Exercise 3

Creating a JAX-WS Service

Prior Knowledge
Basic understanding of SOAP and WSDL

Objectives
Understand how to create Web Services in Java

Software Requirements

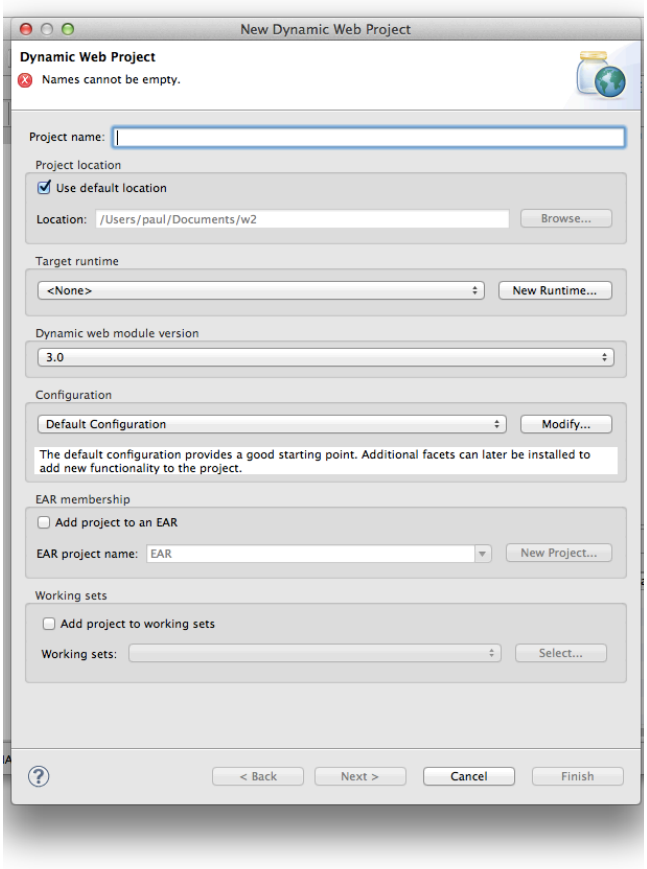
- Java Development Kit 7
- Tomcat 7.0.57 or later
- Eclipse JEE workbench Kepler
- Apache CXF 2.7.x (not 3.0.x)
- SOAPUI

1. Make sure Tomcat is installed (e.g. in ~/servers/tomcat)

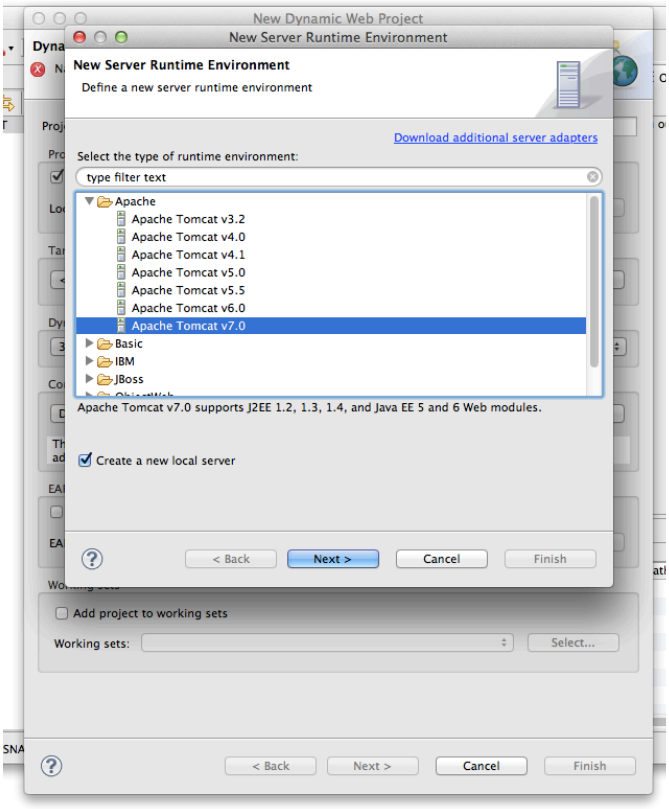
2. Check Apache CXF is installed (in ~/servers/apache-cxf-2.7.13/)

3. Open up Eclipse

4. Create a new Dynamic Web Project (**File -> New -> Dynamic Web Project**) (if this isn't visible, it may be you are on a different perspective – then choose Other and search for Dynamic).

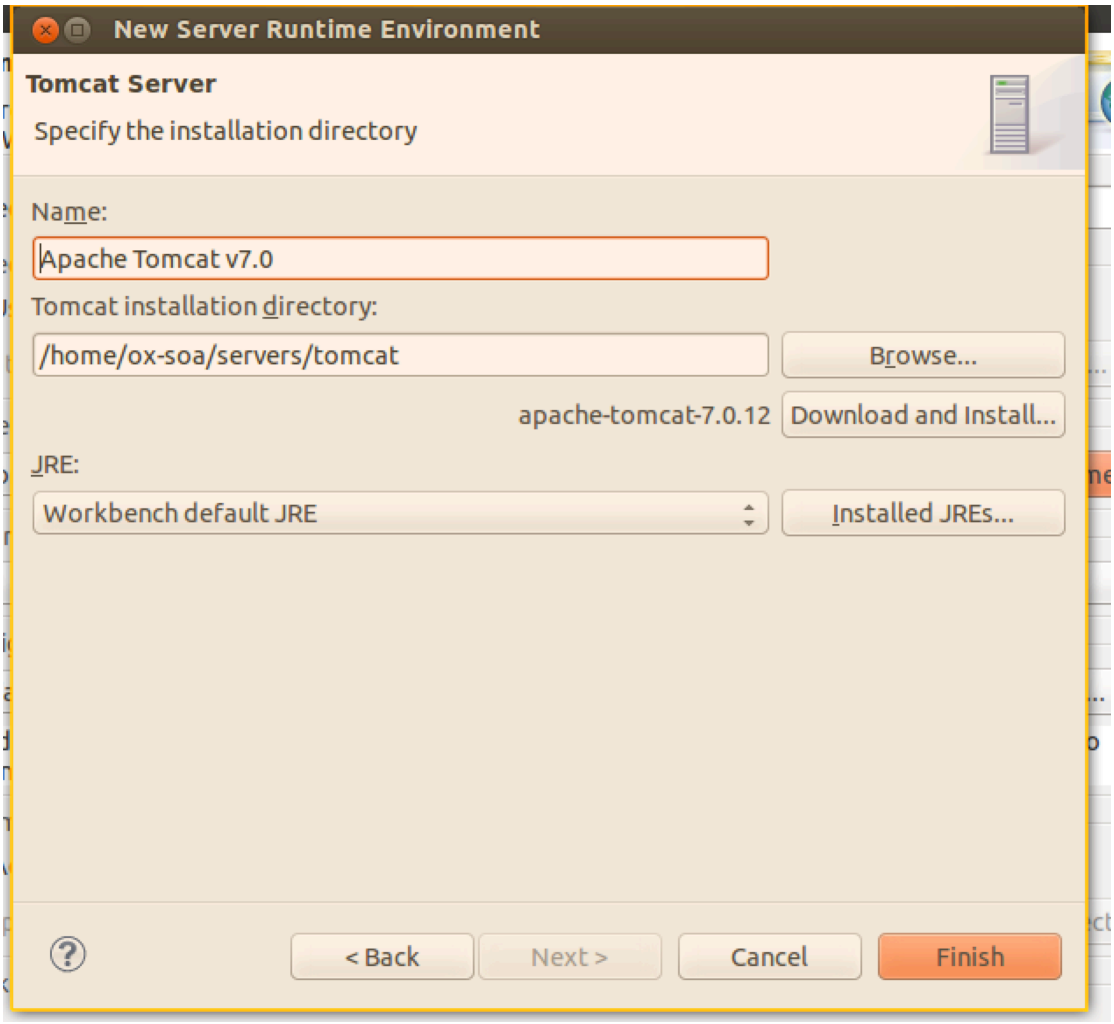


5. Now click **New Runtime** and choose Tomcat 7.0

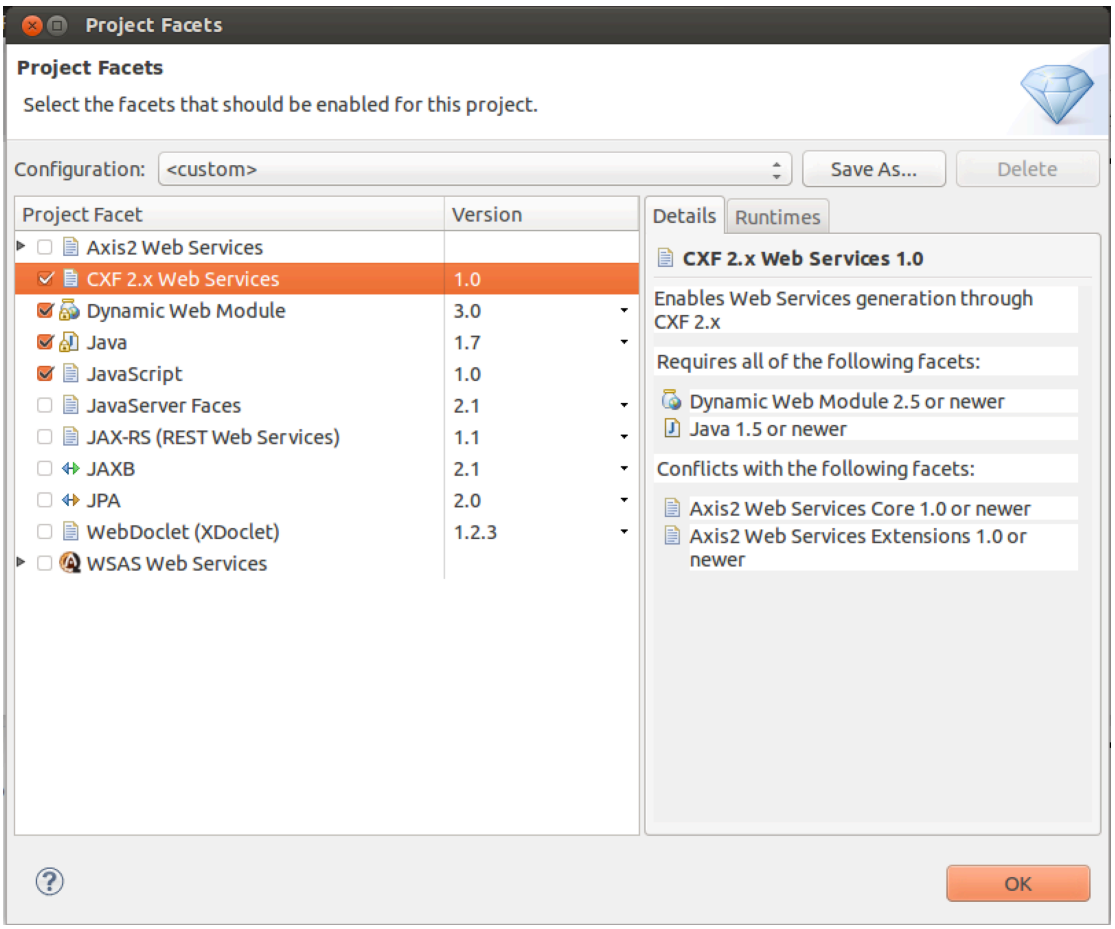


6. Click **Next >**

7. Browse to the directory where you have Apache Tomcat installed and select that:



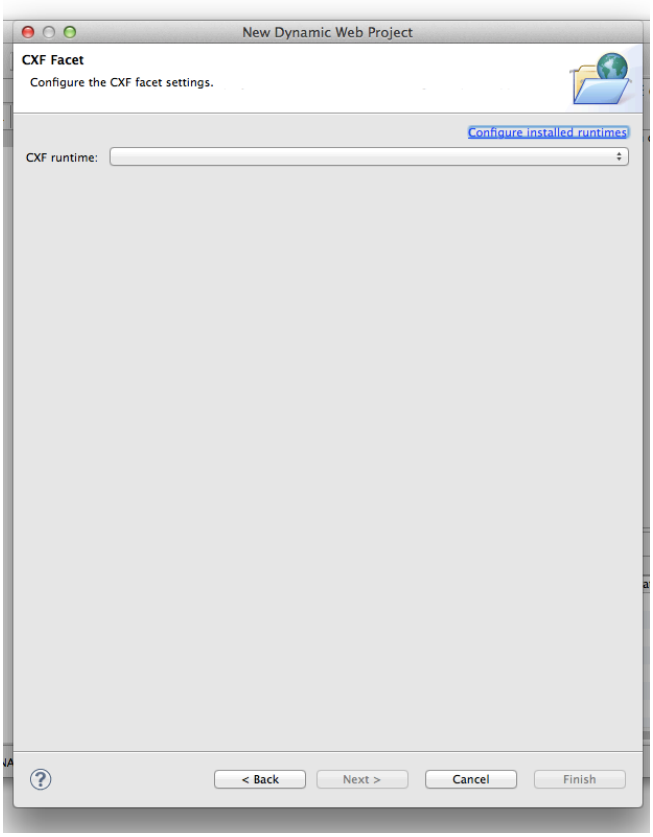
8. Now we need to modify the Tomcat configuration. Click on **Modify** next to Default Tomcat Configuration:



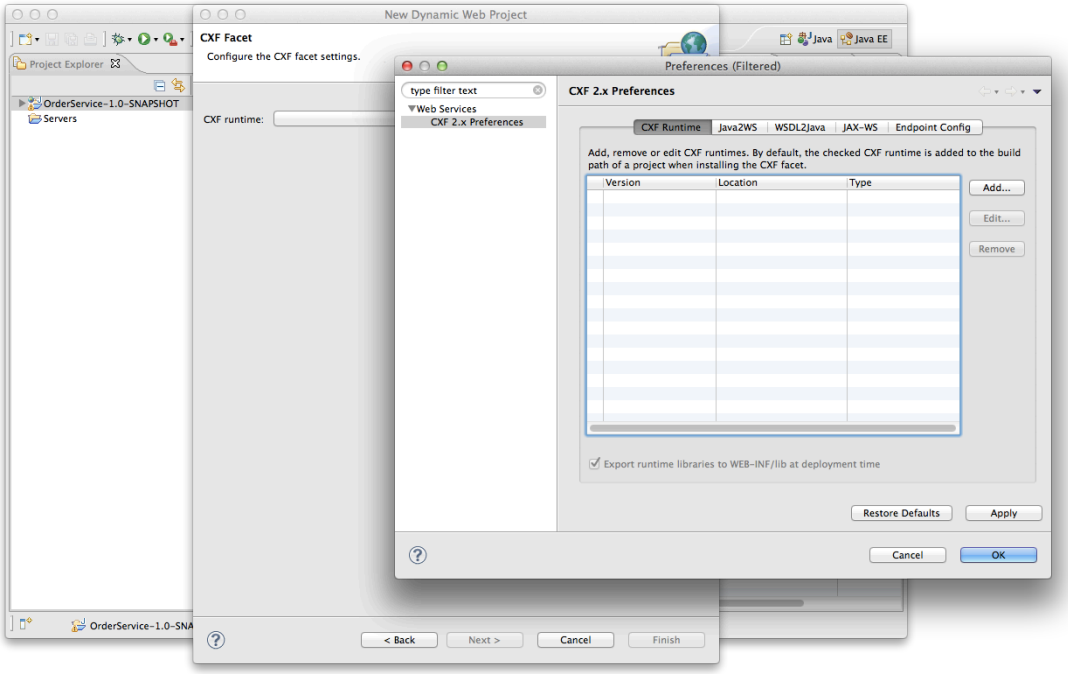
   Enable the **CXF 2.x Web Services** section. Click **OK**

9. Now you should be back in the initial Dialog, so give the project a name, e.g. **JAXWSSample**

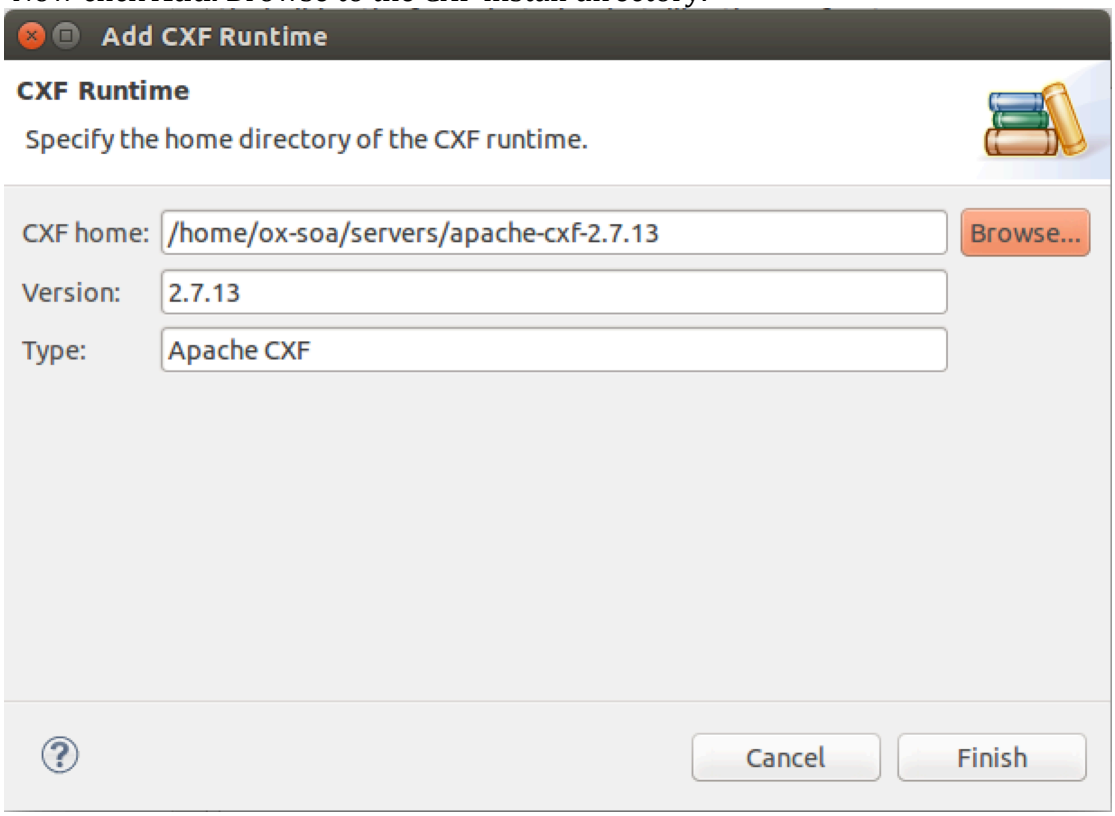
10. Click **Next, Next, Next** until you are at the CXF Facet configuration dialog



11. Click **Configure Installed Runtimes**



12. Now click **Add.** Browse to the CXF install directory.



Now click **Finish**

13. Now select the tick box next to the CXF 2.7.13 version. Click **OK** and **Finish**. You should now have a project.

14. Import the following four files into the project's src folder

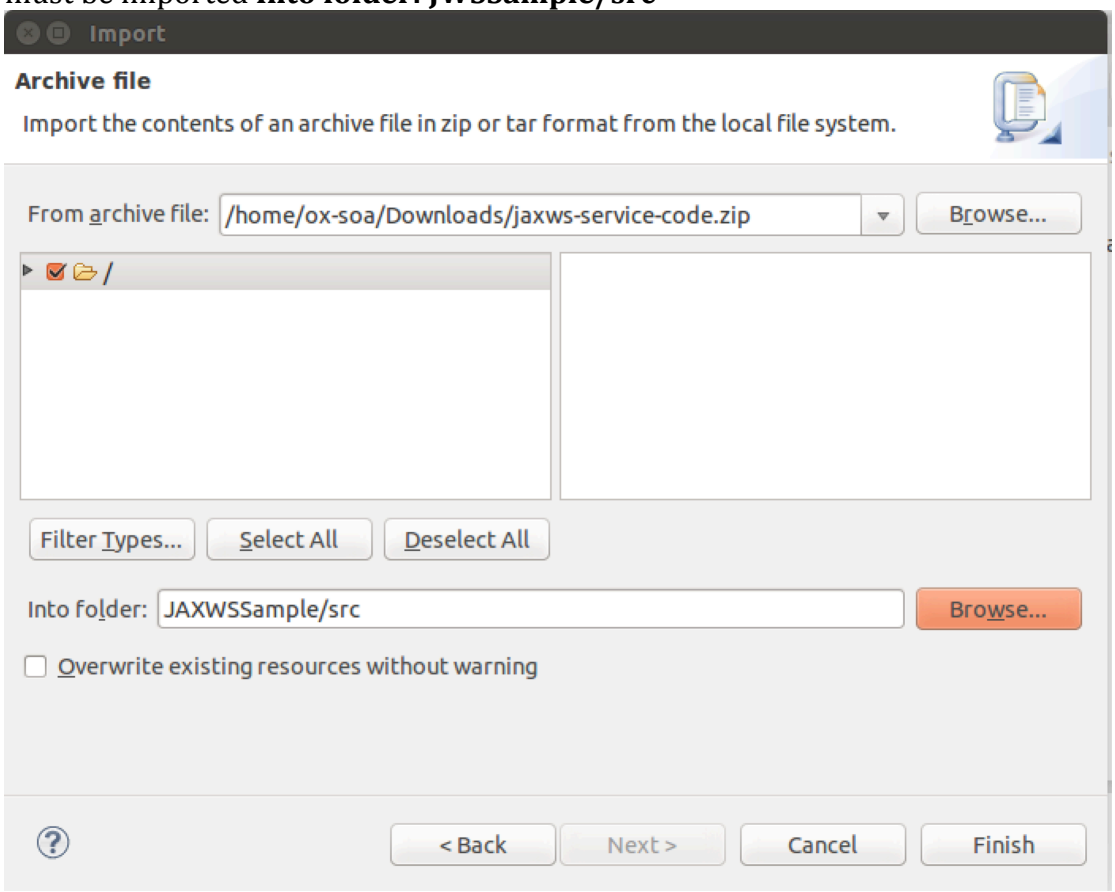
    freo.me.jaxws.Order
    freo.me.jaxws.OrderService
    freo.me.jaxws.OrderServiceImpl
    freo.me.jaxws.NotFoundException

    From the following file:
    ~/Downloads/jaxws-service-code.zip

    [The files are also in:
    https://github.com/pzfreo/ox-soa/blob/master/lab-exercises/code/jaxws-service-code.zip?raw=true

    You can get the file using curl:

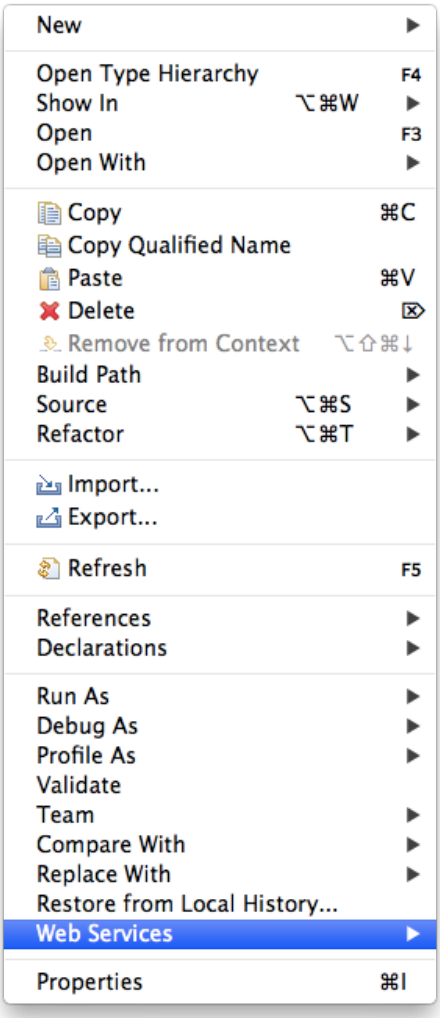
    ```
    curl https://raw.github.com/pzfreo/ox-soa/master/lab-exercises/source/jaxws-service-code.zip -o jaxws-service-code.zip
    ```
    ]
15. You can import using the Eclipse Import Archive File. Note the imported files must be imported **Into folder: JWSSample/src**



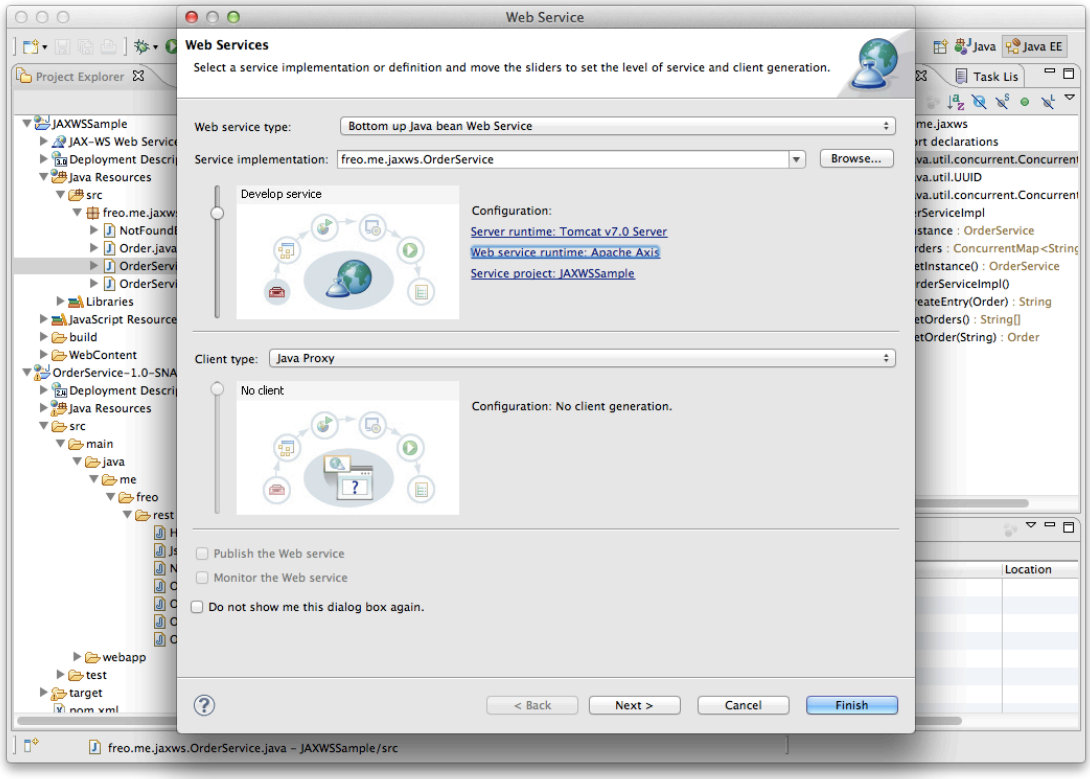
16. Take a good look at the code and understand it.

    Now we can expose this code as a Web Service using the JAXWS wizard built into Eclipse.

17. Select the **OrderService** class and Right-Click on it. Now Select **Web Services**. Then **Create Web Service**.



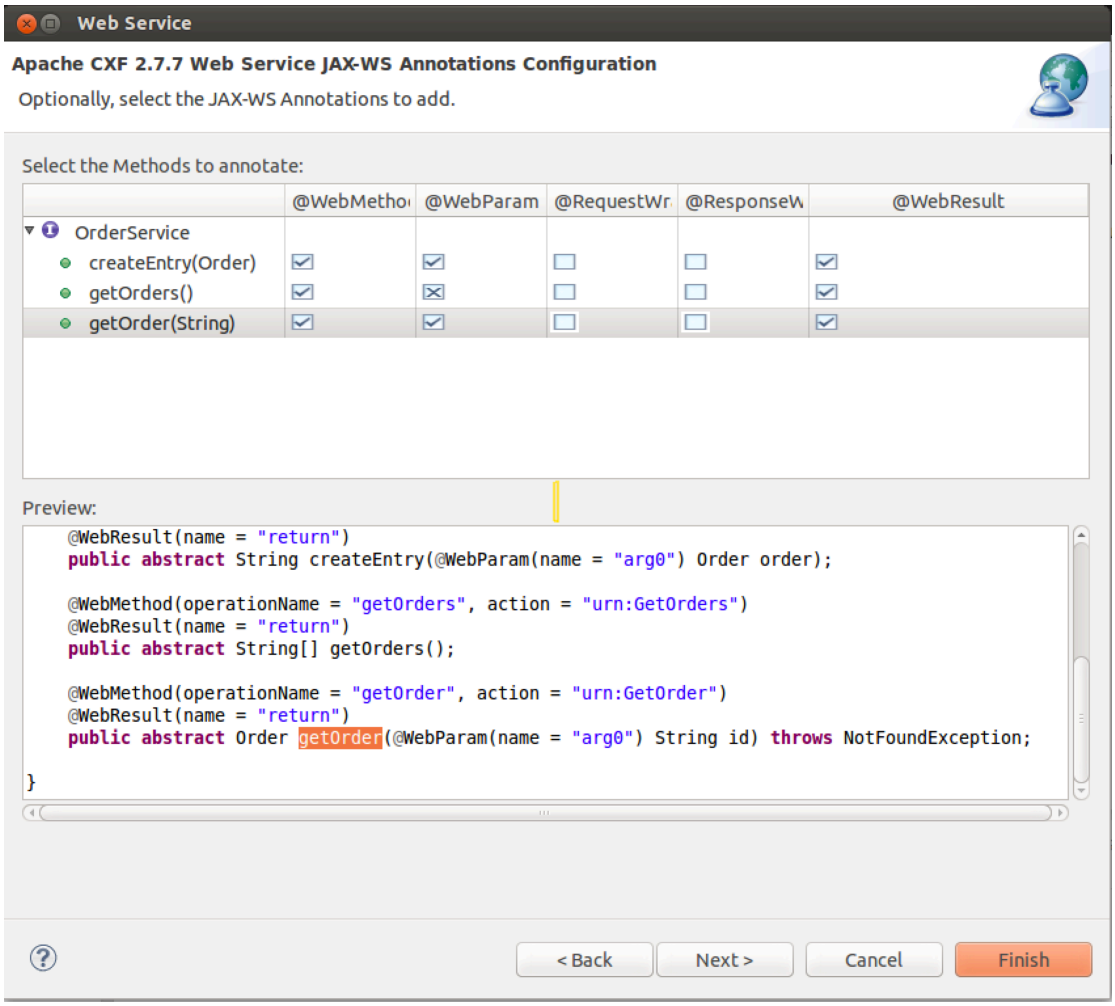
18. You will see the following dialog



19. Now Click on the **Web Service Runtime** and change it to **Apache CXF v2.x.** Click **Next>**

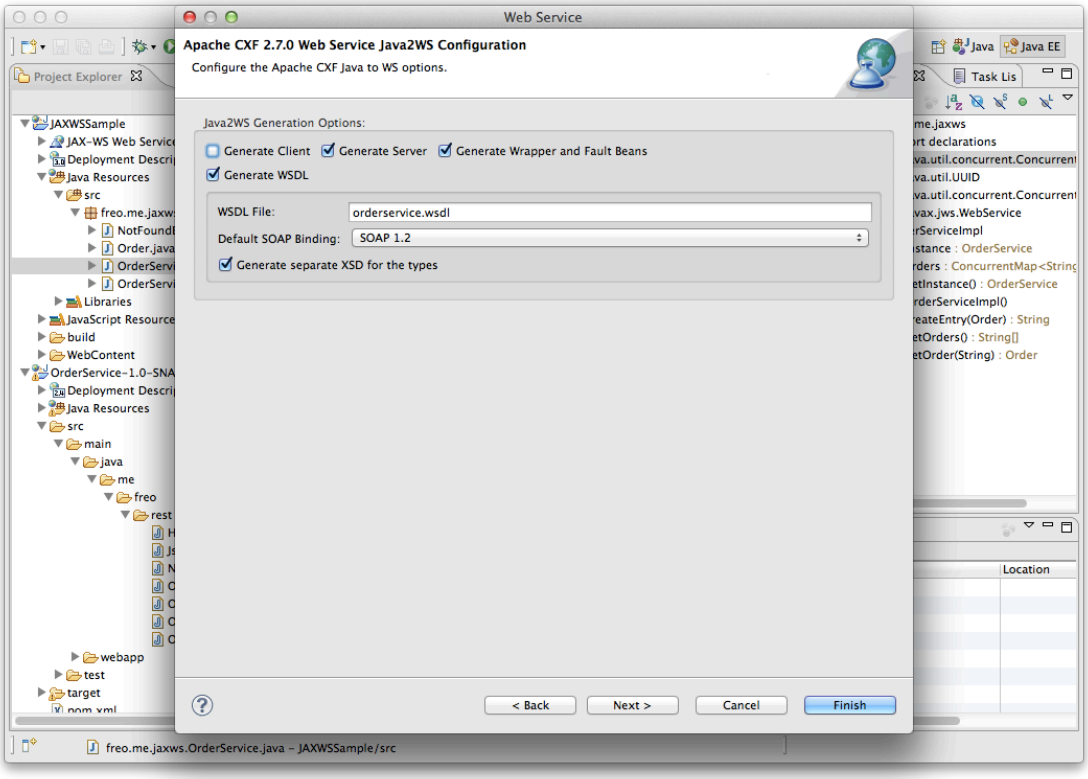
20. Choose **freo.me.jaxws.OrderServiceImpl** as the Service Implementation. Click **Next>**

Click **Next>**
You will see a screen like this:

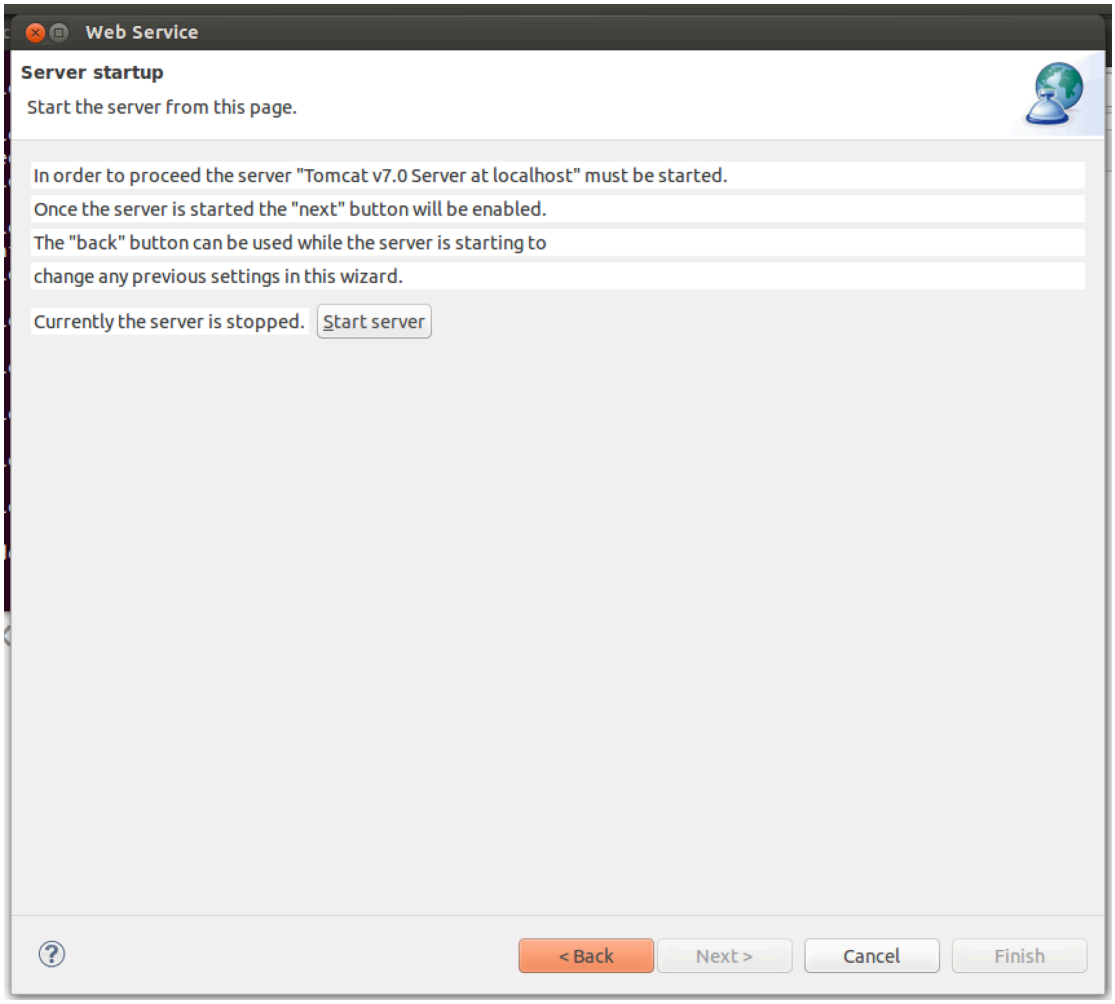


For each method, select the **WebMethod**, **WebParam** and **WebResult** boxes and remove the **Request Wrapper** and **Response Wrapper** checkmarks as shown.
Then click **Next>**.

21. You should see:

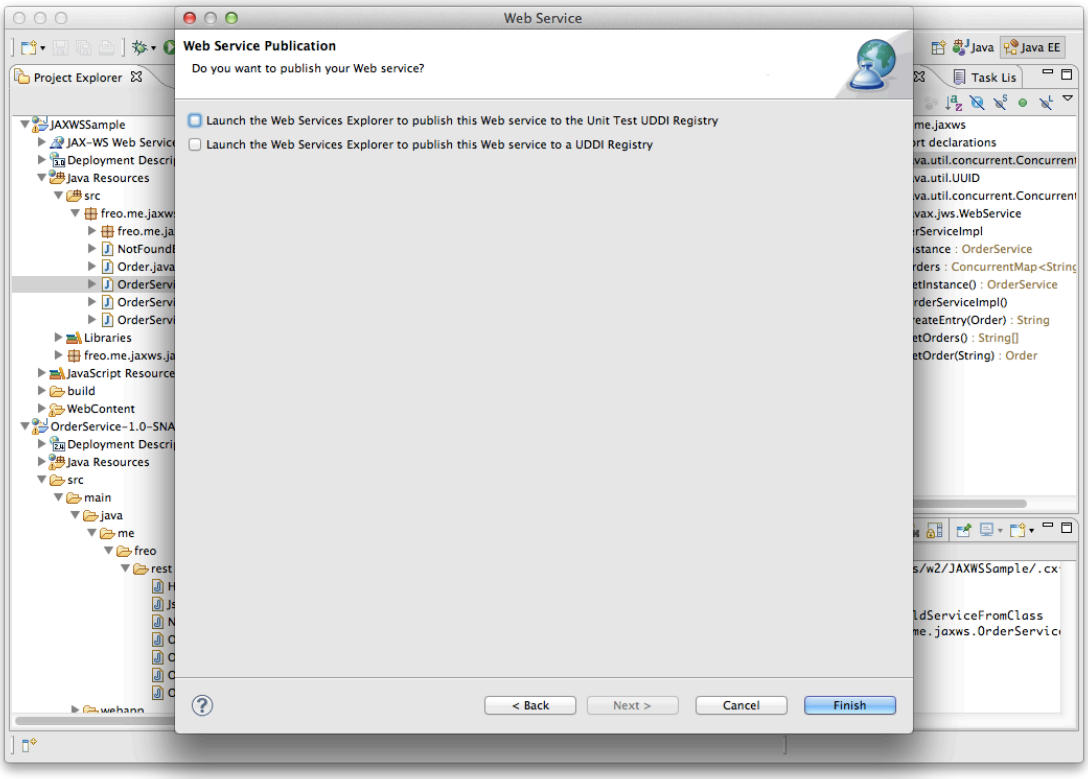


Now make sure Generate Server is clicked. Click **Next>**



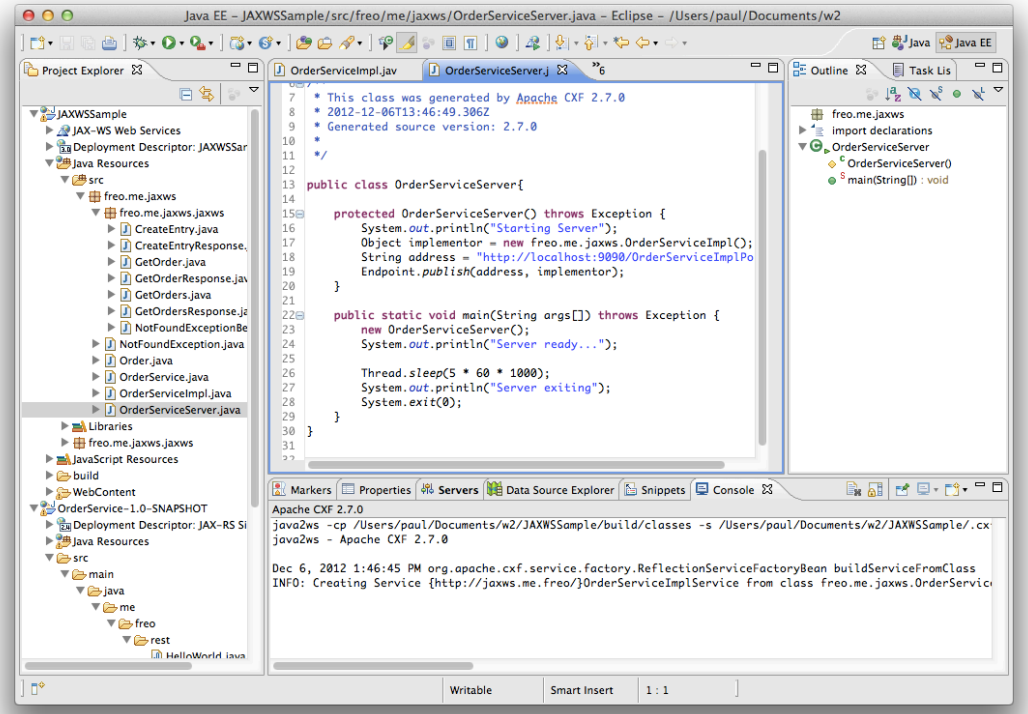
Click **Start Server**
Then **Next**

22. Do **not** select either UDDI option:



23. Click **Finish**

24. This will generate a number of additional classes.

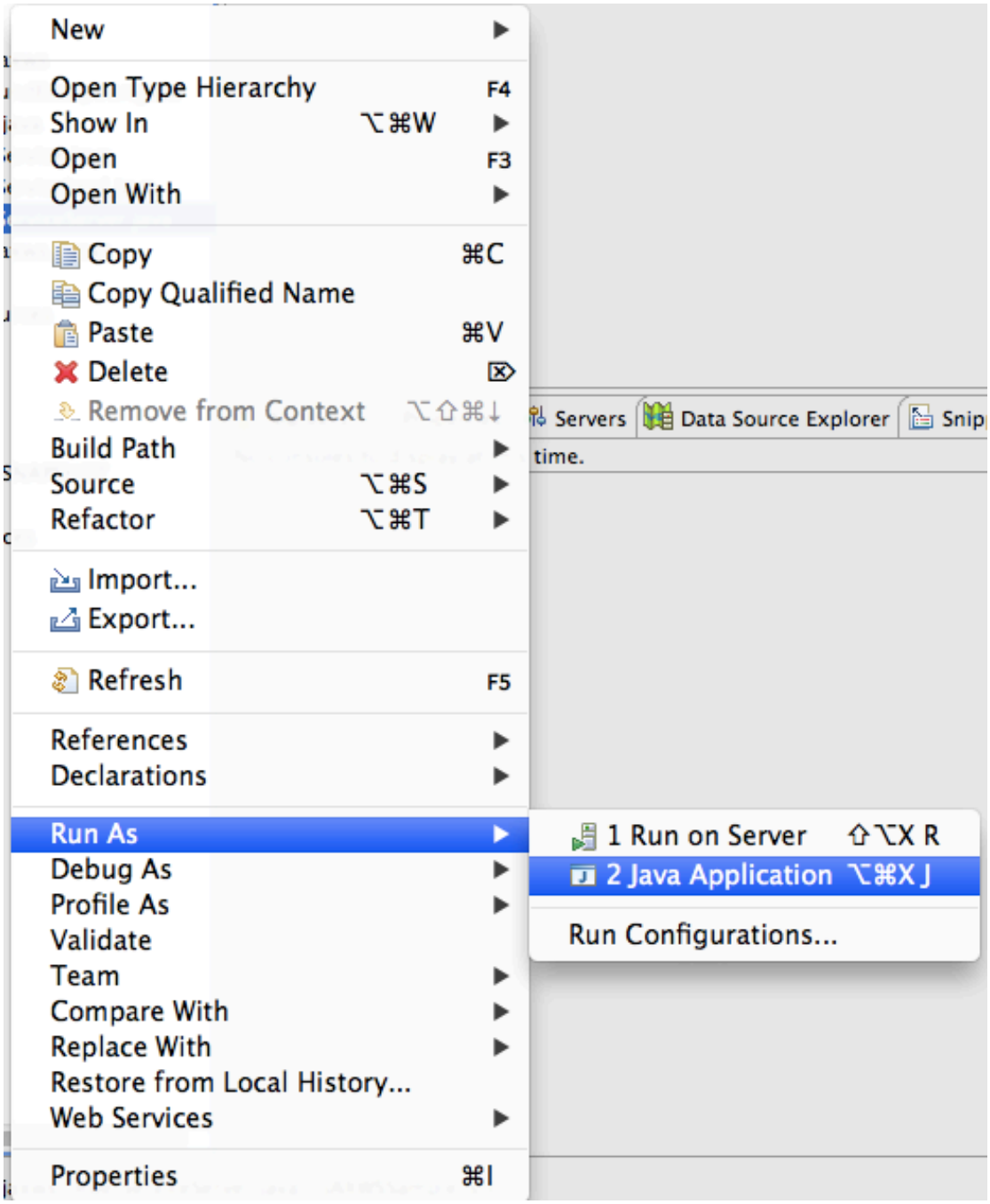


Firstly, it will create a set of classes to map XML types. Secondly it will create a set of annotations in your existing code. Finally it will create a Server class that can be used to run the service standalone.

25. Look through the code and identify all three parts. Take a look at the annotations.

26. You can now run this in one of three ways:
    * Using the generated server class
    * Deploying in Tomcat using Eclipse [This may already be running!]
    * Deploying in Tomcat standalone

27. To run the Generated Server class, find the class called OrderServiceServer, and right-click on it. Now choose **Run As -> Java Application**



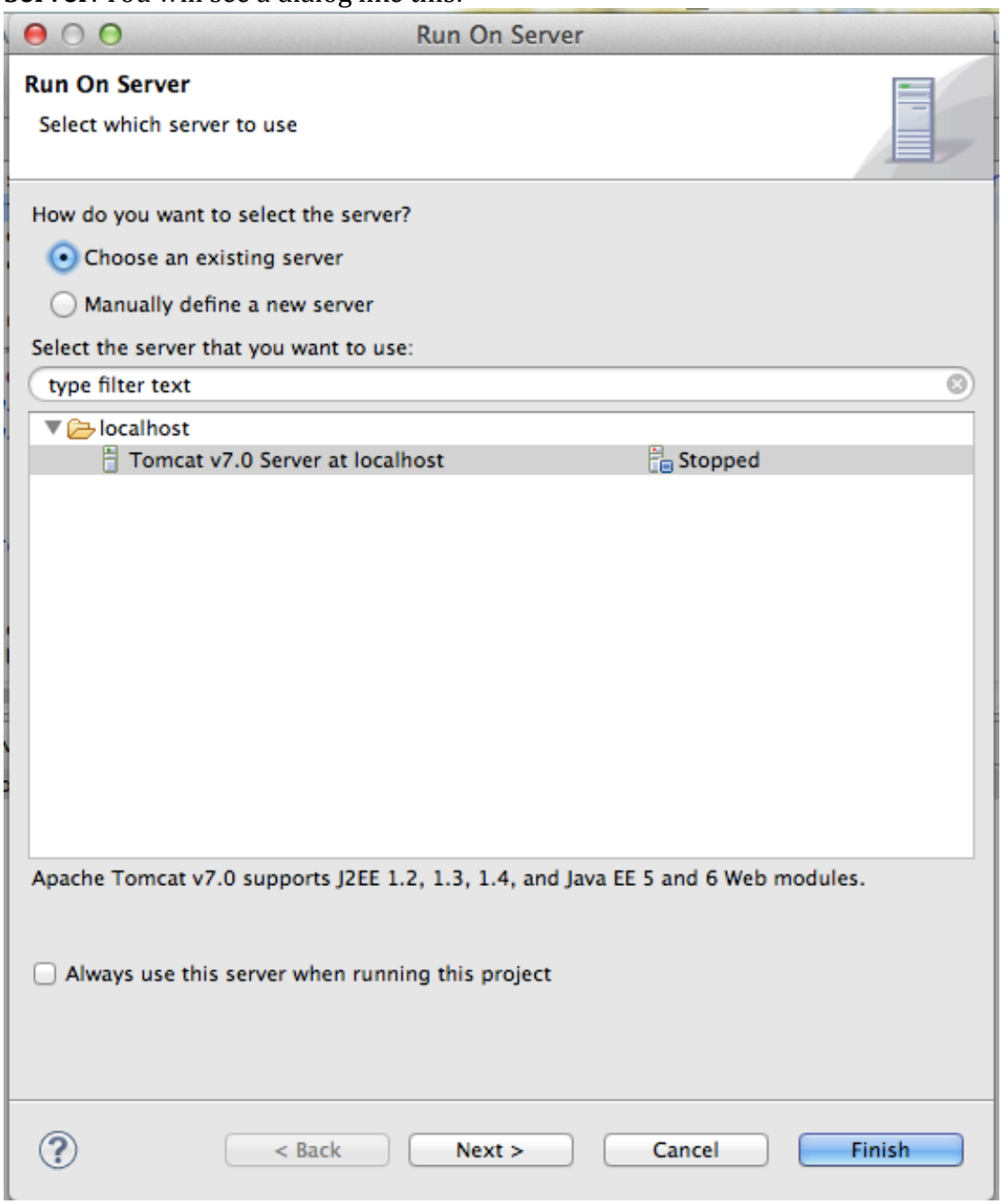
In the console window you should see output similar to:

```
Starting Server
Dec 8, 2012 6:46:04 AM
org.apache.cxf.service.factory.ReflectionServiceFactoryBean
buildServiceFromClass
INFO: Creating Service {http://jaxws.me.freo/}OrderServiceImplService from
class freo.me.jaxws.OrderService
Dec 8, 2012 6:46:06 AM org.apache.cxf.endpoint.ServerImpl initDestination
INFO: Setting the server's publish address to be
http://localhost:9090/OrderServiceImplPort
Dec 8, 2012 6:46:06 AM org.eclipse.jetty.server.Server doStart
INFO: jetty-8.1.7.v20120910
Dec 8, 2012 6:46:06 AM org.eclipse.jetty.server.AbstractConnector doStart
INFO: Started SelectChannelConnector@localhost:9090
Dec 8, 2012 6:46:06 AM
org.apache.cxf.service.factory.ReflectionServiceFactoryBean
buildServiceFromWSDL
INFO: Creating Service {http://docs.oasis-open.org/ws-
dd/ns/discovery/2009/01}Discovery from WSDL:
classpath:/org/apache/cxf/ws/discovery/wsdl/wsdd-discovery-1.1-wsdl-os.wsdl
Dec 8, 2012 6:46:06 AM org.apache.cxf.endpoint.ServerImpl initDestination
INFO: Setting the server's publish address to be
soap.udp://239.255.255.250:3702
Dec 8, 2012 6:46:06 AM
org.apache.cxf.service.factory.ReflectionServiceFactoryBean
buildServiceFromClass
INFO: Creating Service {http://docs.oasis-open.org/ws-
dd/ns/discovery/2009/01}DiscoveryProxy from class
org.apache.cxf.jaxws.support.DummyImpl
Server ready...
```

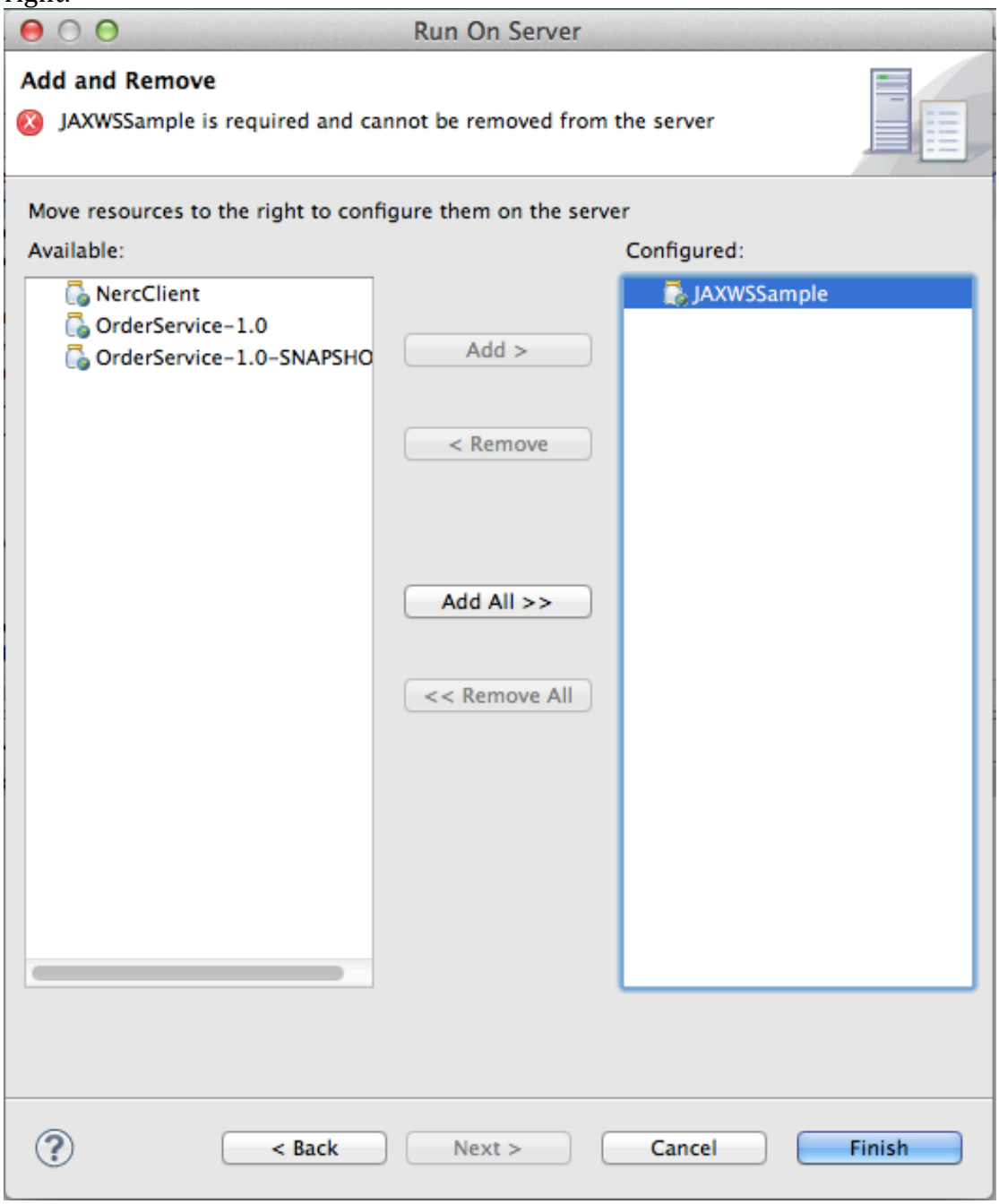
28. Notice the line I have bold/italicized. Find the same line in your console and cut/paste it into your browser address. Don't hit enter (well you can – it won't explode). Instead add a **?wsdl** to the end. Now hit enter. You should see a WSDL.

29. Try your service out with SOAPUI as before.

30. To run Tomcat locally "within" Eclipse, choose the overall JAXWSSample app in the left hand Project Explorer, right-click and select **Run As->Run on Server**. You will see a dialog like this:

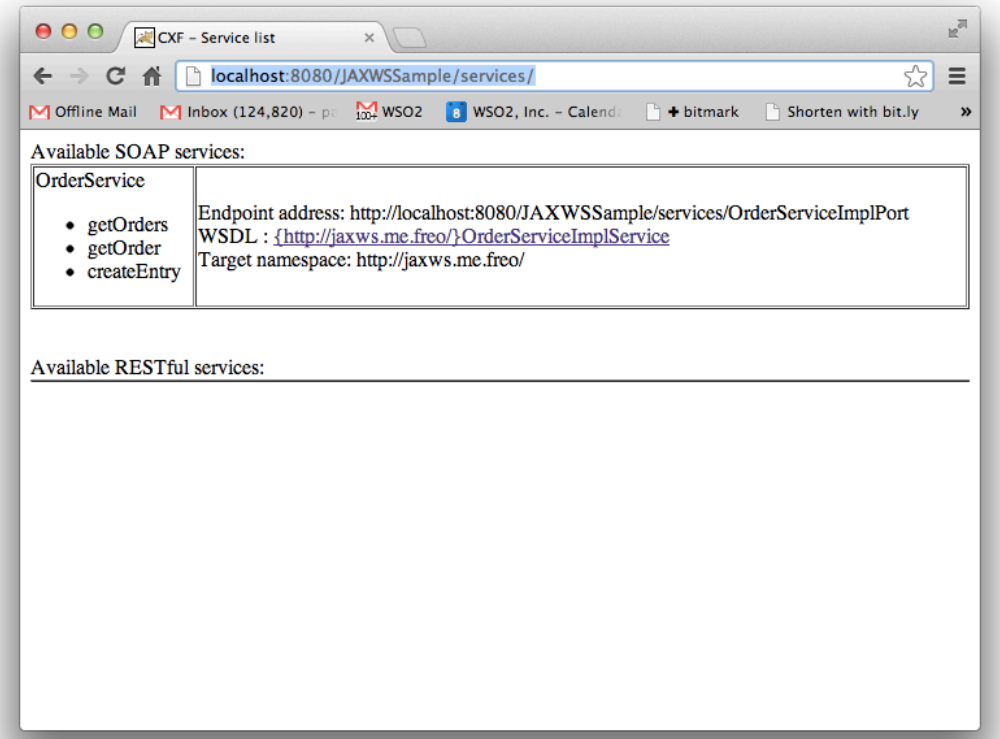


31. Choose Tomcat 7.0 and click **Next>**
    You should see your JAXWSSample app in the Configured column on the right:



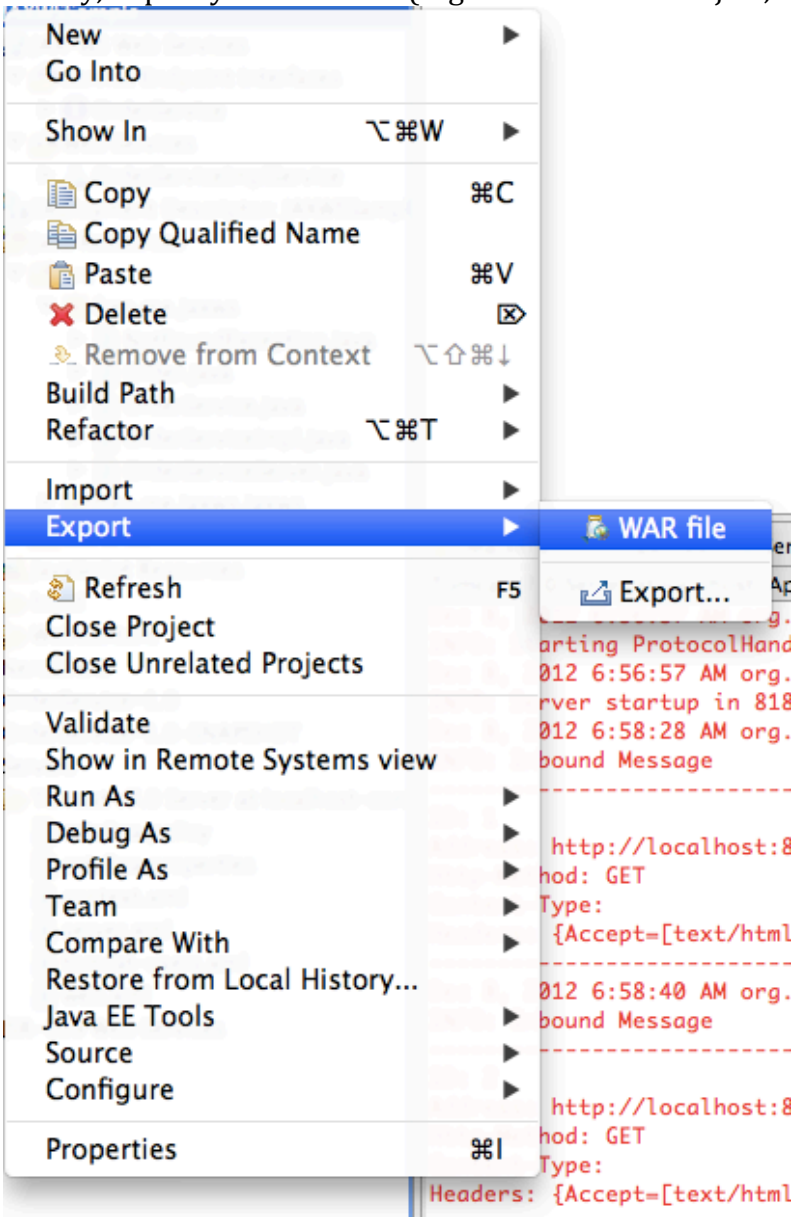
32. Click Finish
    The server will start up and Eclipse will try to start up a browser against your app. There will be a 404 Not Found error because our WAR file has no "Welcome Page".

33. Browse http://localhost:8080/JAXWSSample/services/ and you should see a CXF generated page:



34. Click on the link and it should take you the URL. Once again make sure everything is working using SOAPUI.

35. Finally, Export your WAR file (Right Click on the Project, Export->WAR File)



36. Stop the Server inside Eclipse, and copy your exported WAR file to Tomcat's **webapp** directory. Start Tomcat from the command line as before, and test again.

37. Congratulations if you got this far!

38. Extension Exercise.

    Refactor the code between the SOAP and REST services so there is a single model which is exposed as both SOAP and REST.