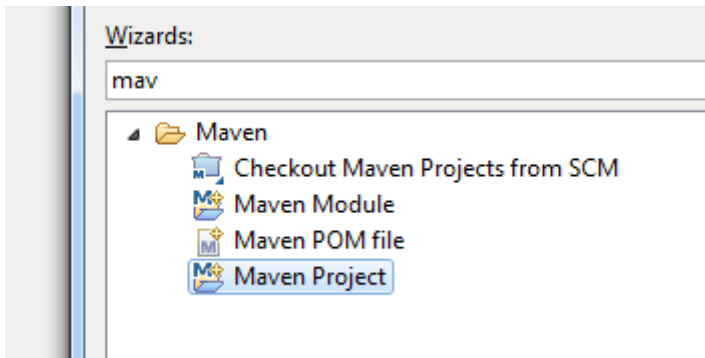
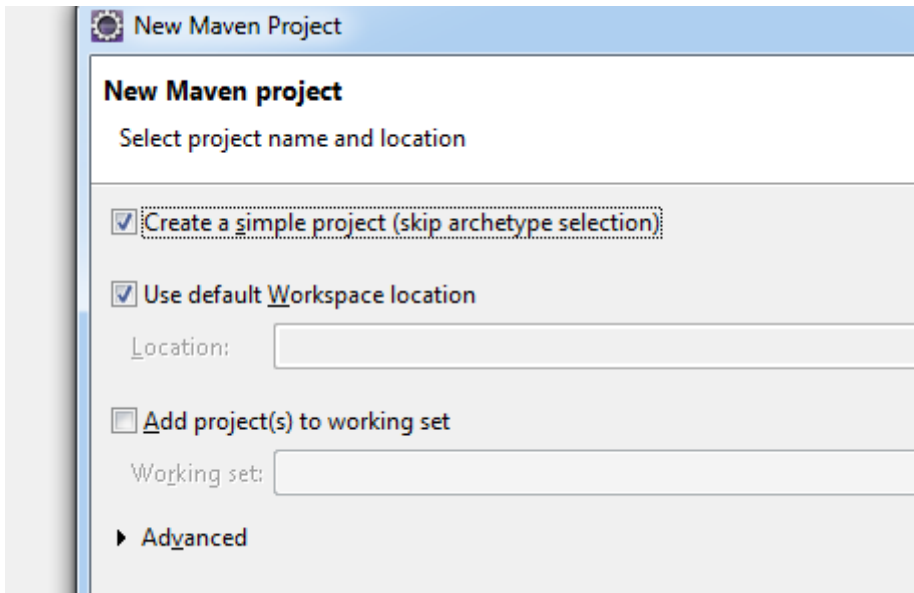


参考 <http://www.kancloud.cn/digest/lunaticssm/122063>
<http://www.myexception.cn/eclipse/1927141.html>

首先创建一个Maven的Project如下图



我们勾选上Create a simple project （不使用骨架）



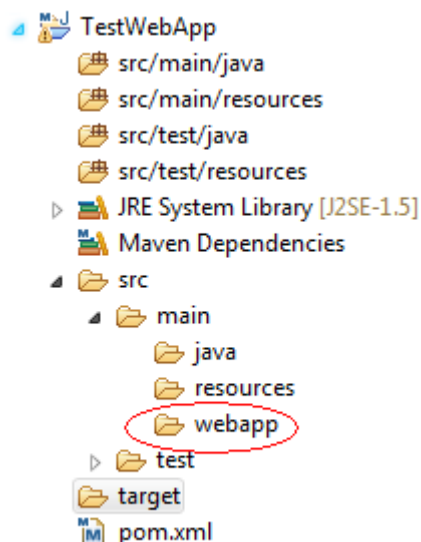
这里的Packing 选择 war的形式

New Maven project

Configure project

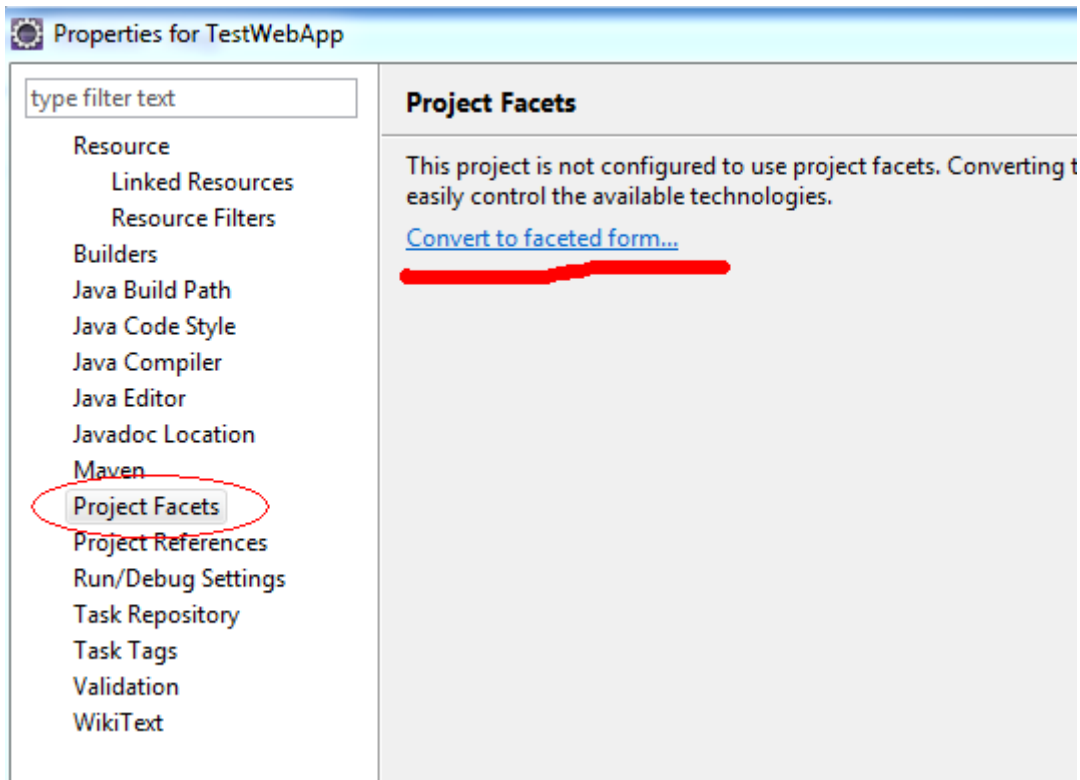
Artifact	
Group Id:	com.zhaoyang
Artifact Id:	TestWebApp
Version:	0.0.1-SNAPSHOT
Packaging:	war
Name:	
Description:	
Parent Project	
Group Id:	
Artifact Id:	
Version:	

由于packaging是war包，那么下面也就多出了webapp的目录

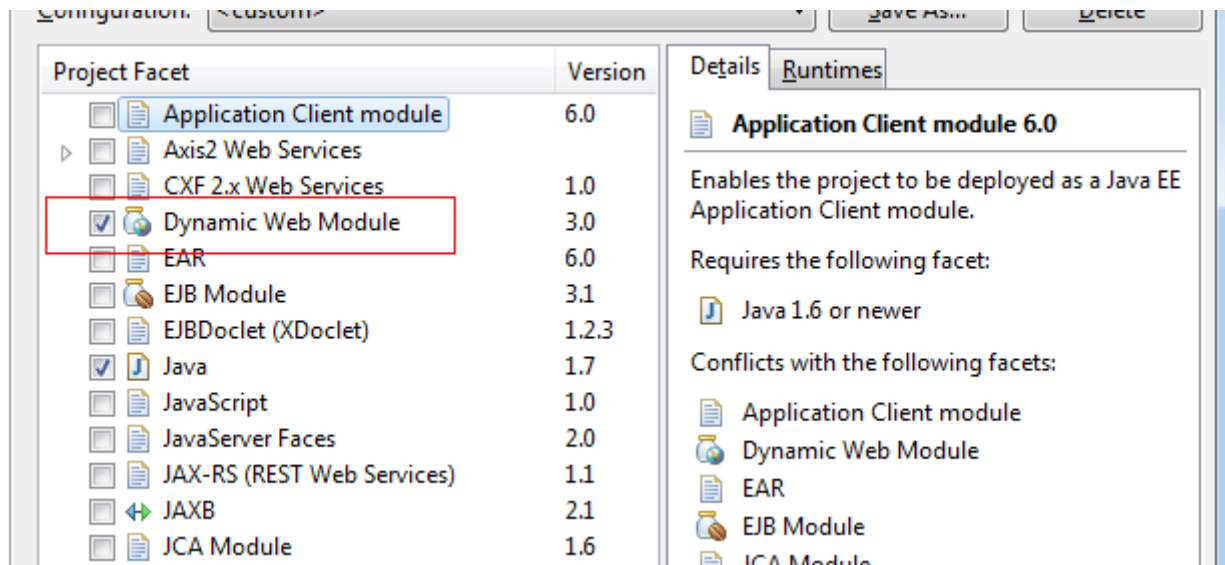


由于我们的项目要使用eclipse发布到tomcat下面，这里我们需要先把项目转成dynamic web project

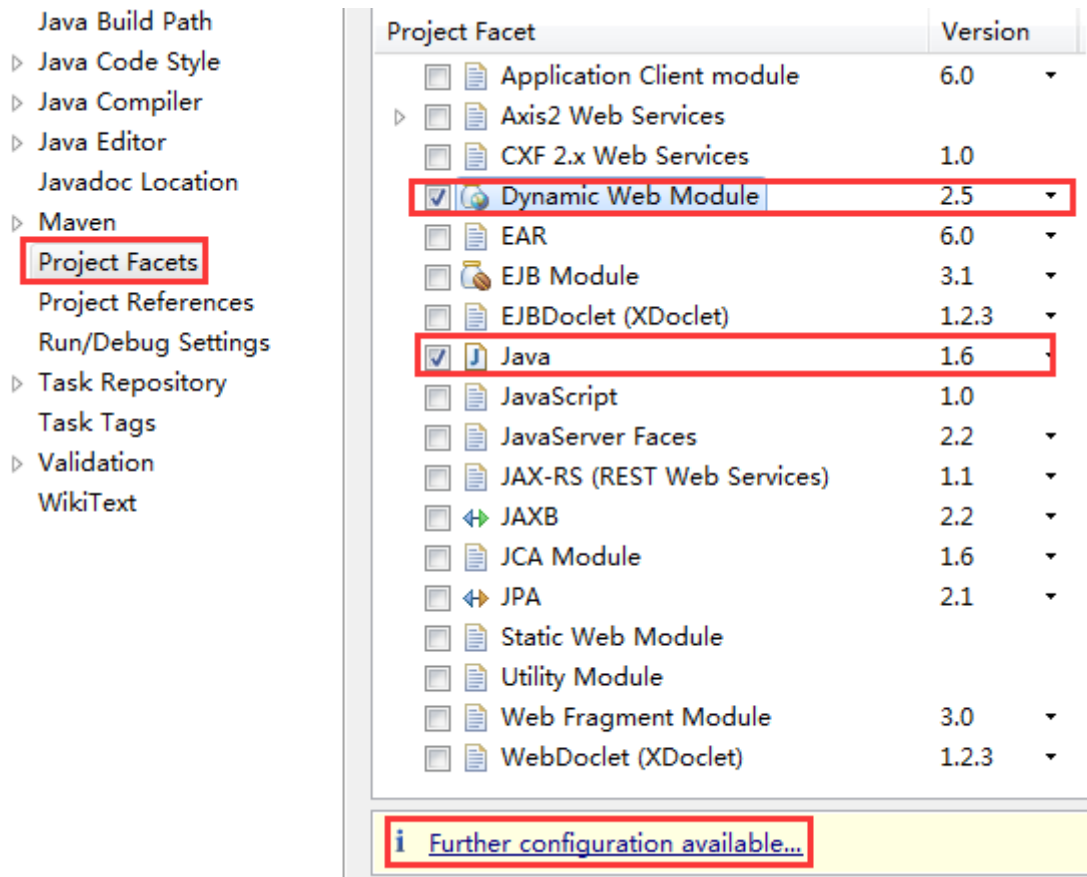
在我们的项目上点击右键，选择properties 并找到 Project Facets ，并点击Convert to faceted form... 如下图：



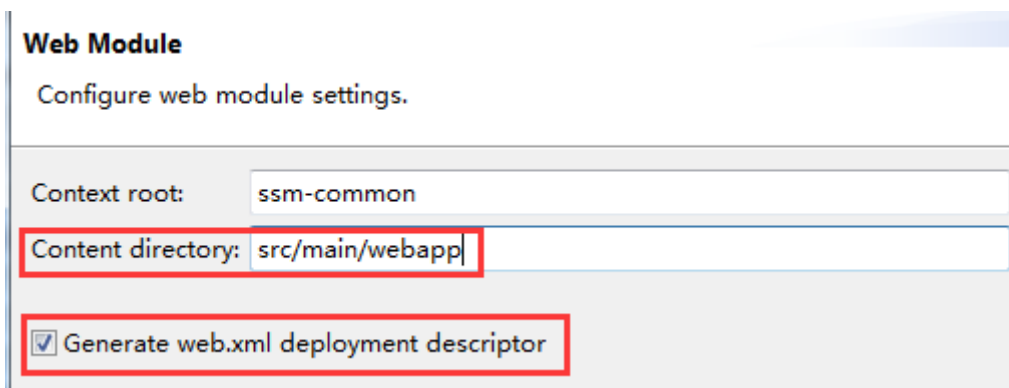
然后勾选Dynamic Web Module 并点击ok 如下图：（3.0只有tomcat7才支持）



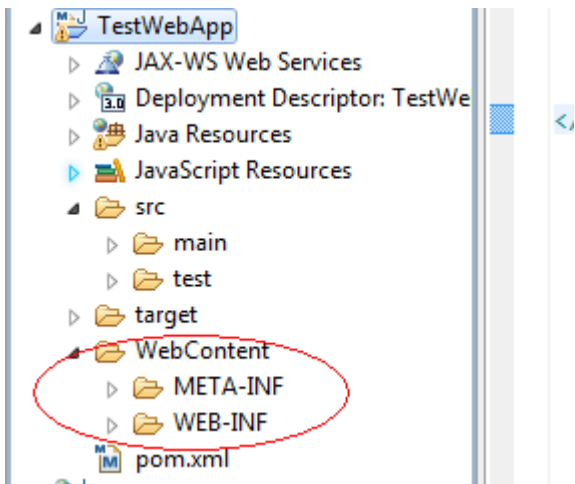
加载完成之后，选中Dynamic Web Module，版本号选择2.5，如果使用的是Tomcat7和JDK7可以选择3.0，在点击Further configuration available...进行配置：



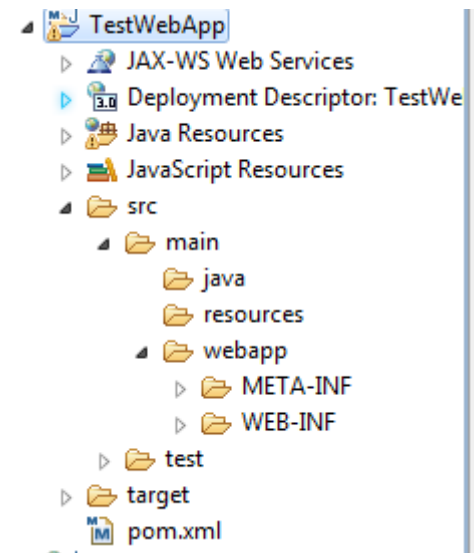
3.配置Content directory:"src/main/webapp", 并勾选生成web.xml的选项:



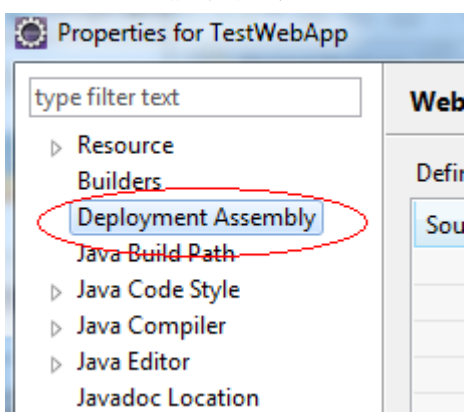
接下来观察我们的项目结构，多了一个web content目录



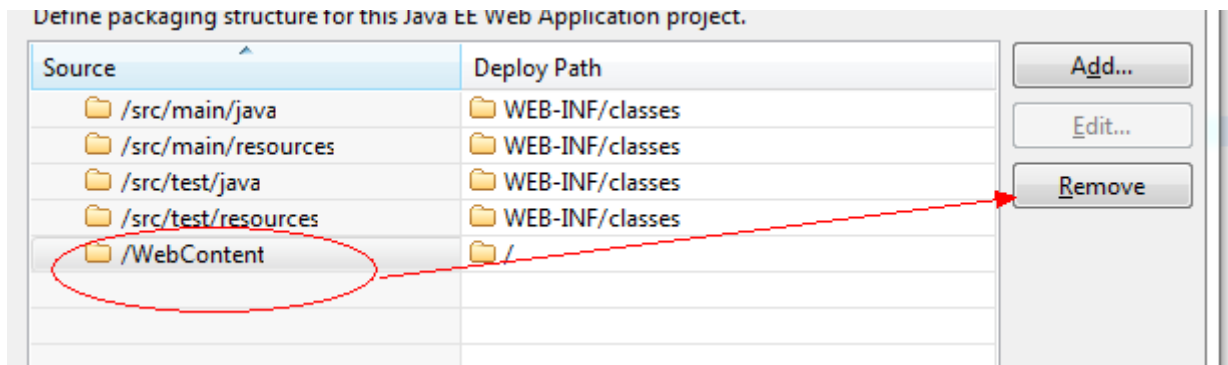
虽然此时我们可以发布到tomcat中，但这不符合maven的结构，我们还要做如下修改
把上图WebContent下面两个目录 META-INF ， WEB-INF 直接剪切到src/main/webapp目录下，并删掉WebContent目录，那么现在的项目结构如下图：



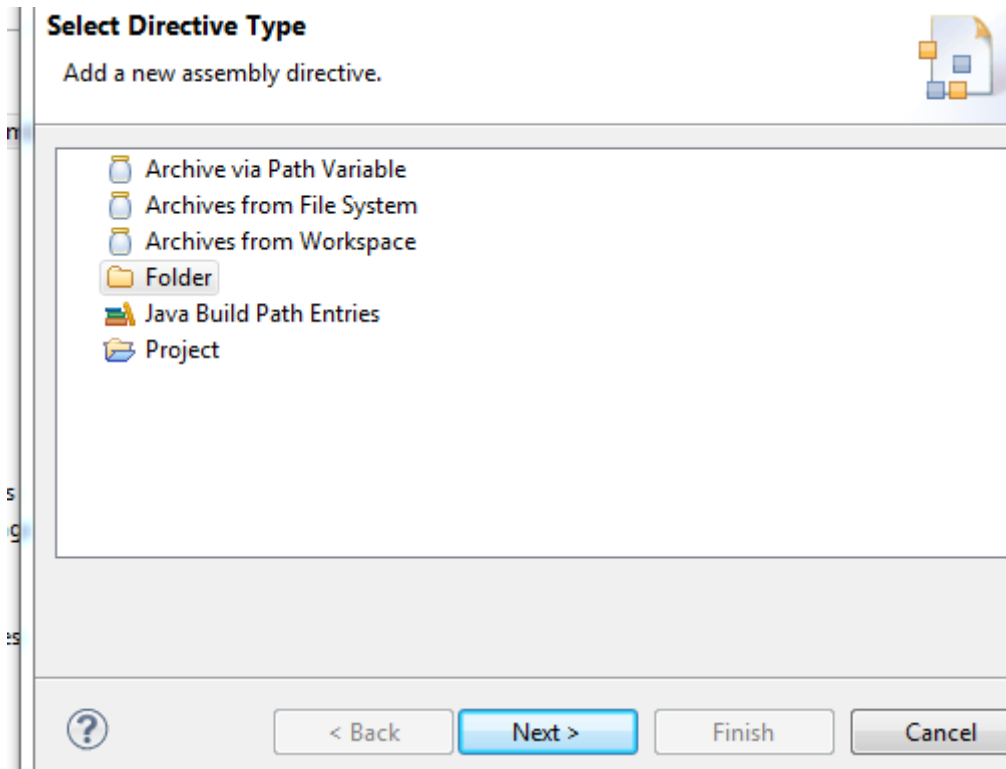
然后我们要修改发布规则，右键点击项目， 选择 Deployment Assembly



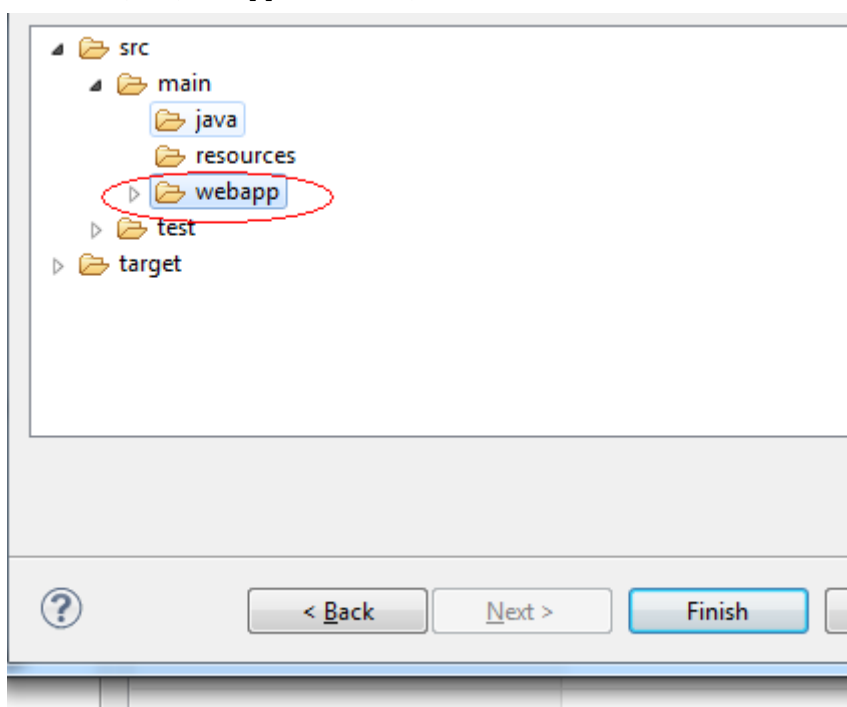
选择WebContent，把它remove掉，测试类我们也不需要发布，test的两个目录页可以remove



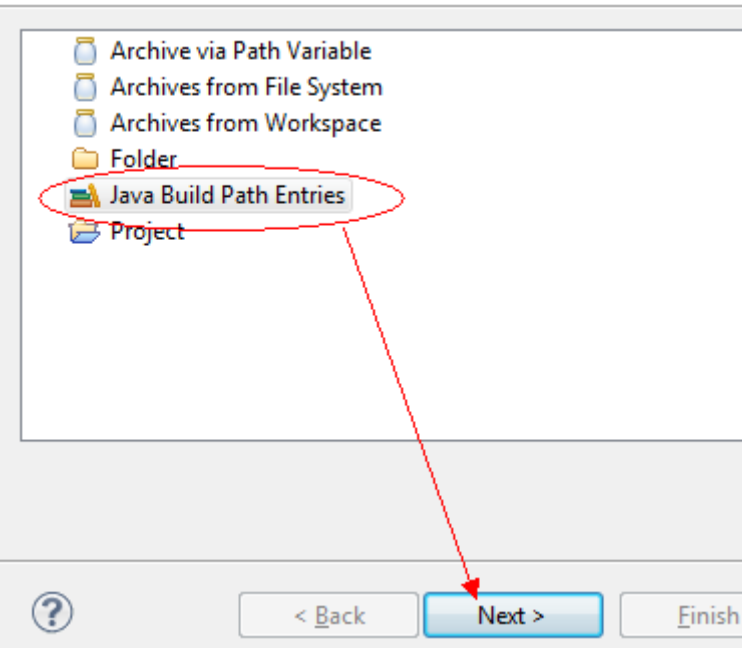
接着重新指定一个web的路径，点击add，选择Folder --> next



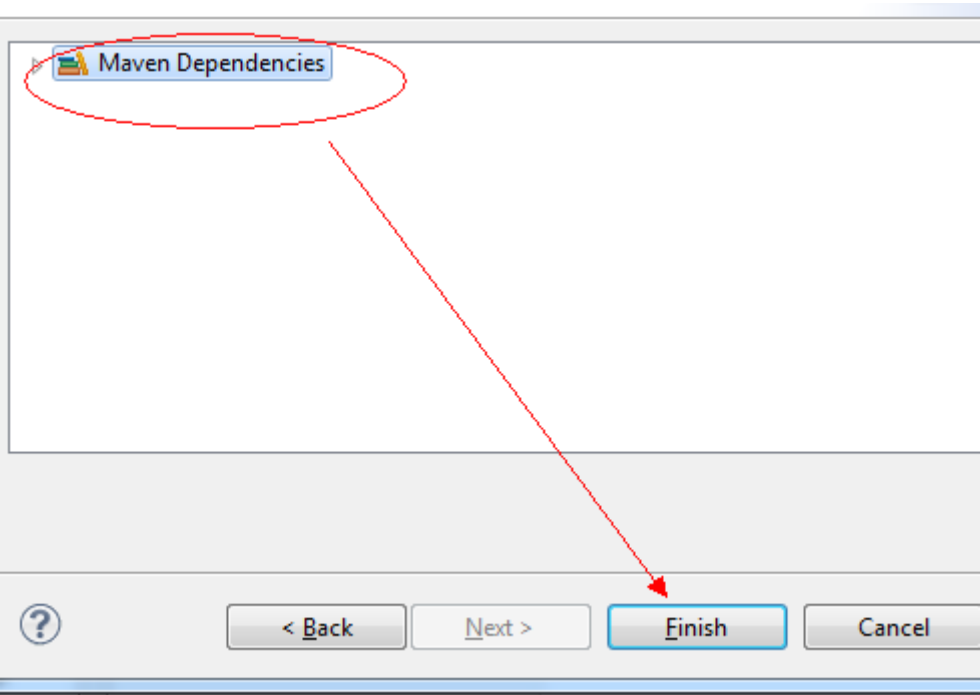
在src下找到webapp目录，然后finish



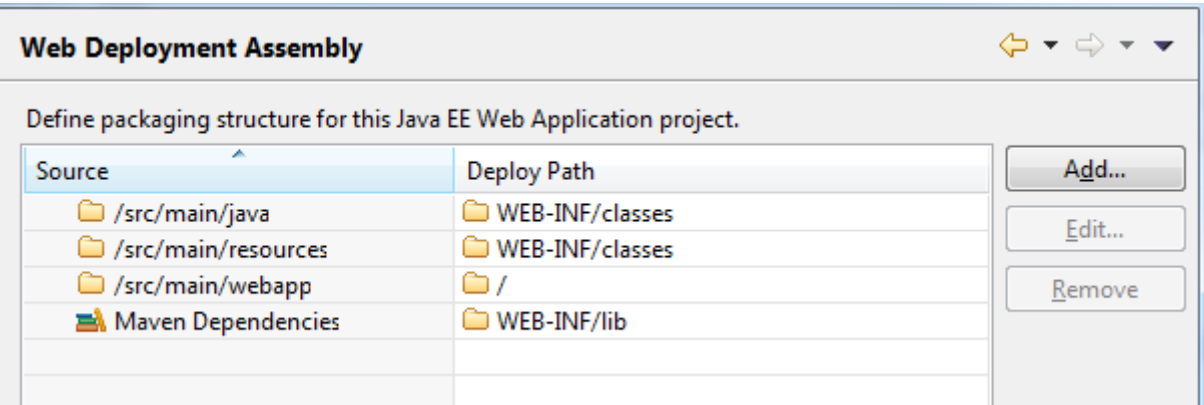
最后一步，我们要把当前的build path 指向 Maven Dependency， 直接点击add， 选择Java Build Path Entries 然后next



然后再点击finish完成

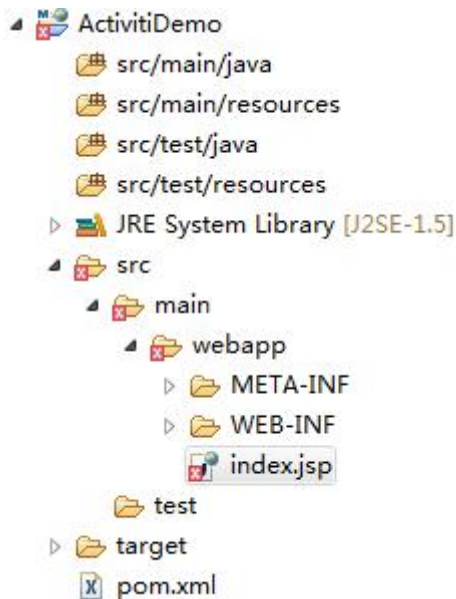


完成后如下图:

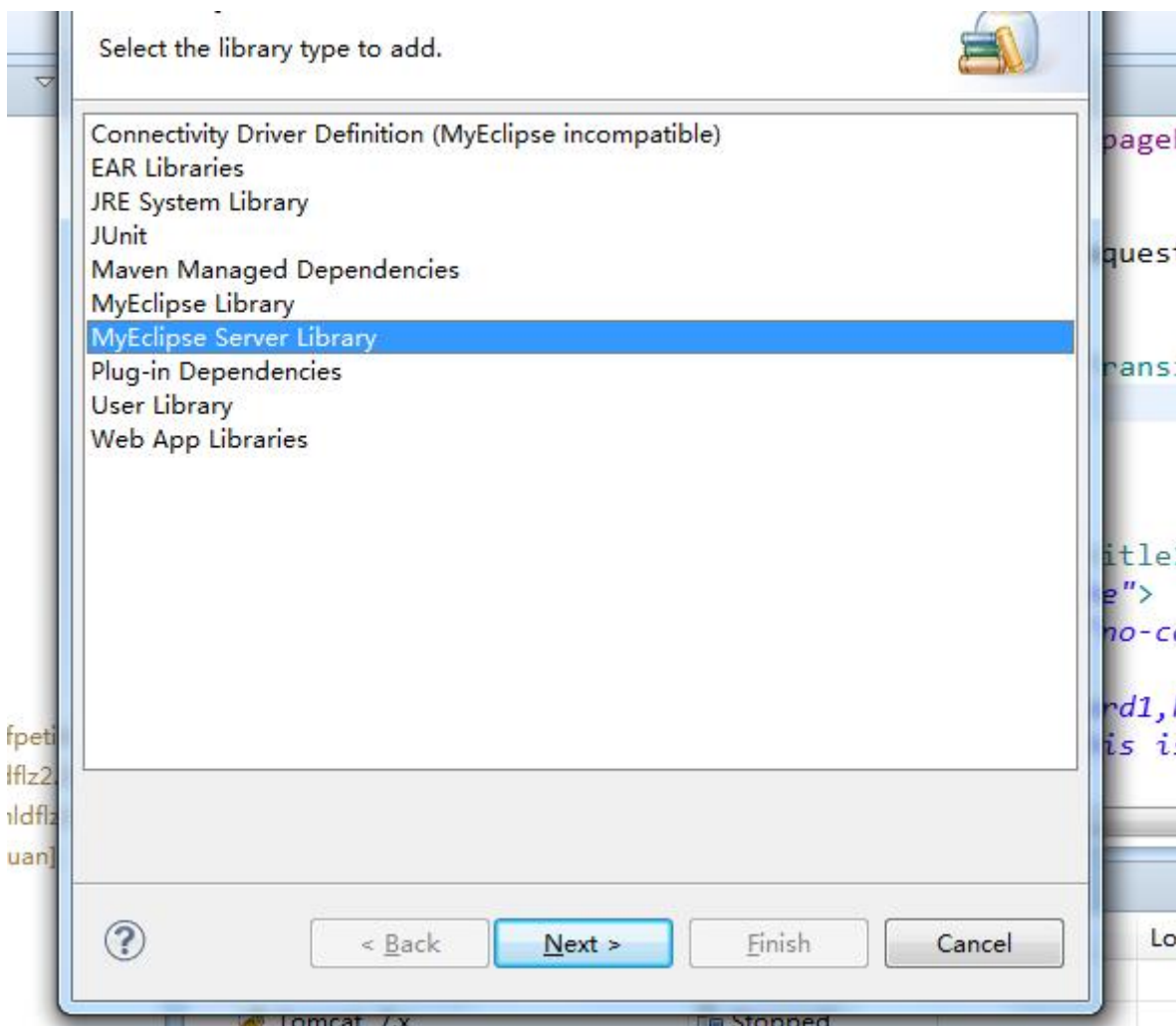


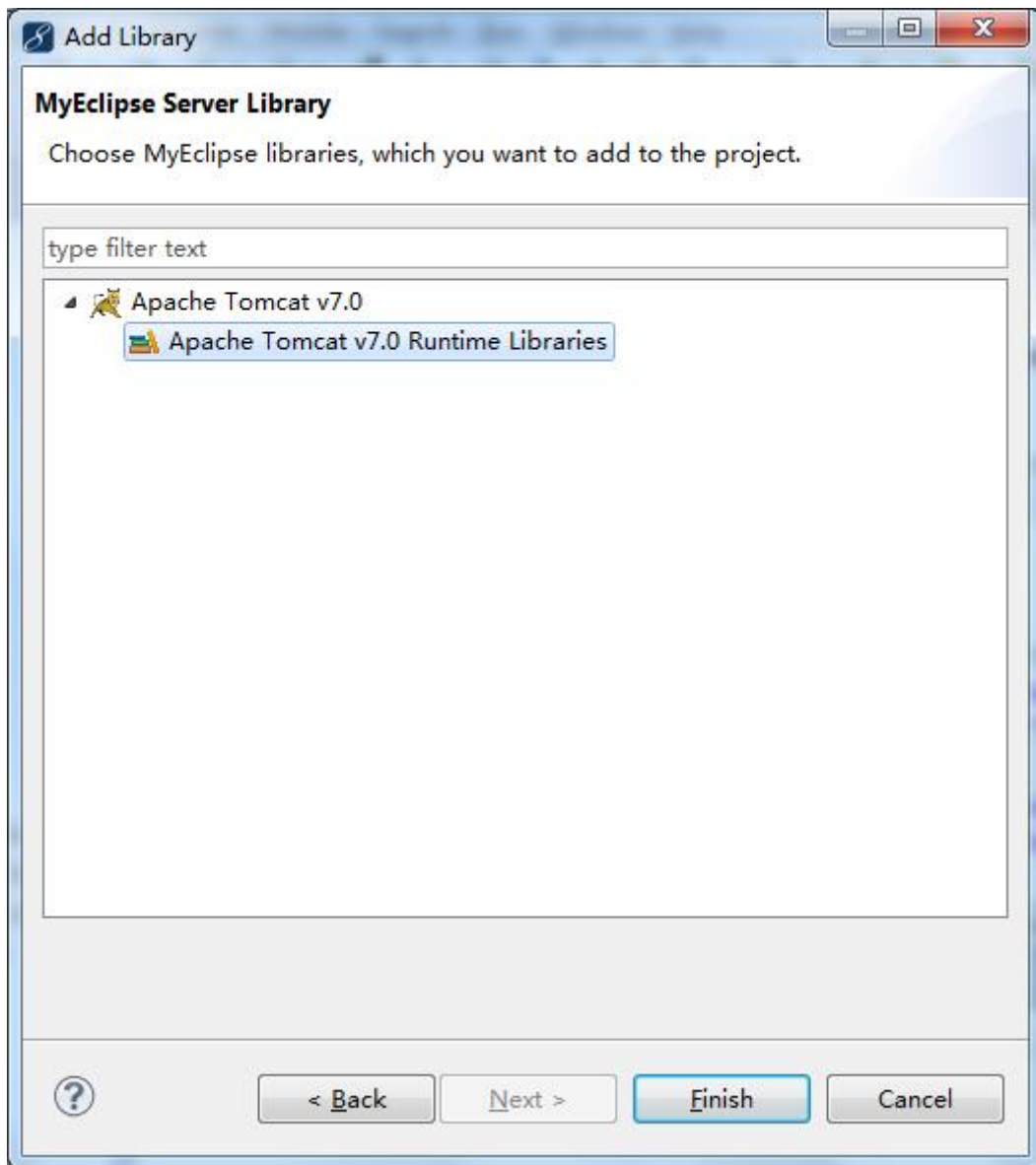
至此一个基于maven的webapp就建立好了，并可以直接从eclipse中发布到tomcat中

问题：jsp页面出现红叉



此时需要点击项目--Build Path--add Libraries





测试连接: <http://localhost:8080/ActivitiDemo/index.jsp>