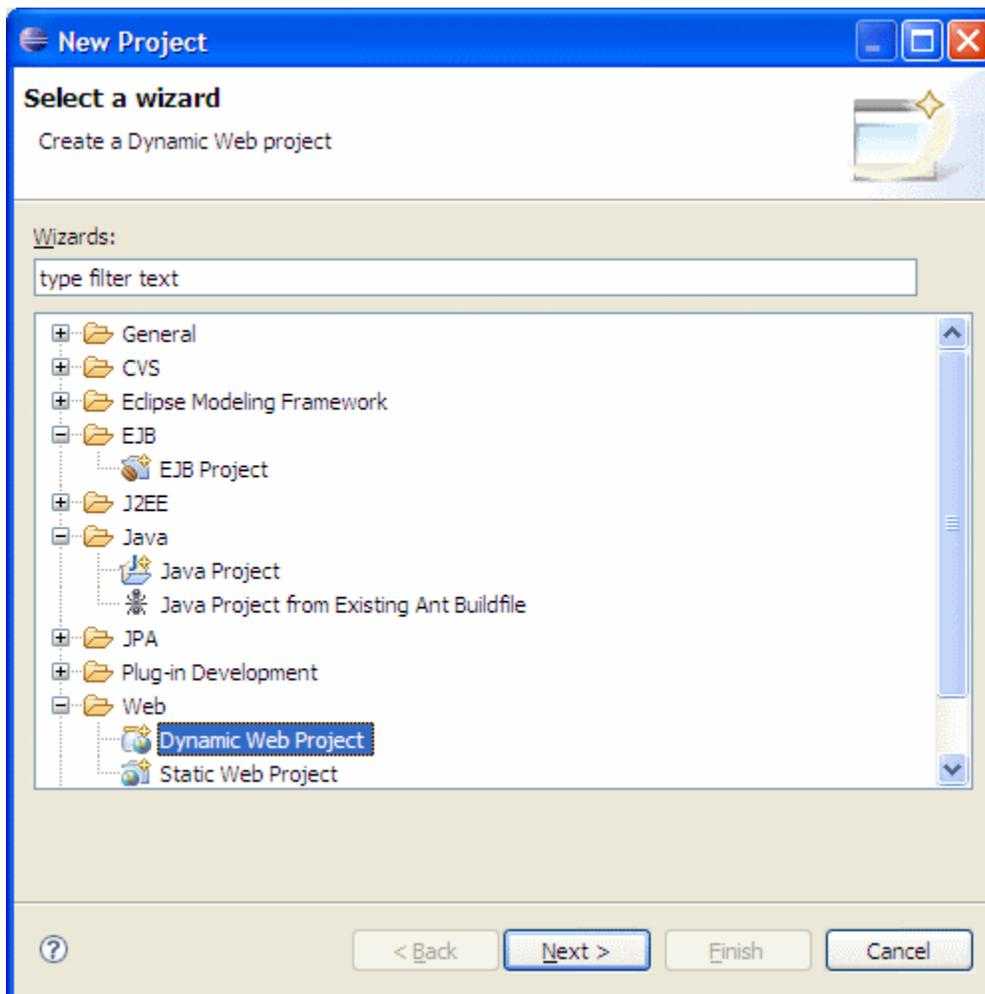
Deployed Web Service

This tutorial assumes that you have completed the [Developing a simple RESTful Service](#) tutorial. Here we will try to develop the client for the web service deployed in the above mentioned tutorial.

This tutorial will take you through the following steps:  
2listpipe

## Create a Dynamic Web Project to consume the Web Service

- From Eclipse main menu, select **File->New->Other**
- In the **New** dialog, select **Web->Dynamic Web Project** and click **Next**



- Type **jaxws-rest-converterclient** as the **Project Name** and click **Next**

**New Dynamic Web Project**

**Dynamic Web Project**  
Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name:

Project contents:

Use default

Directory:

Target Runtime

Configurations

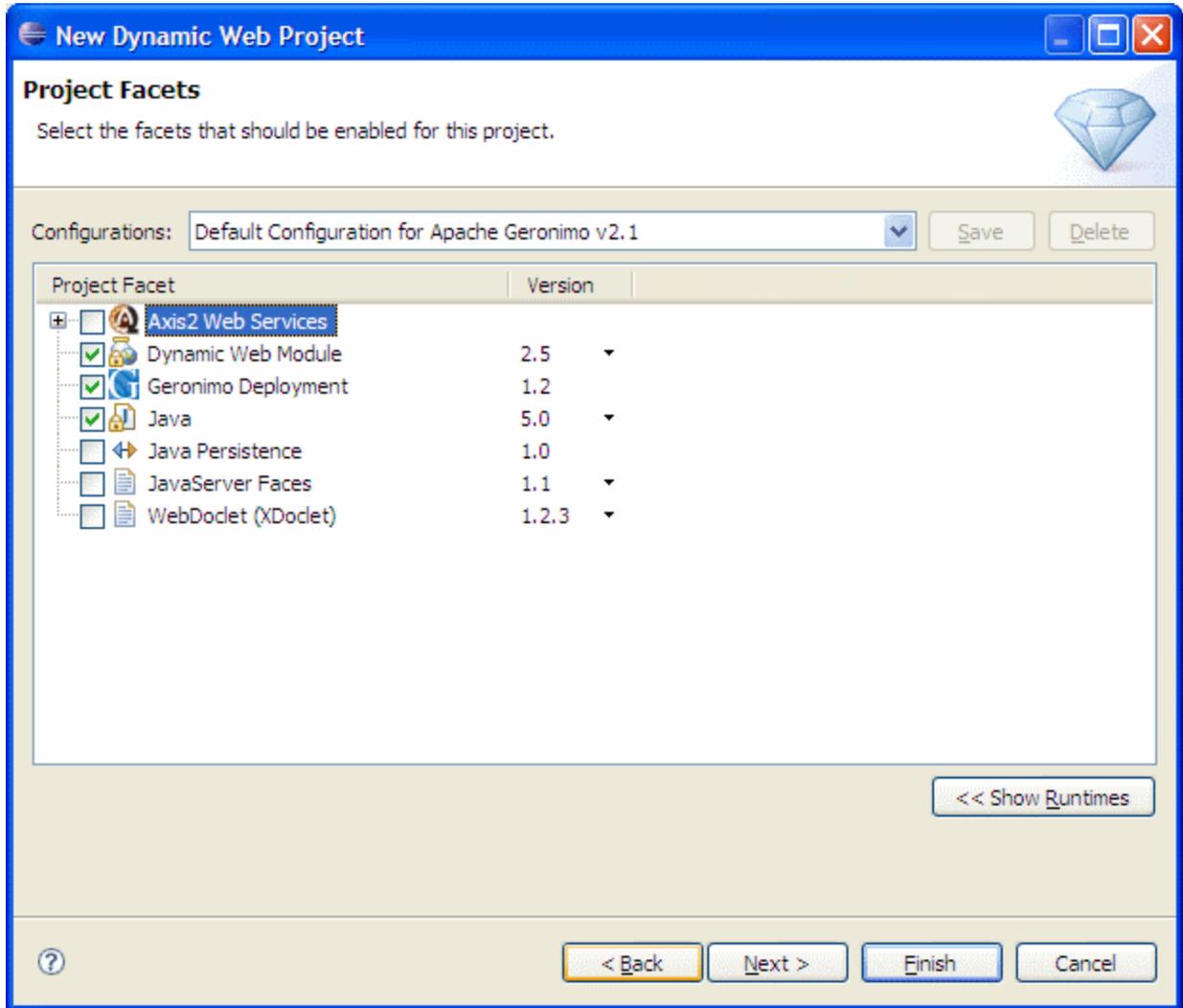
A good starting point for working with Apache Geronimo v2.1 runtime. Additional facets can later be installed to add new functionality to the project.

EAR Membership

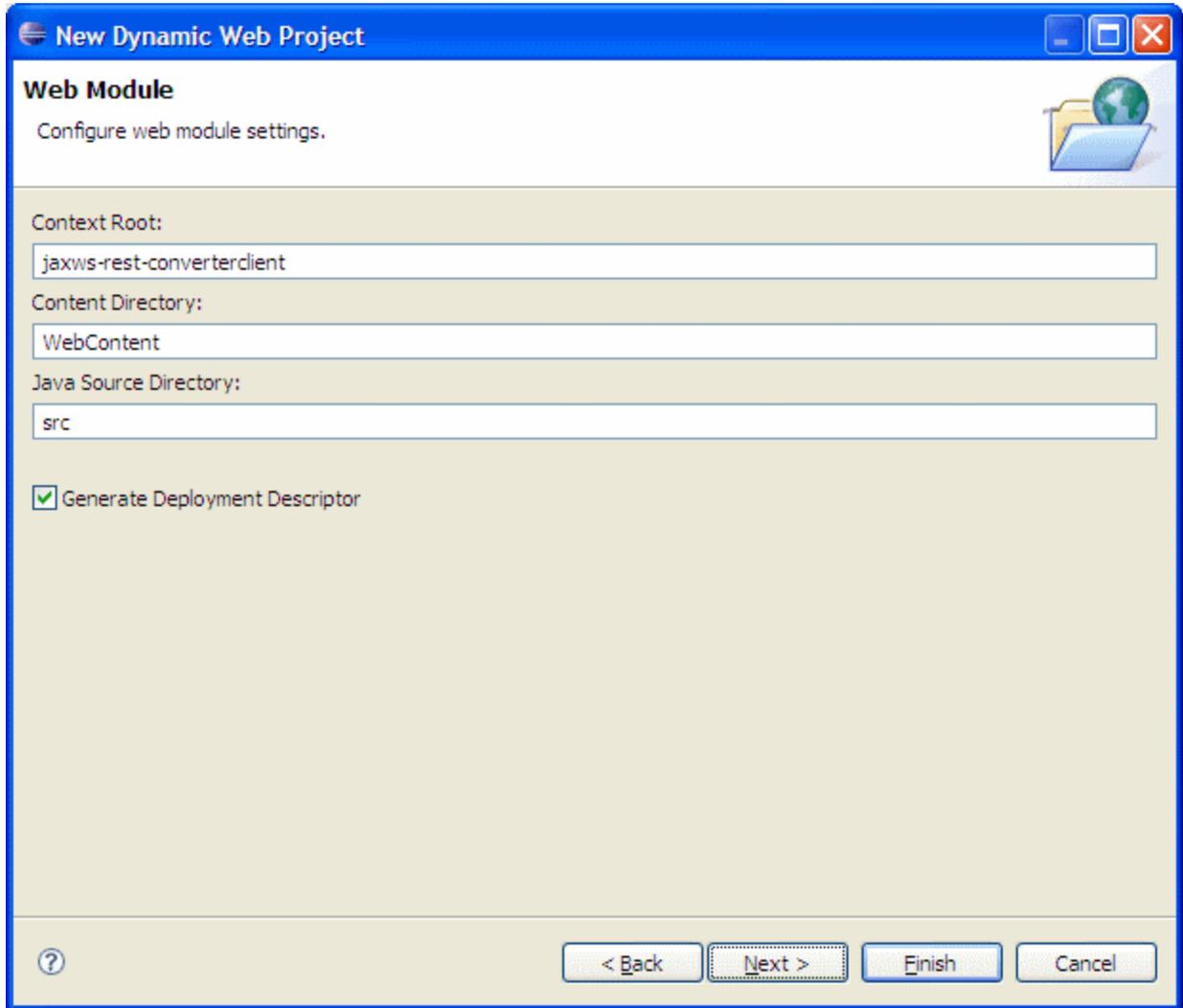
Add project to an EAR

EAR Project Name:

- On the **Project Facets** page, the default selections are enough.



- Make sure that the check box **Generate Deployment Descriptor** is selected and click **Next**



- On the **Geronimo Deployment Page** modify the **Group Id** to **org.apache.geronimo.samples.jaxws.rest** and the **Artifact Id** to **jaxws-rest-converterclient**.

**New Dynamic Web Project**

**Geronimo Deployment Plan**  
Configure the geronimo deployment plan.

Group Id:

Artifact Id:

Version:

Artifact Type:

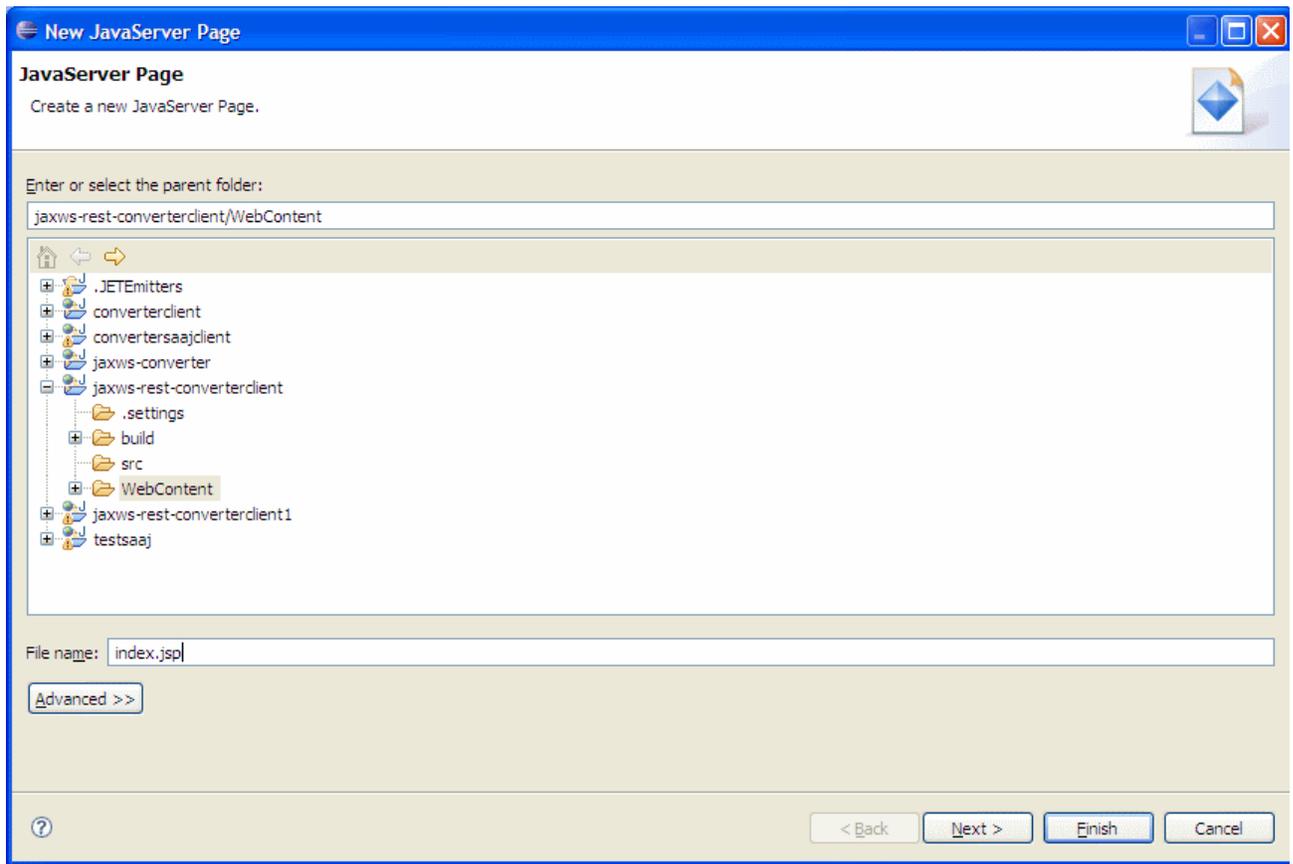
Add a runtime dependency to Geronimo's shared library

? < Back Next > Finish Cancel

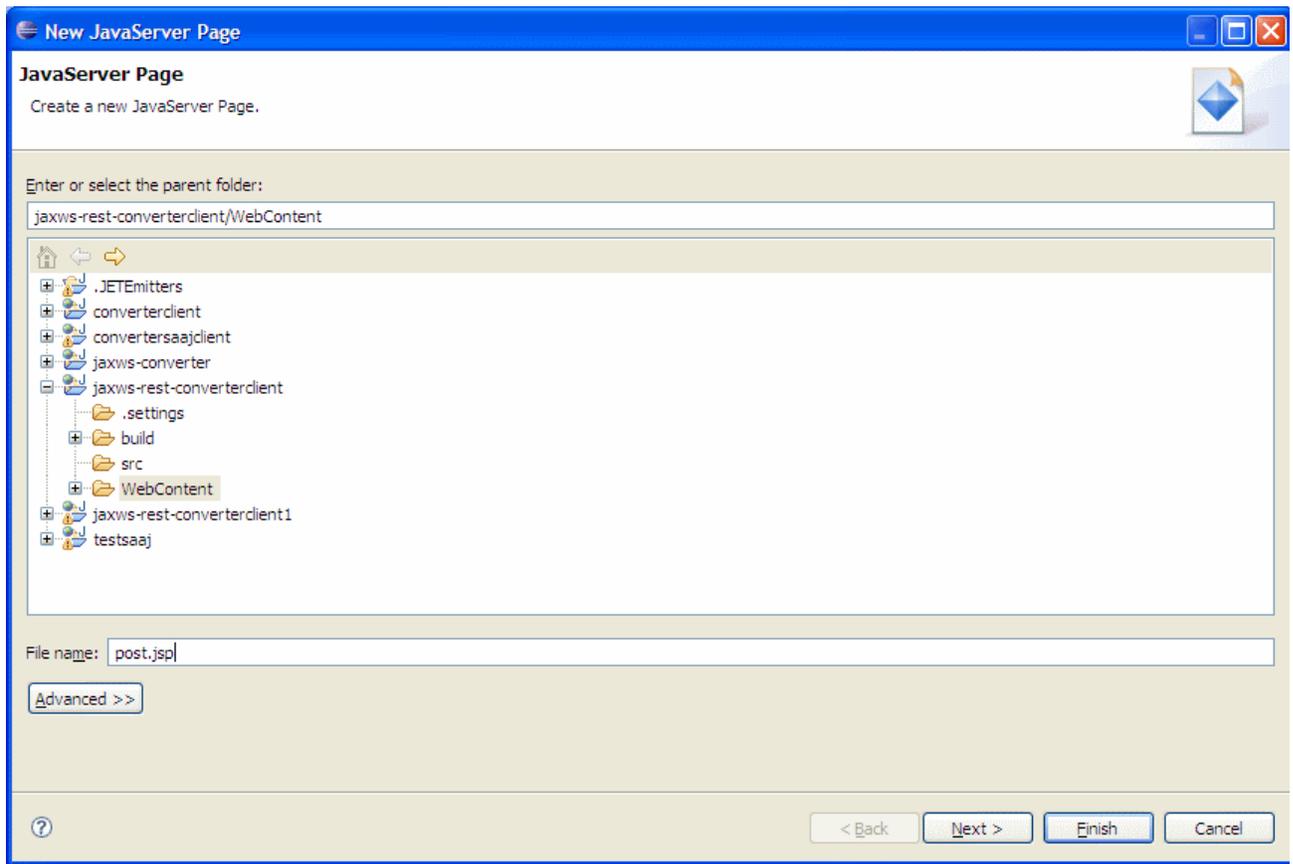
- Click **Finish**

## Developing the Web based Client

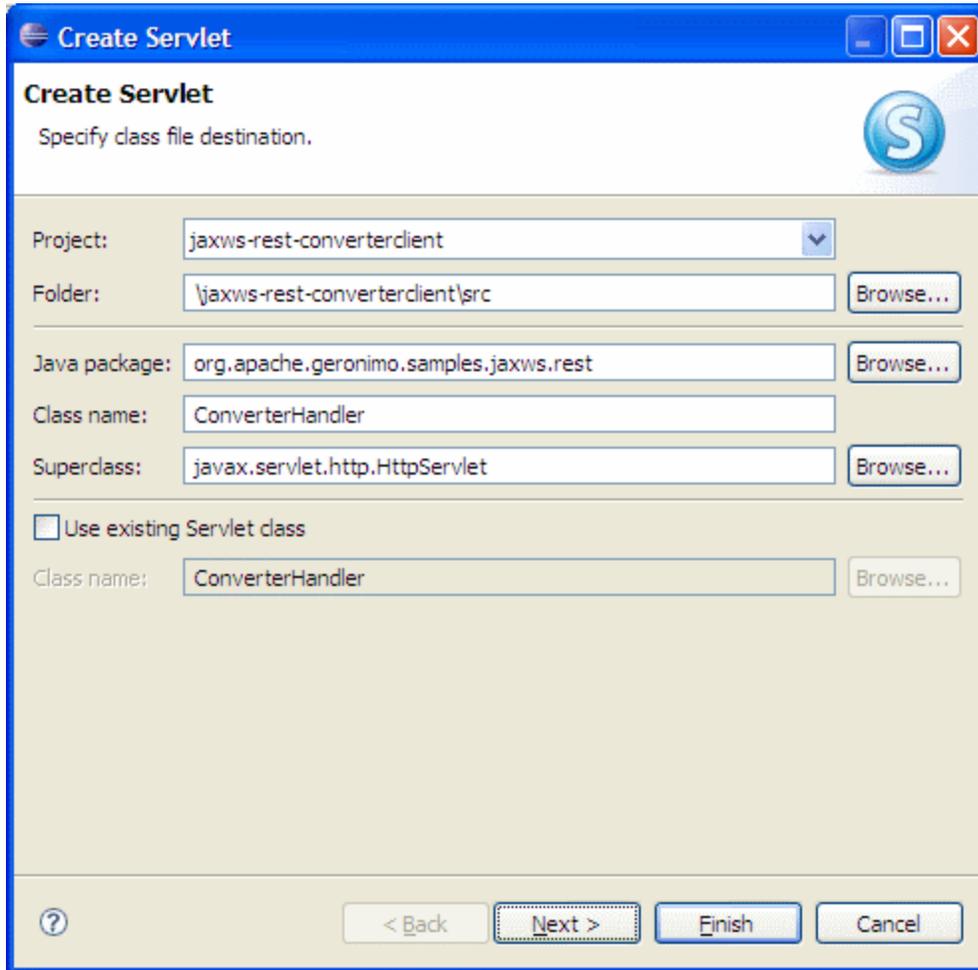
- Right Click the **jaxws-converterclient**, and Select **New->JSP**
- Name the jsp as **index.jsp** and click **Finish**



- Add the following code to the **index.jsp**  
index.jspsolid
- Right click again and add another jsp named **post.jsp**



- Add the following code to **post.jsp**  
post.jspsolid
- Right click again and add a **Servlet** named **ConverterHandler**



- Add the following code to **ConverterHandler.java**

ConverterHandler.java  
 Errors in Servlet

Here the servlet might notify that some imports are not resolved. We need to add two external jar files to resolve these errors.

The jar files that needed to be added in the build path are:

**commons-codec-1.3.jar** - <GERONIMO\_INSTALL\_DIR>\repository\commons-codec\commons-codec\1.3\commons-codec-1.3.jar

**commons-httpclient-3.0.1.jar** -

<GERONIMO\_INSTALL\_DIR>\repository\commons-httpclient\commons-httpclient\3.0.1\commons-httpclient-3.0.1.jar

- Let us walkthrough the code of **ConverterHandler** servlet.
  - We will create a **HttpClient** object and **GetMethod** object for our **requestquery**. Then we will execute the method on our client object.
  - The response after executing the method will be taken as a stream so as to parse the XML elements in response message sent by the Web Service.
  - Here we used XPath to process XML response messages sent by web service. XPath Processing XPath (XML Path Language) is a language for selecting nodes from an XML document with very advanced features.

Our response message sent by web service may look like this:  
 response.xml

We will first try to get the NodeList present between <return> elements, Then we evaluate each child node to get the results.

This concludes the development section of our web based client.

## Setting Up the Deployment Plan

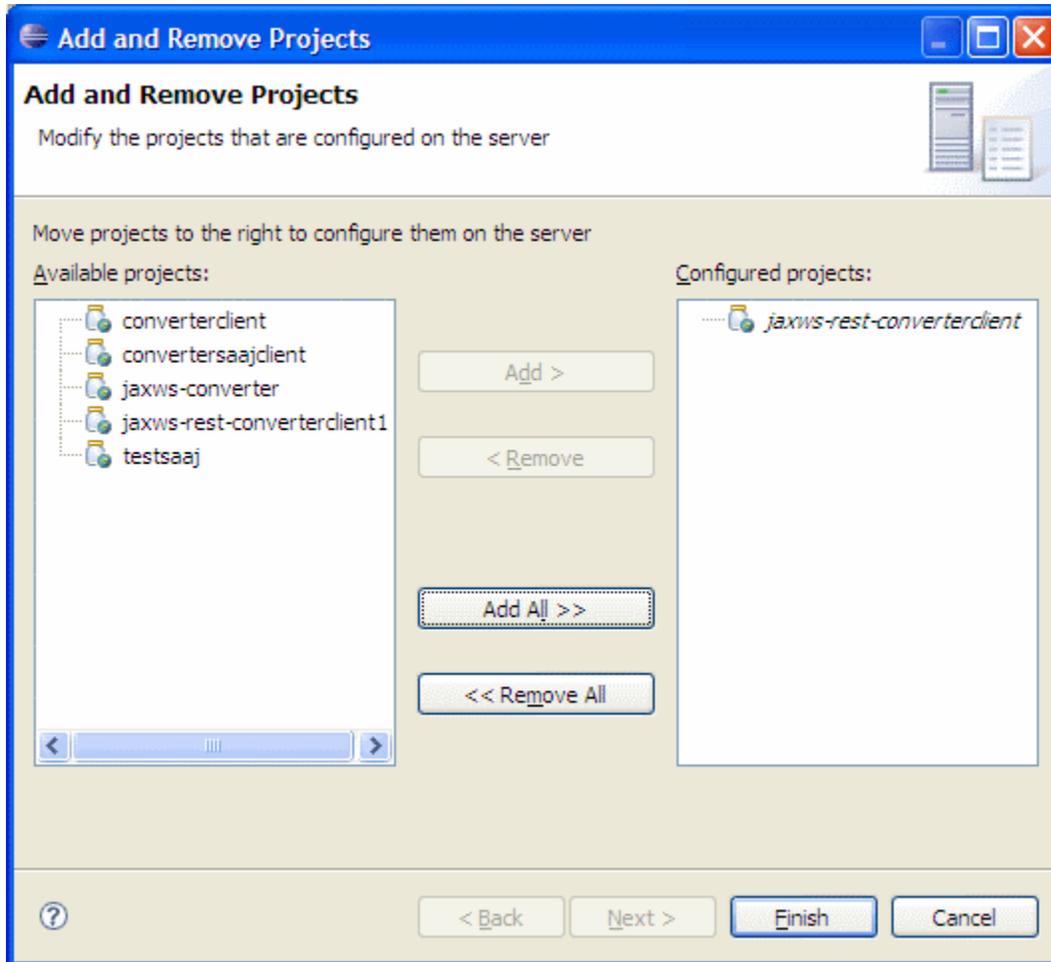
- As two external jars have been added to the project's build path, One needs to specify them as dependencies so that at runtime application can resolve the classes.
- Expand **WebContent/WEB-INF** directory and open **geronimo-web.xml**

- Add the following code to **geronimo-web.xml**  
geronimo-web.xmlsolid

## Deploying and Testing the Web Client

### Deploy

- Right click on the **Apache Geronimo** Server Runtime present in the servers view and select **Add and Remove Projects**
- Add **jaxws-rest-converterclient** to configured projects list and then click **Finish**



- Wait for some time till the server status changes to **Synchronized**

### Testing

- Right click the **index.jsp** present under WebContent directory of our project and select **Run As->Run On Server**
- In the popup, check the check box **Always use this server when running the project** and then click **Finish**
- Now Eclipse will try to open the jsp in a web browser which shows you a form to enter amount in Dollars.
- Enter any amount and press **submit**, the jsp should display the result that is returned by the web service.

## This form invokes a Web Service.

Please type an amount and click submit to see the result.

Amount:

23 Dollars equals to 933.34 Rupees  
933.34 Rupees equals to 17.15 Euros

### Test via Post

The code for testing the RESTful service via POST method hasn't been added to the ConverterHandler.

A sample post file might look like this

SampleRequest.xmlsolid

This completes the development of a simple RESTful client that consumes a RESTful Web Service that is already deployed on the server.