一、简介

与 *WebServices* 相关的 *J2EE* 技术称为 *JWS*（*Java WebServices*），其中含有 *JAX-WS*、*JAX-RS*、*JAXB*、*JAXR*、*SAAJ*、*StAX* 等技术。

支持 *SOAP* 的是 *JAX-WS*，即 *JSR 224*，*http://jcp.org/en/jsr/detail?id=224* →**JWS**
支持 *REST* 的是 *JAX-RS*，即 *JSR 311*，*http://jcp.org/en/jsr/detail?id=311* →*JRS*

- CXF——XFire 和 Celtix 的合并（一个由 IONA 赞助的开源 ESB，最初寄存在 ObjectWeb 上）。
- Jersey——Sun 公司的 JAX-RS 参考实现。
- RESTEasy——JBoss 的 JAX-RS 项目。
- Restlet——也许是最早的 REST 框架了，它 JAX-RS 之前就有了。 http://www.restlet.org/downloads/stable

下面是本人经过学习 webservices 而自作的一个说明文档，此文档仅仅适用于初学者，文档中将 webservices 的创建过程、配置过程、调用过程进行了讲述，本 webservices 的创建基于 JAX-WS，对于 xFire 现在已基本放弃，不建议大家使用 XFire，希望此文档可以帮到正在走弯路的童鞋,下面是对几种技术的简介；

1、JWS 是 Java 语言对 WebService 服务的一种实现，用来开发和发布服务。而从服务本身的角度来看 JWS 服务是没有语言界限的。但是 Java 语言为 Java 开发者提供便捷发布和调用 WebService 服务的一种途径。

2、Axis2 是 Apache 下的一个重量级 WebService 框架，准确说它是一个 Web Services / SOAP / WSDL 的引擎，是 WebService 框架的集大成者，它能不但能制作和发布 WebService，而且可以生成 Java 和其他语言版 WebService 客户端和服务端代码。这是它的优势所在。但是，这也不可避免的导致了 Axis2 的复杂性，使用过的开发者都知道，它所依赖的包数量和大小都是很惊人的，打包部署发布都比较麻烦，不能很好的与现有应用整合为一体。但是如果你要开发 Java 之外别的语言客户端，Axis2 提供的丰富工具将是你不二的选择。

3、XFire 是一个高性能的 WebService 框架，在 Java6 之前，它的知名度甚至超过了 Apache 的 Axis2，XFire 的优点是开发方便，与现有的 Web 整合很好，可以融为一体，并且开发也很方便。但是对 Java 之外的语言，没有提供相关的代码工具。XFire 后来被 Apache 收购了，原因是它太优秀了，收购后，随着 Java6 JWS 的兴起，开源的 WebService 引擎已经不再被看好，渐渐的都败落了。

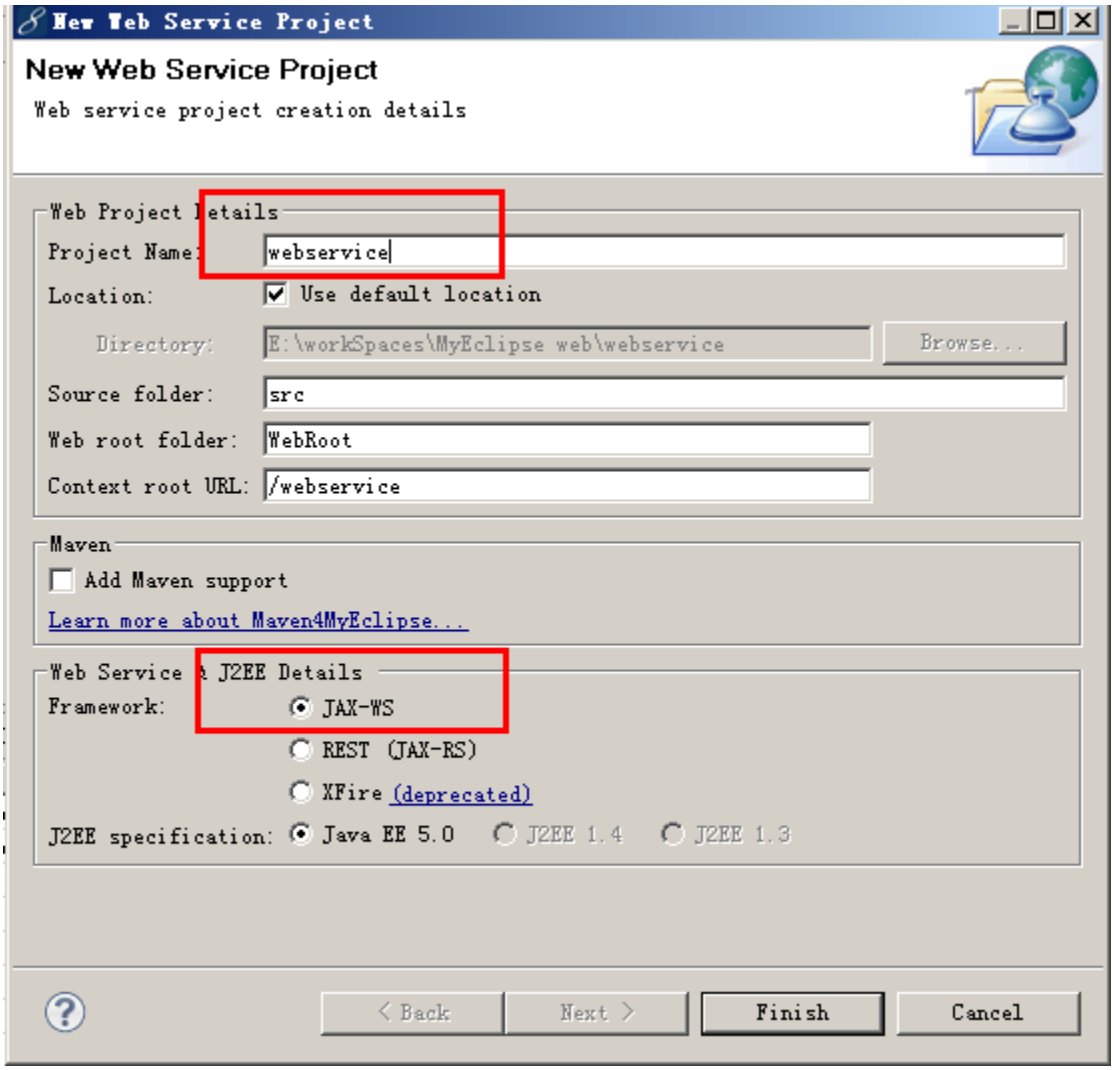
4、CXF 是 Apache 旗下一个重磅的 SOA 简易框架，它实现了 ESB（企业服务总线）。CXF 来自于 XFire 项目，经过改造后形成的，就像目前的 Struts2 来自 WebWork 一样。可以看出 XFire 的命运会和 WebWork 的命运一样，最终会淡出人们的视线。CXF 不但是一个优秀的 Web Services / SOAP / WSDL 引擎，也是一个不错的 ESB 总线，为 SOA 的实施提供了一种选择方案，当然他不是最好的，它仅仅实现了 SOA 架构的一部分。

基于以上的认识，我们可以得知，虽然有了 Java6，但是我们还可以选择 Axis2、XFire、CXF 等。我们不能指望有了 Java6 JWS，就能异想天开去实施 SOA。如果要与别的语言交互，也许我们还有赖于 Axis2 等等，当然这不是唯一选择，仅仅是一种可供选择的方案。

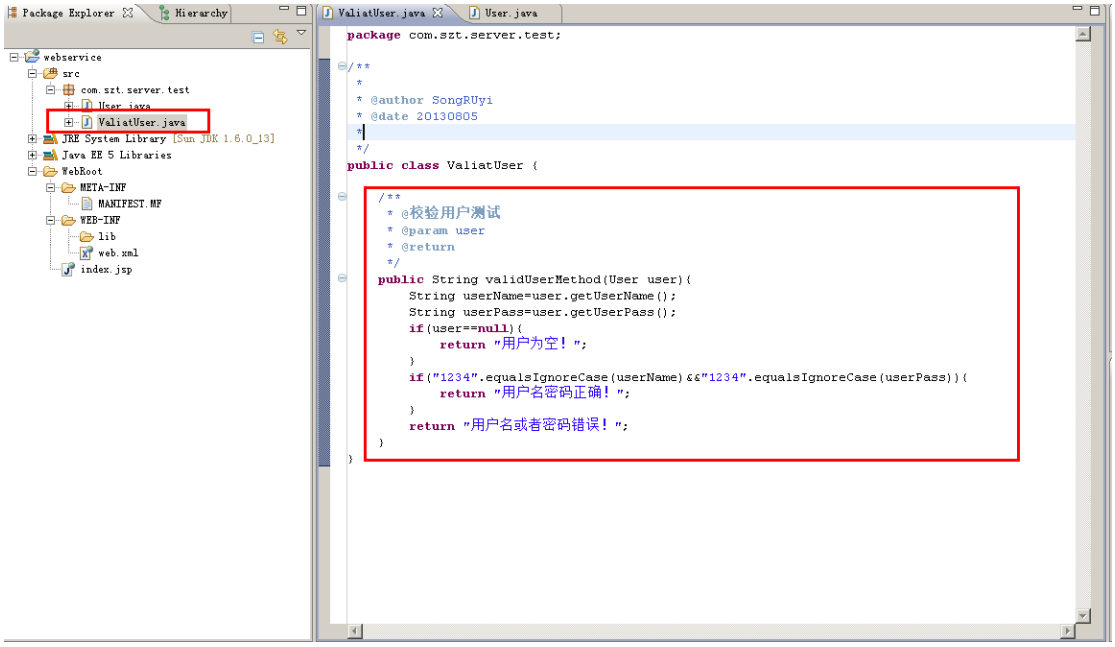
还有，目前很多企业的应用还是基于 Java5 的，而 Java5 的项目不会瞬间都升级到 Java6，如果要在老项目上做扩展，我们还有赖于其他开源的 WS 引擎。

二、*Myeclipse8.5* 创建服务器端步骤

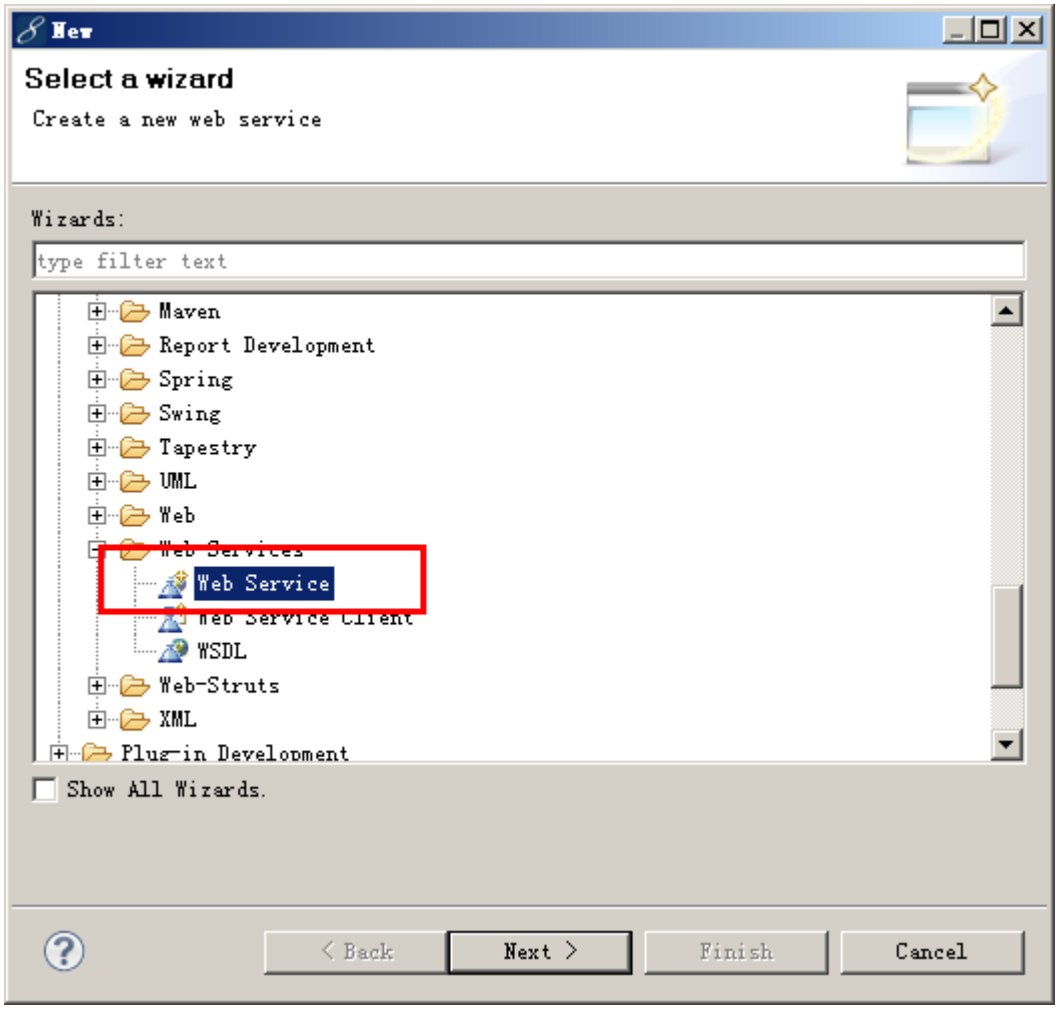
1. 创建 *webservices* 项目；



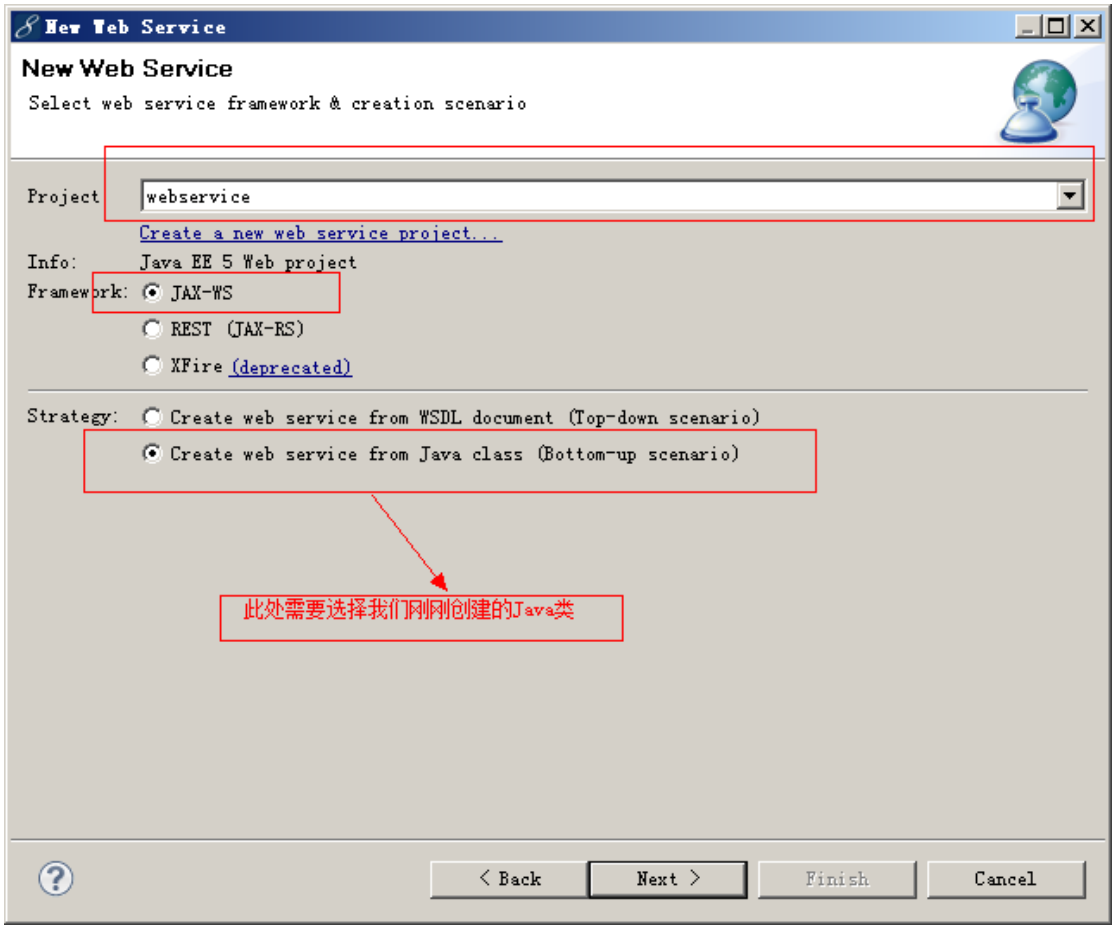
2. 创建 *Class* 类 →书写 *ValiatUser*，书写接口方法；



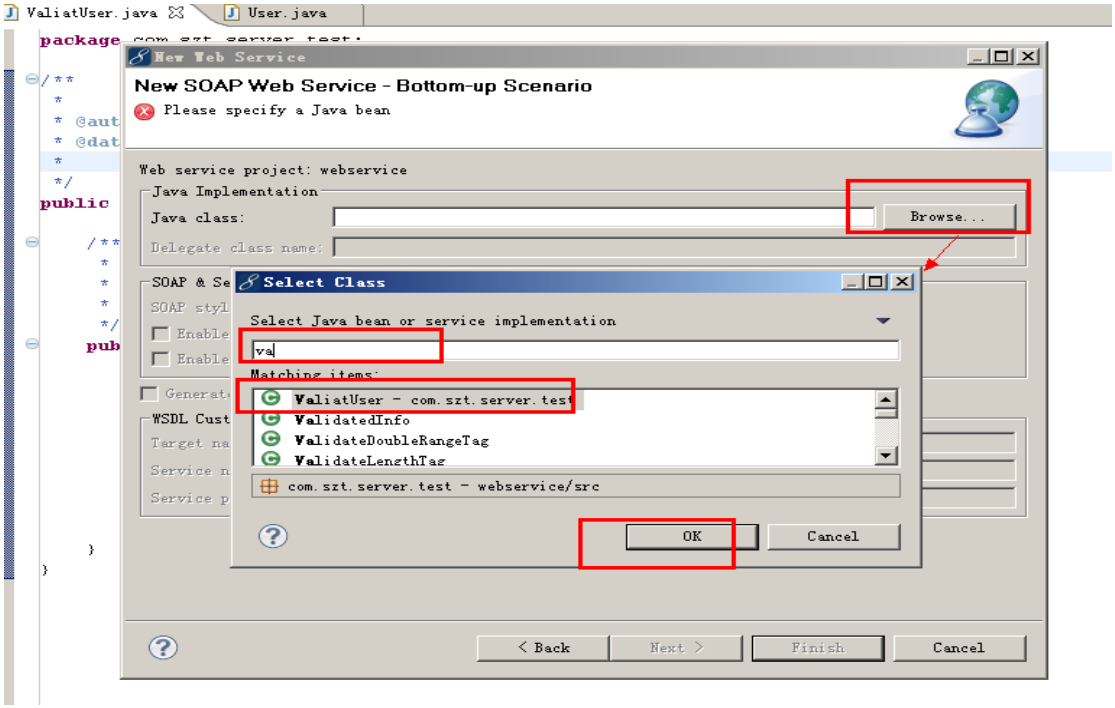
3. 类和方法写好之后，在 *src* 上右键——*other*——*MyEclipse*——*Web Services* 下选中 *Web Service*，然后 *Next,*创建 *webservices* 选择



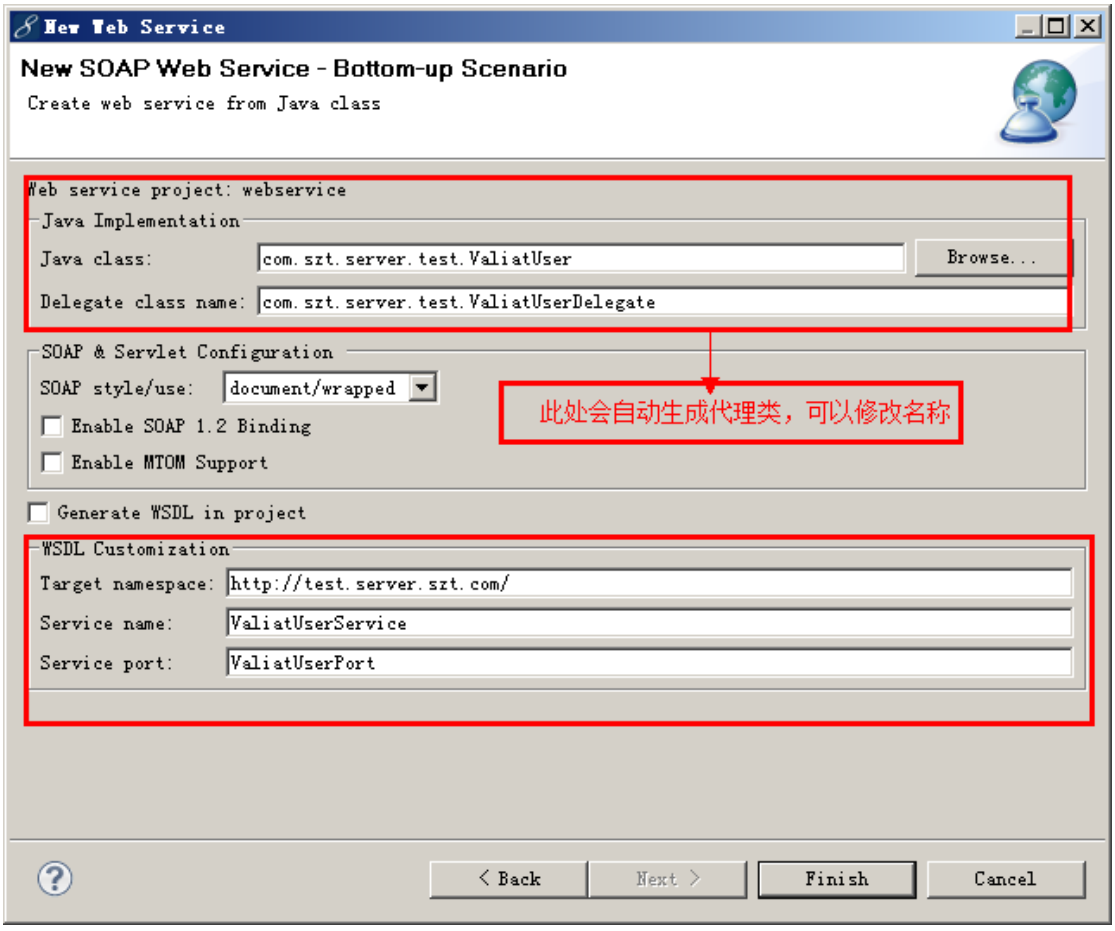
4. 选中 *webservices* 项目，框架选择 *jax-ws*，策略选择第二项：意思是从已经建好的项目中选择 *class*



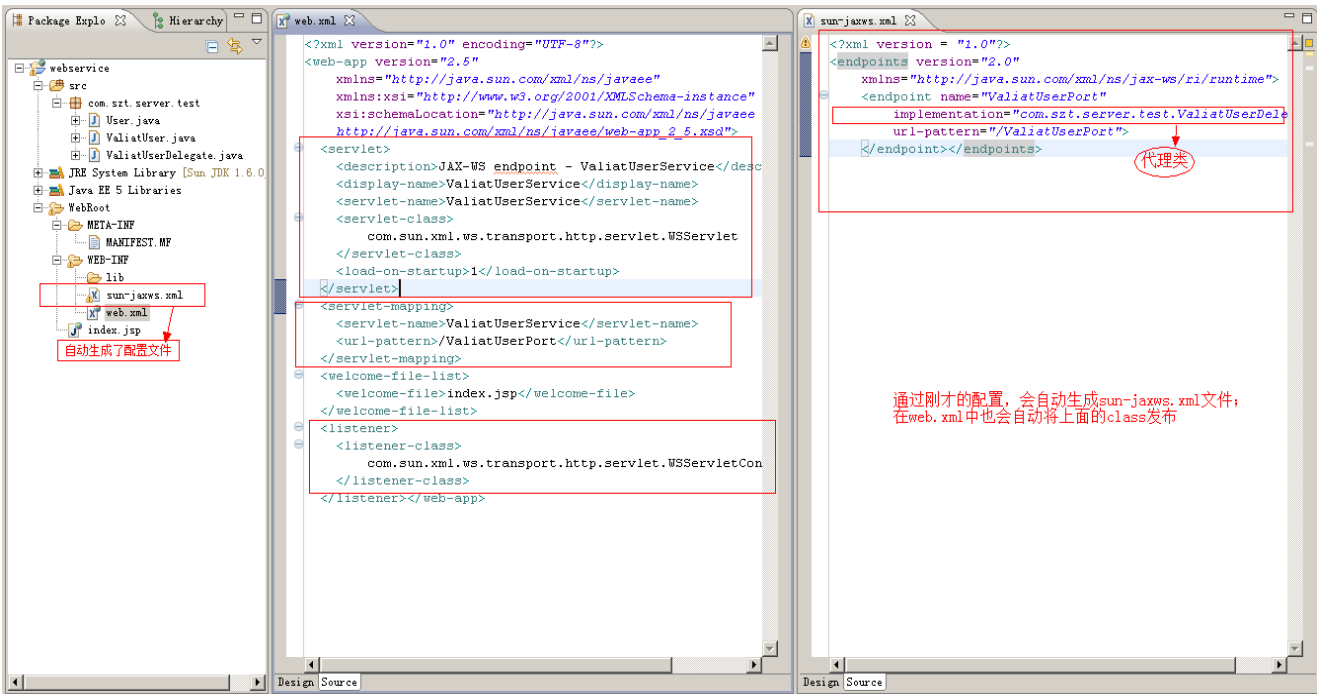
5. 此处选择我们已经书写好的类，点击 *finish* 完成；



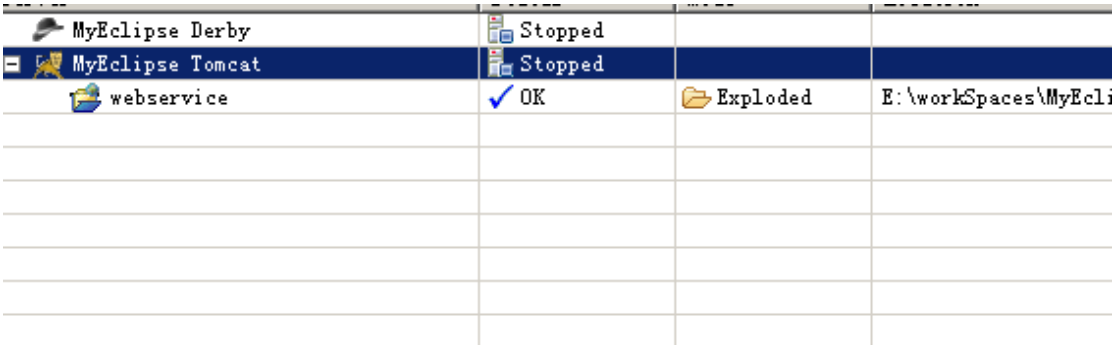
6. 选择我们已经写好的实现类，工具会自动生成代理类，自动生成 *wsdl* 选择卡中的选项，无特殊修改，可以不修改；

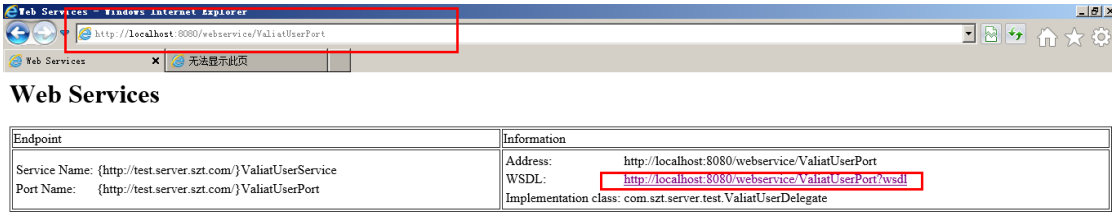


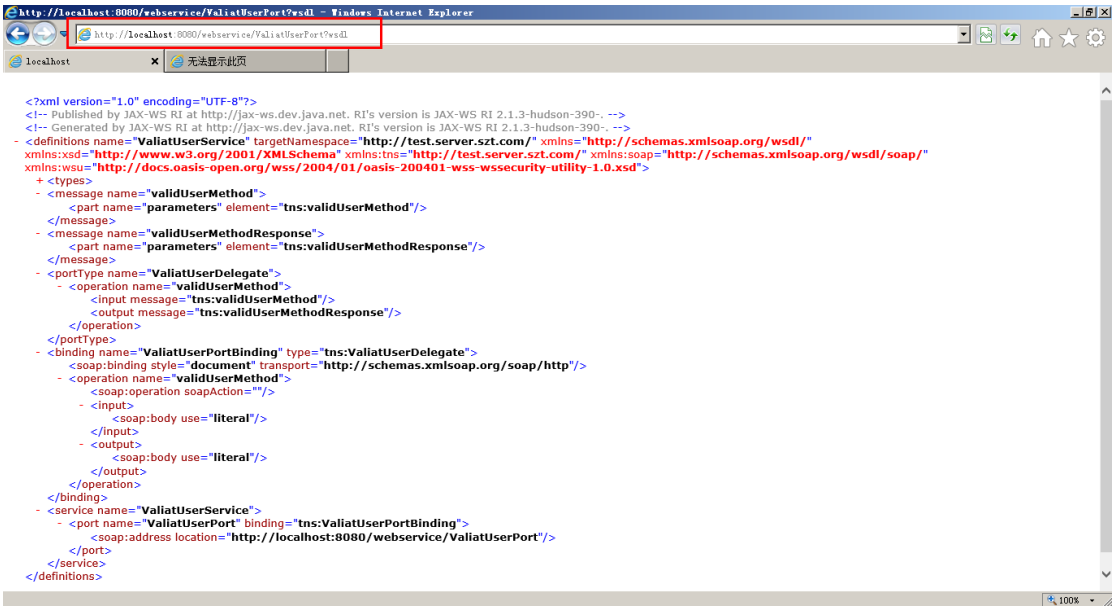
7. 工具会自动生成所有必须文件，包括 *web.xml* 启动配置，*sun-jaxws.xml*；



8. 此时启动服务器进行项目发布：



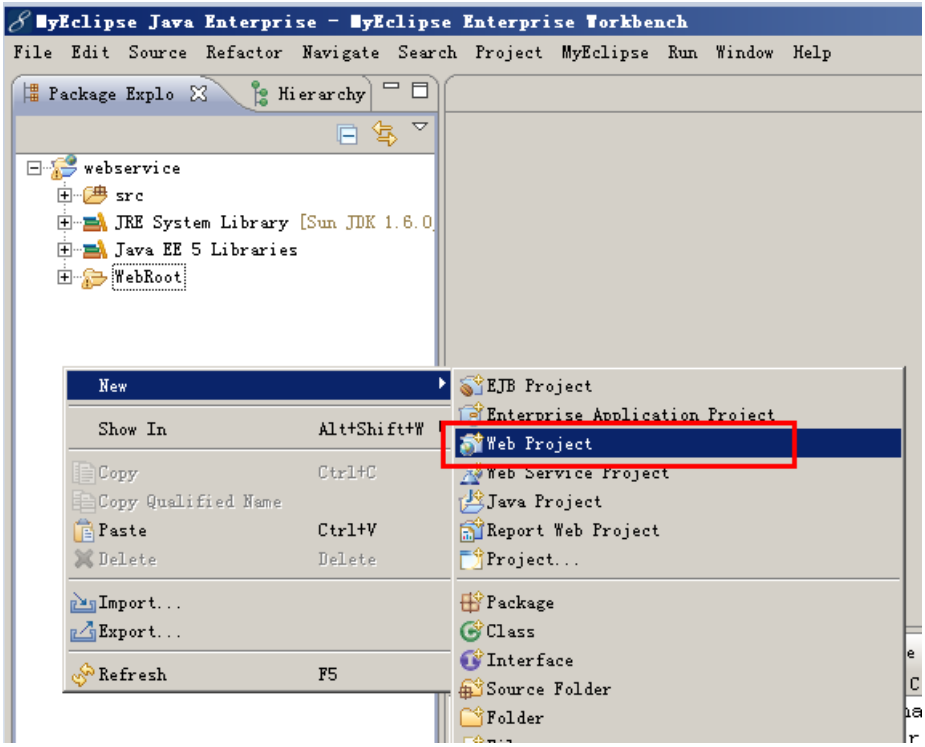




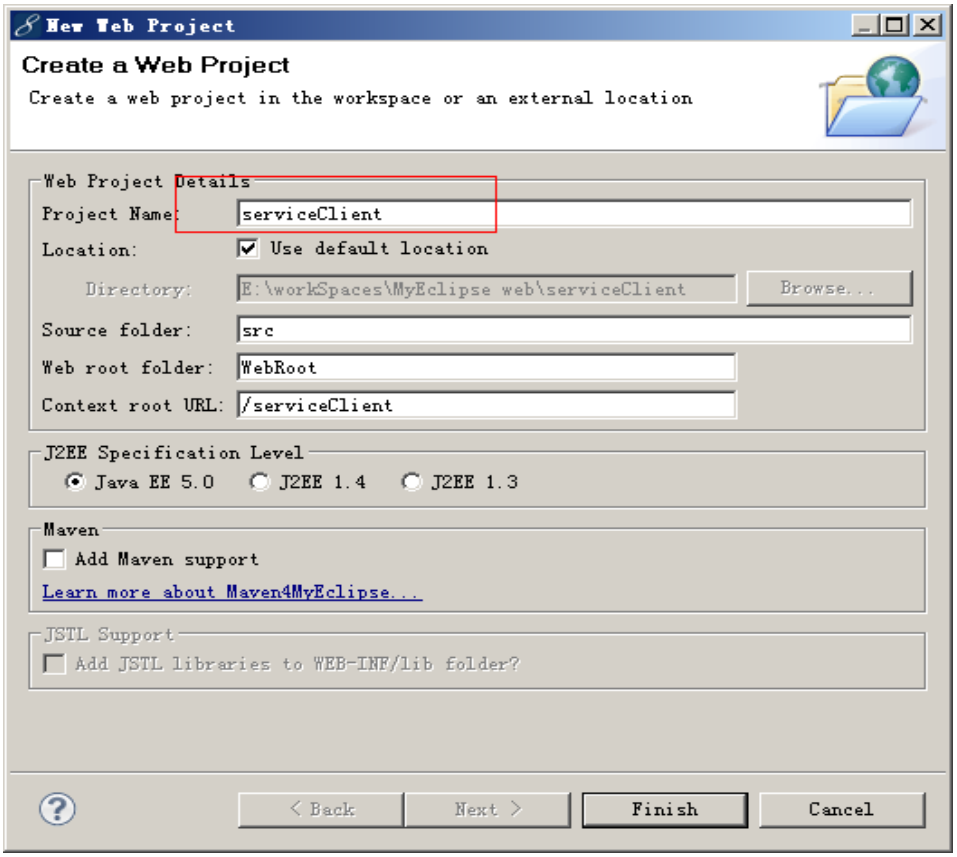
此时说明数据库发布成功；

三、 创建客户端：

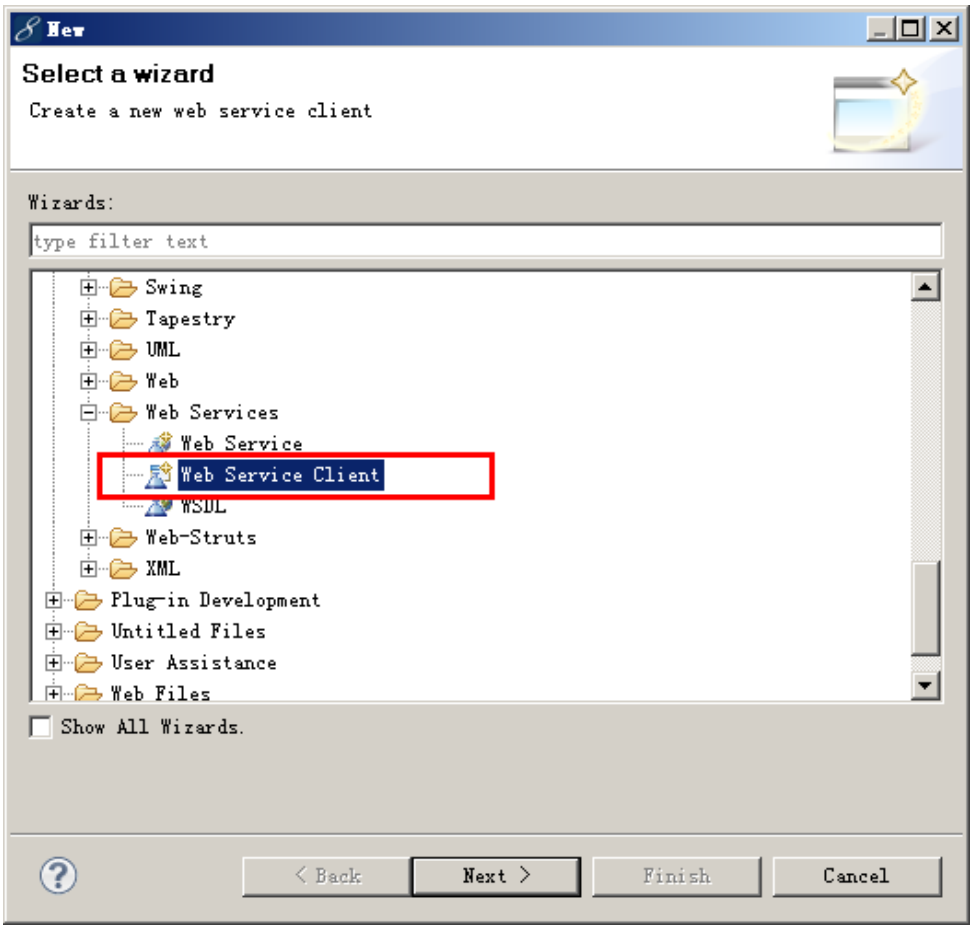
1. 创建 *web* 项目；



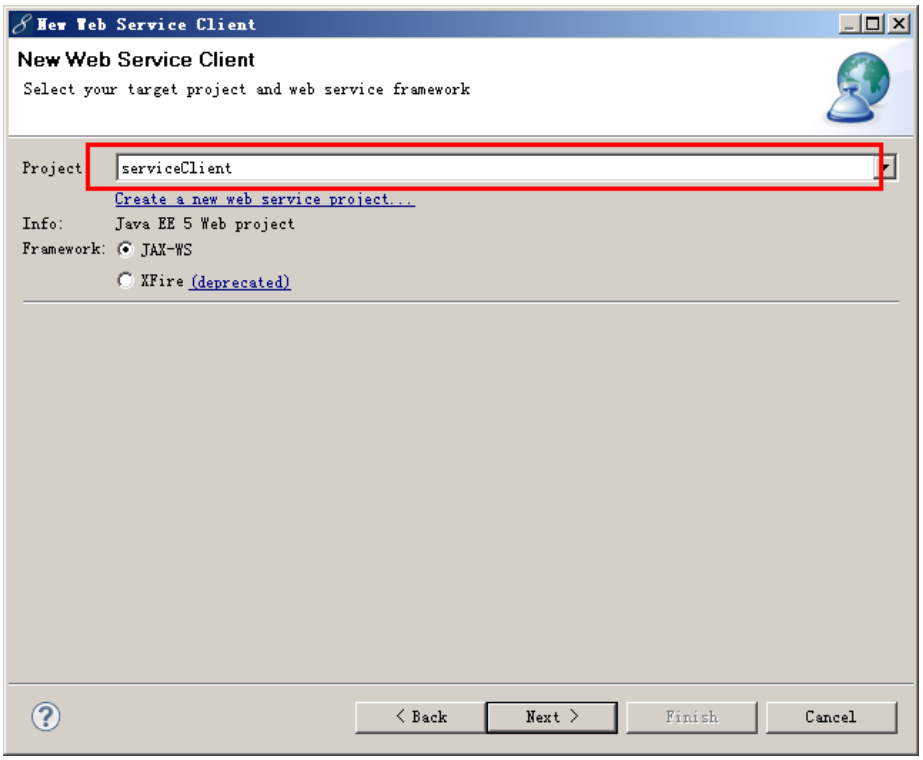
添加项目的名称



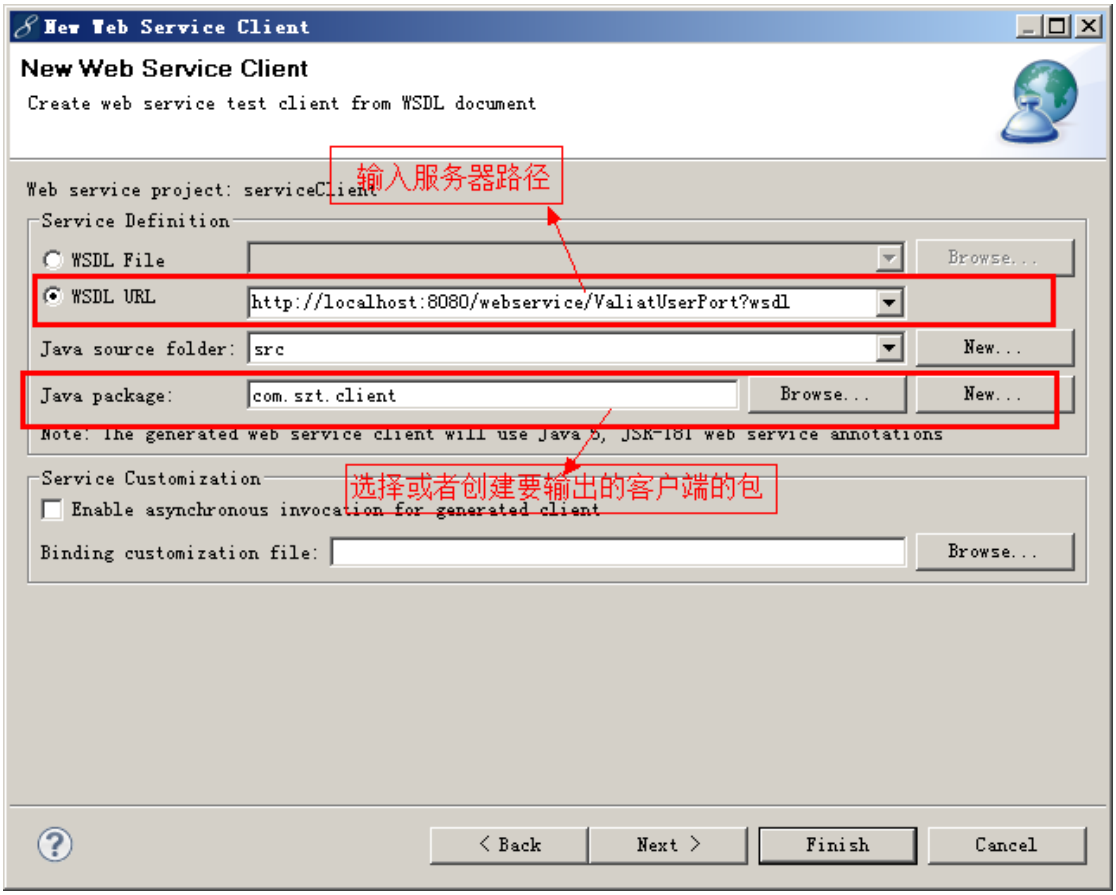
2. 创建 *webServices Client* 项目，然后点击下一步；



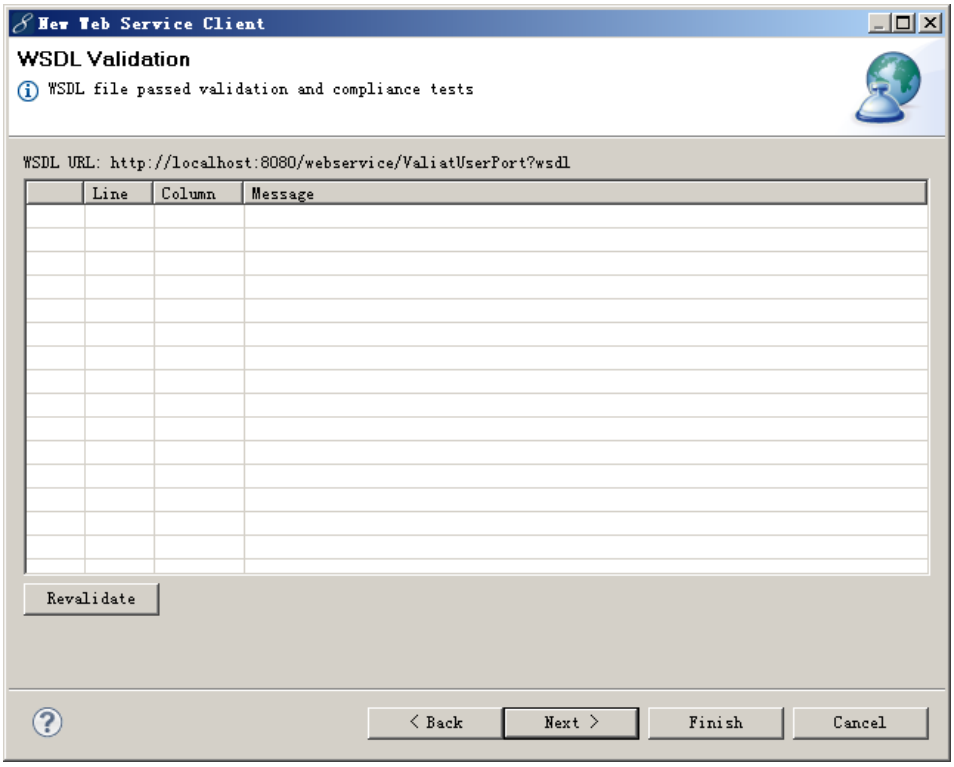
3. *project* 选择当前项目，框架选择 *jax-ws*：



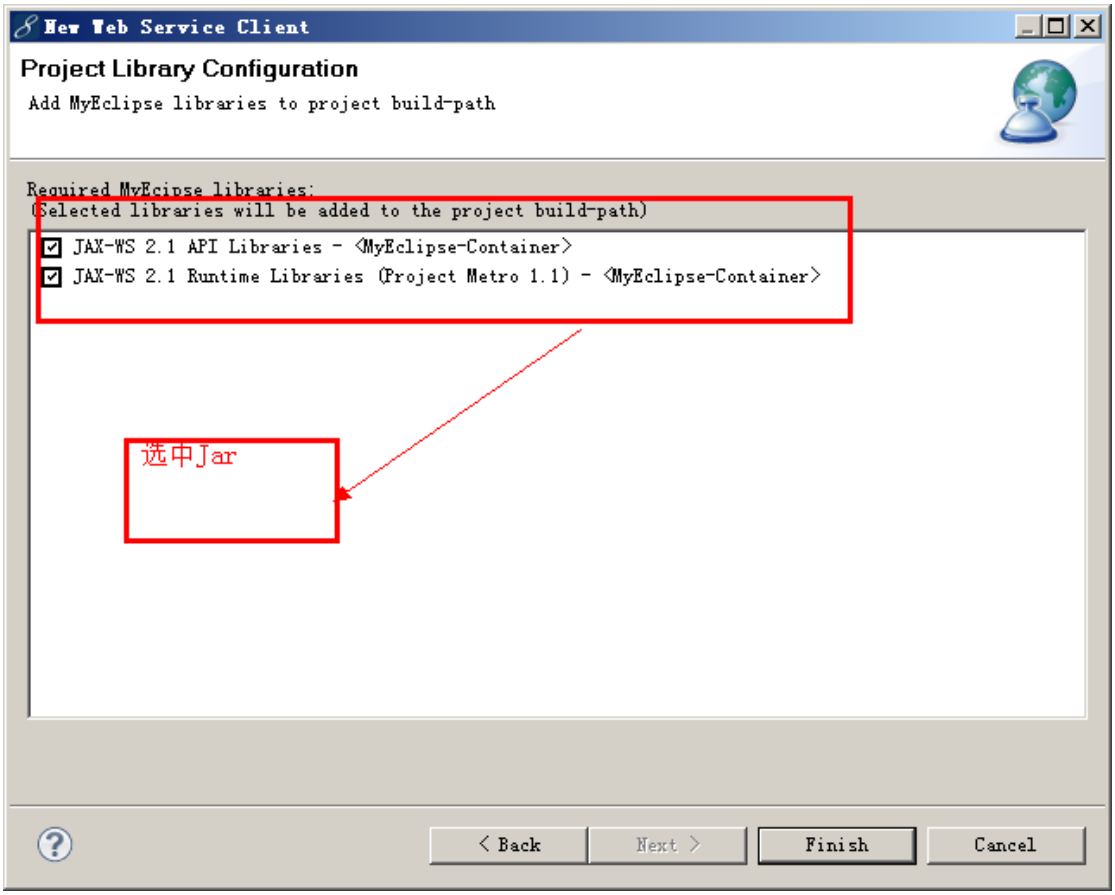
4. 输入服务器的 *wsdl* 路径，选择输出的包路径，点击下一步：



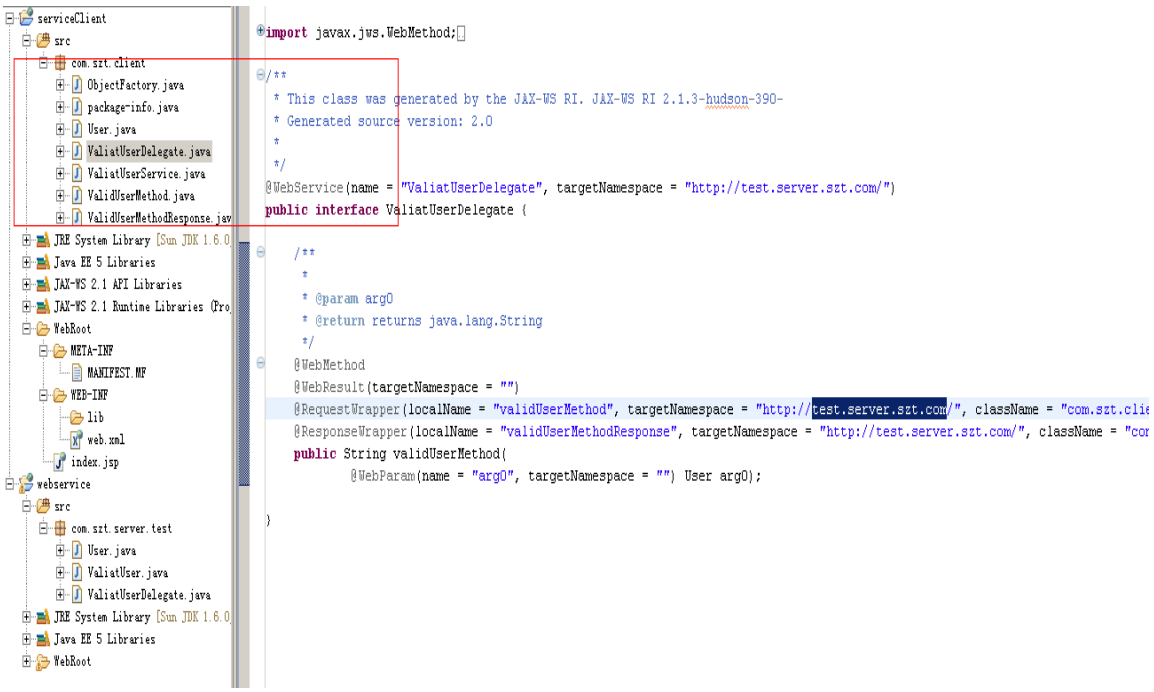
5. 校验文件：



6. 校验没有问题，选中 *jax-ws* 的 *Jar* 包，点击 *finish*：



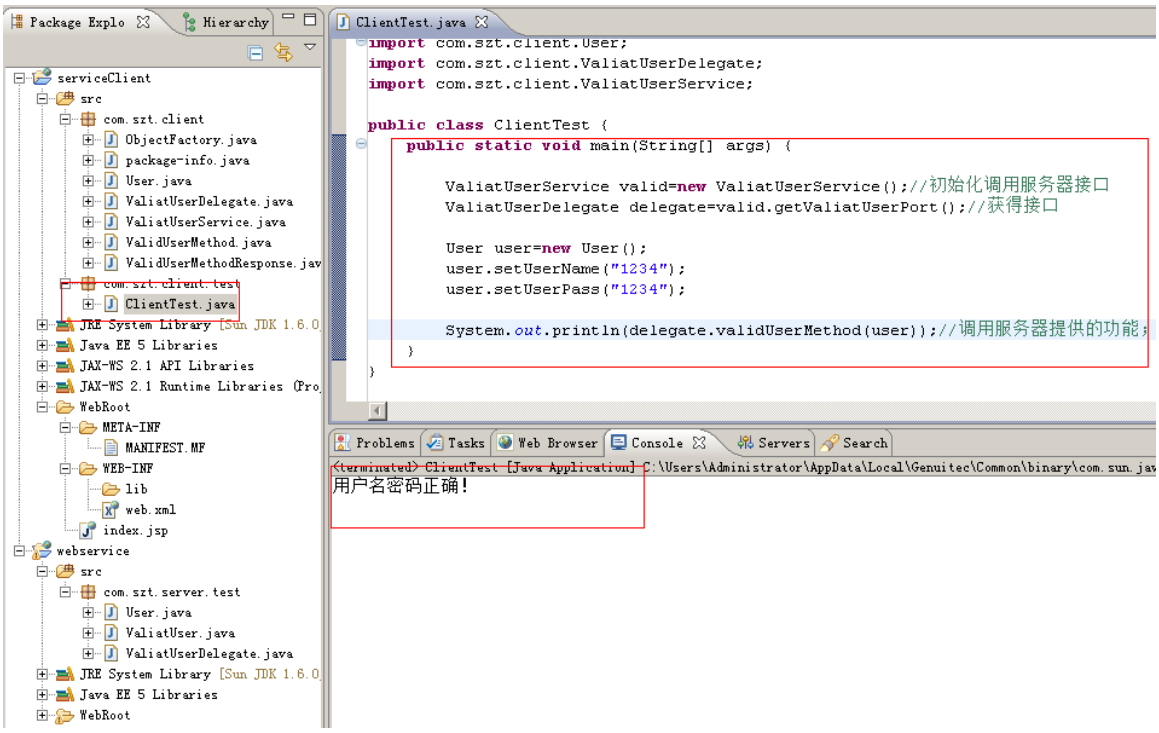
7. 点击 *finish* 系统会自动所有需要调用服务器端必须的文件：



8. 创建测试类，对服务器的功能进行调用：

```
ValiatUserService valid=new ValiatUserService();//初始化调用服务器接口
ValiatUserDelegate delegate=valid.getValiatUserPort();//获得接口\

    User user=new User();
    user.setUserName("1234");
    user.setUserPass("1234");
    System.out.println(delegate.validUserMethod(user));
```

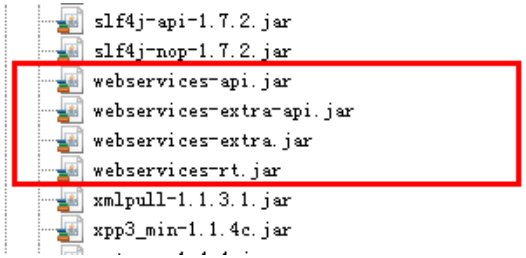


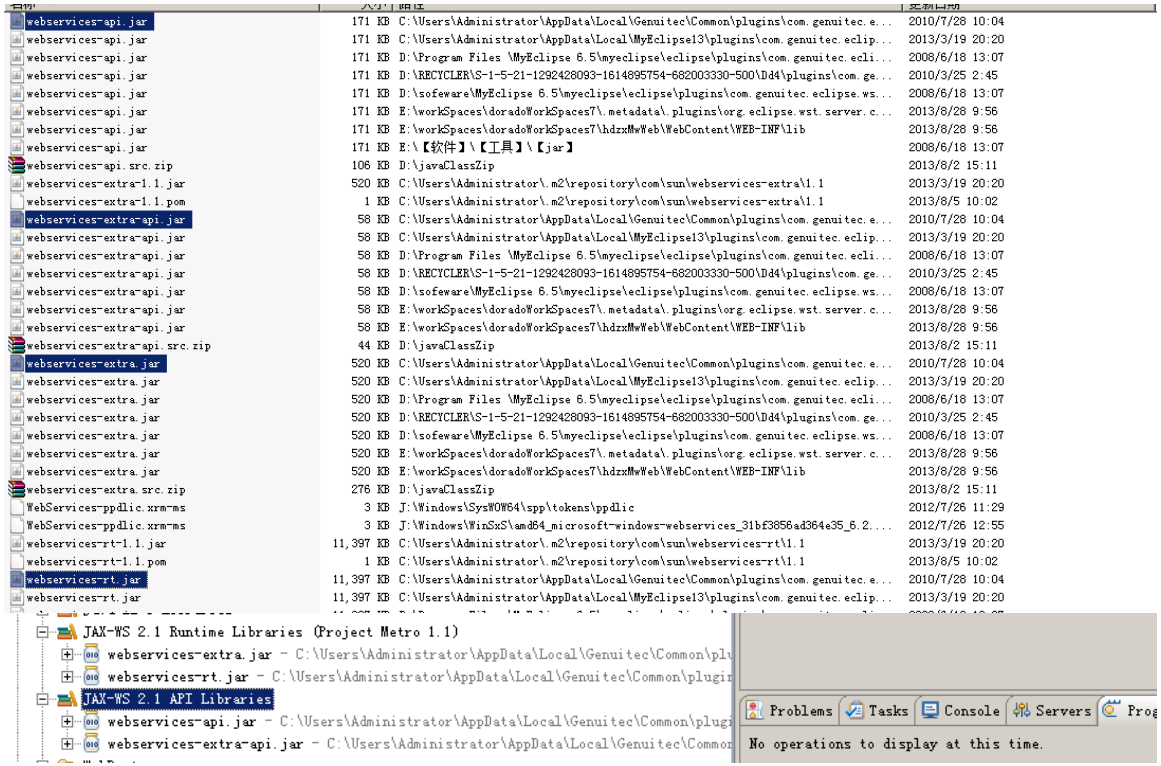
四、问题：

```
严重: Error configuring application listener of class
com.sun.xml.ws.transport.http.servlet.WSServletContextListener
java.lang.ClassNotFoundException: com.sun.xml.ws.transport.http.servlet.WSServletContextListener
    at org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1516)
    at org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1361)
    at org.apache.catalina.core.StandardContext.listenerStart(StandardContext.java:3915)
    at org.apache.catalina.core.StandardContext.start(StandardContext.java:4467)
    at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:791)
    at org.apache.catalina.core.ContainerBase.addChild(ContainerBase.java:771)
    at org.apache.catalina.core.StandardHost.addChild(StandardHost.java:546)
    at org.apache.catalina.startup.HostConfig.deployDirectory(HostConfig.java:1041)
    at org.apache.catalina.startup.HostConfig.deployDirectories(HostConfig.java:964)
    at org.apache.catalina.startup.HostConfig.deployApps(HostConfig.java:502)
    at org.apache.catalina.startup.HostConfig.start(HostConfig.java:1277)
    at org.apache.catalina.startup.HostConfig.lifecycleEvent(HostConfig.java:321)
```

解决方案：遇到这个问题肯定是因为包的问题，此时解决方法有两种：

1. 下载 *JAX-WS* 架包，放置到 *tomcat* 的 *lib* 下面 http://jax-ws.java.net/.；
2. 另外一种使用 *myeclipse* 自带的 *tomcat*；





五、注解：关于注解的具体含义，本文档不再详述，大家进入此站点：*http://418684644-qq-com.iteye.com/blog/1208149* 进行了解学习；