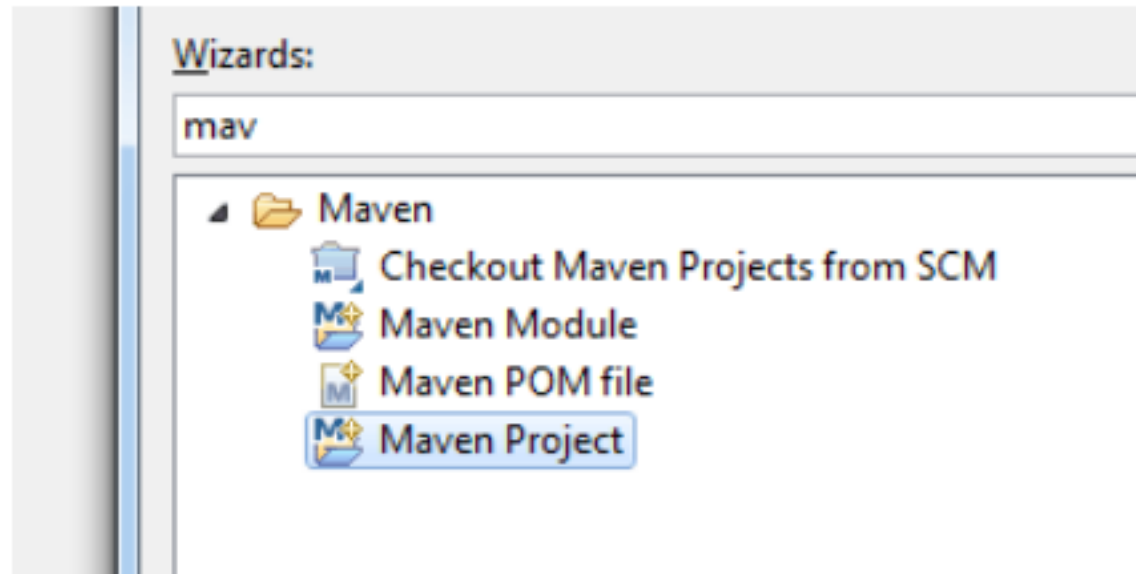
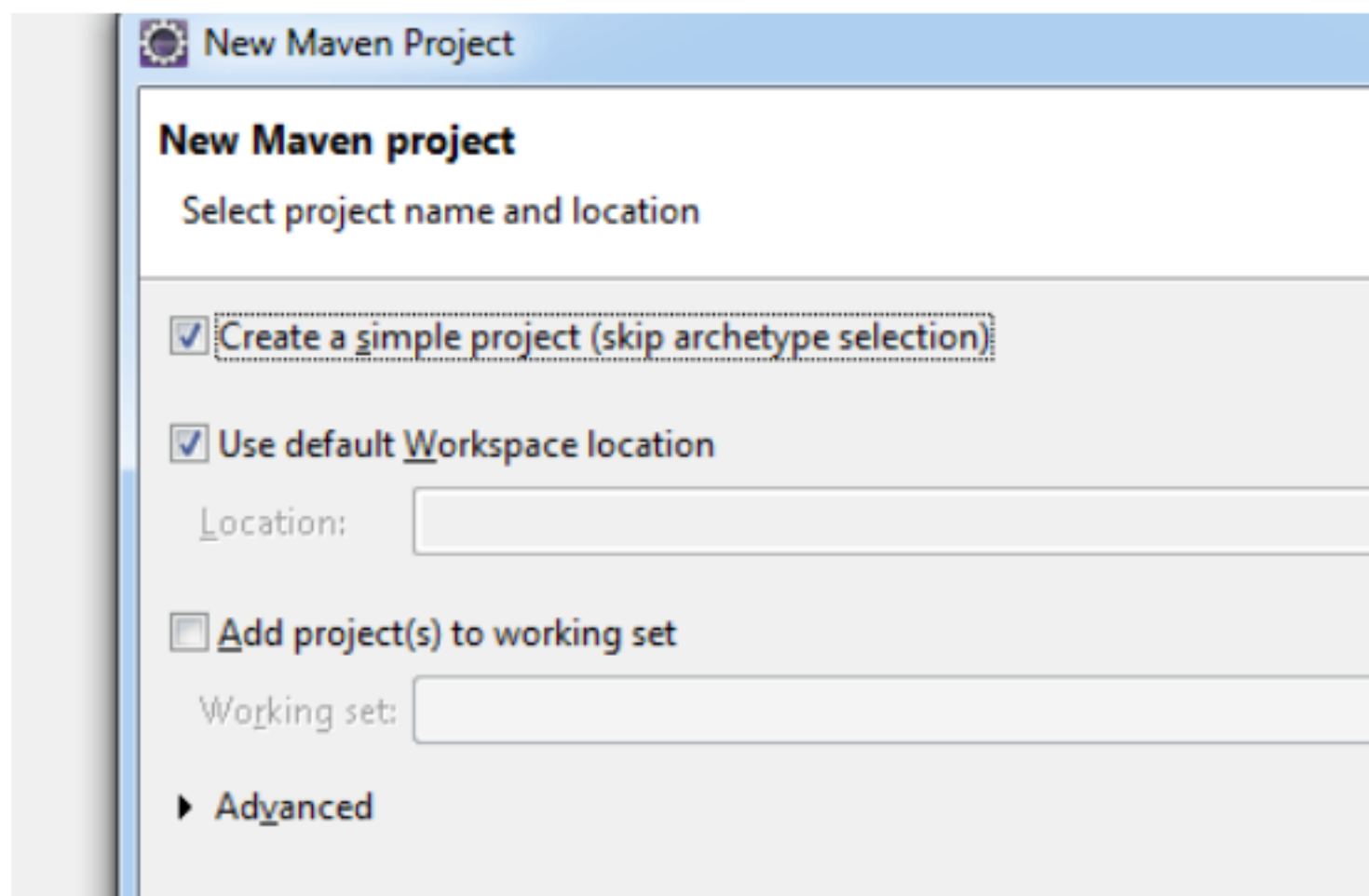


使用 eclipse 插件创建一个 web project

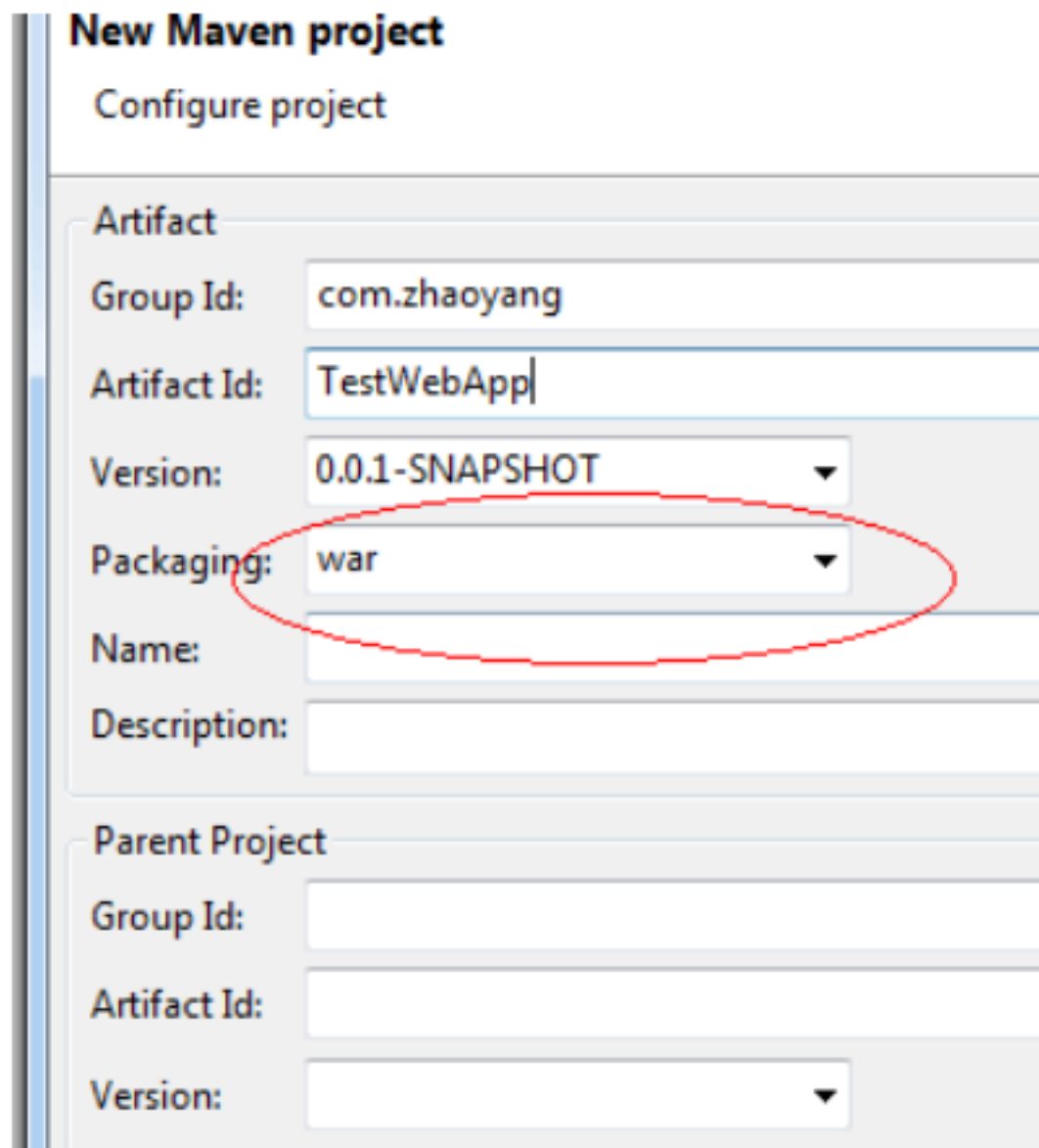
首先创建一个 Maven 的 Project 如下图



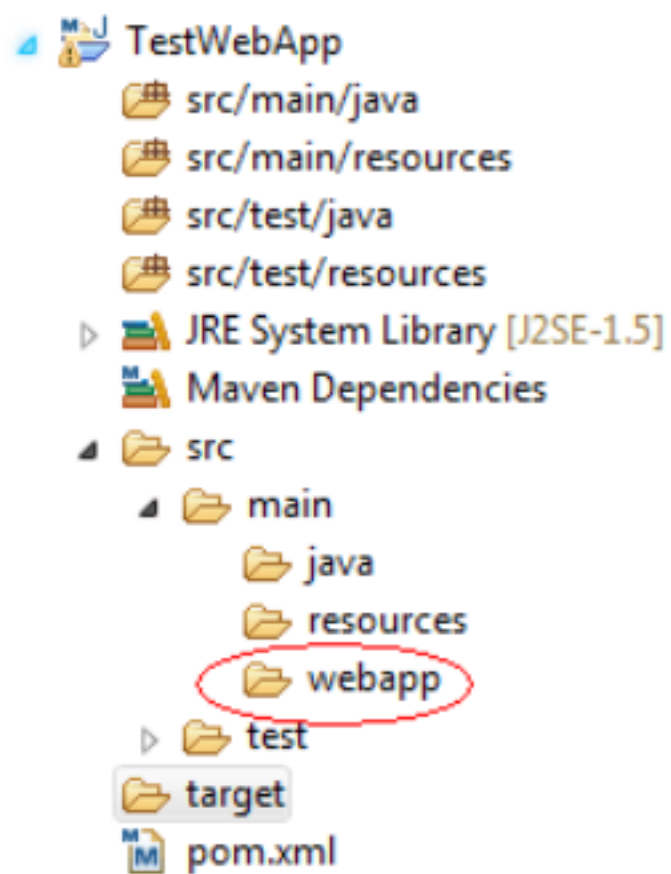
我们勾选上 Create a simple project (不使用骨架)



这里的 Packing 选择 war 的形式

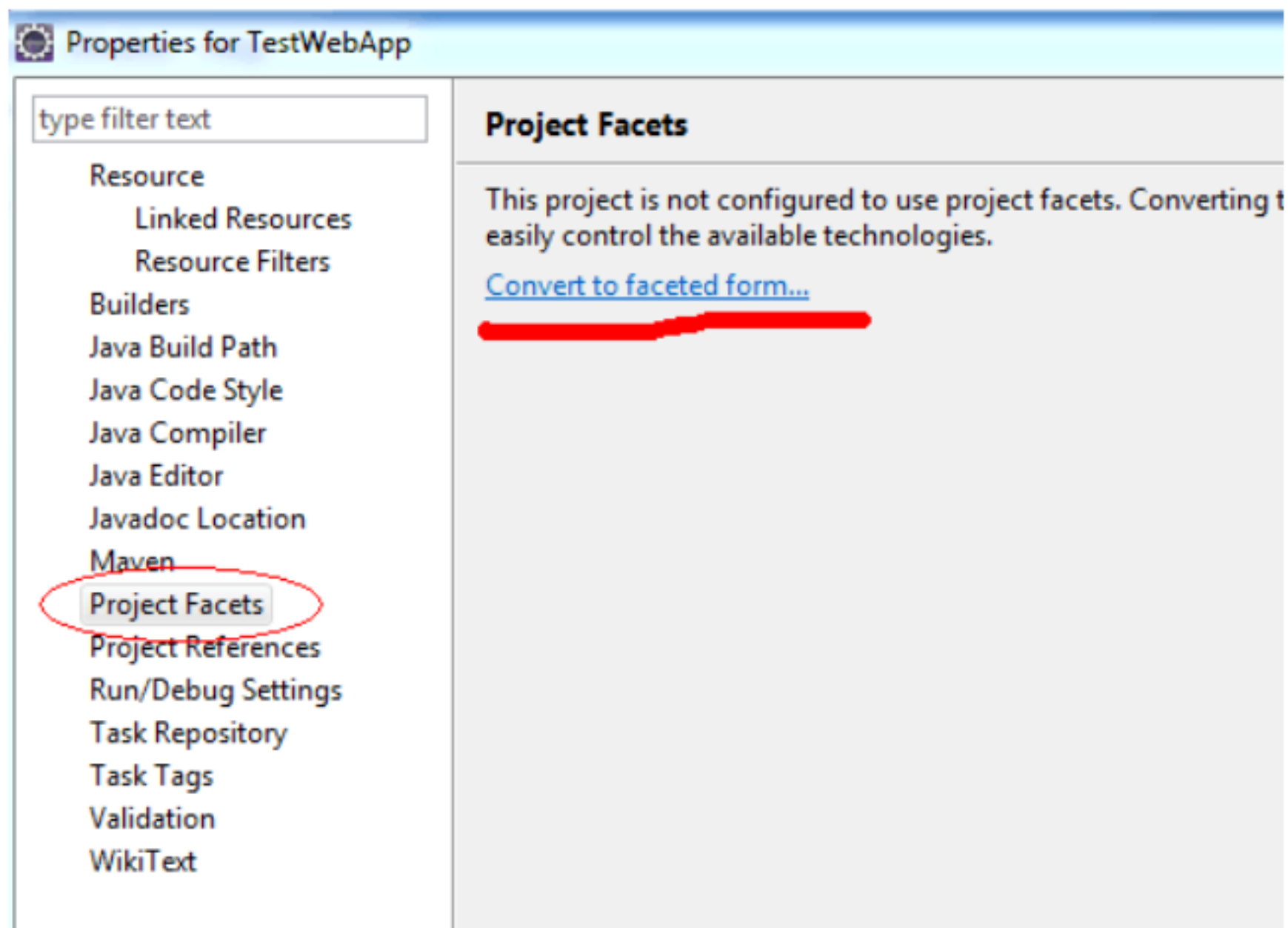


由于 packing 是 war 包，那么下面也就多出了 webapp 的目录

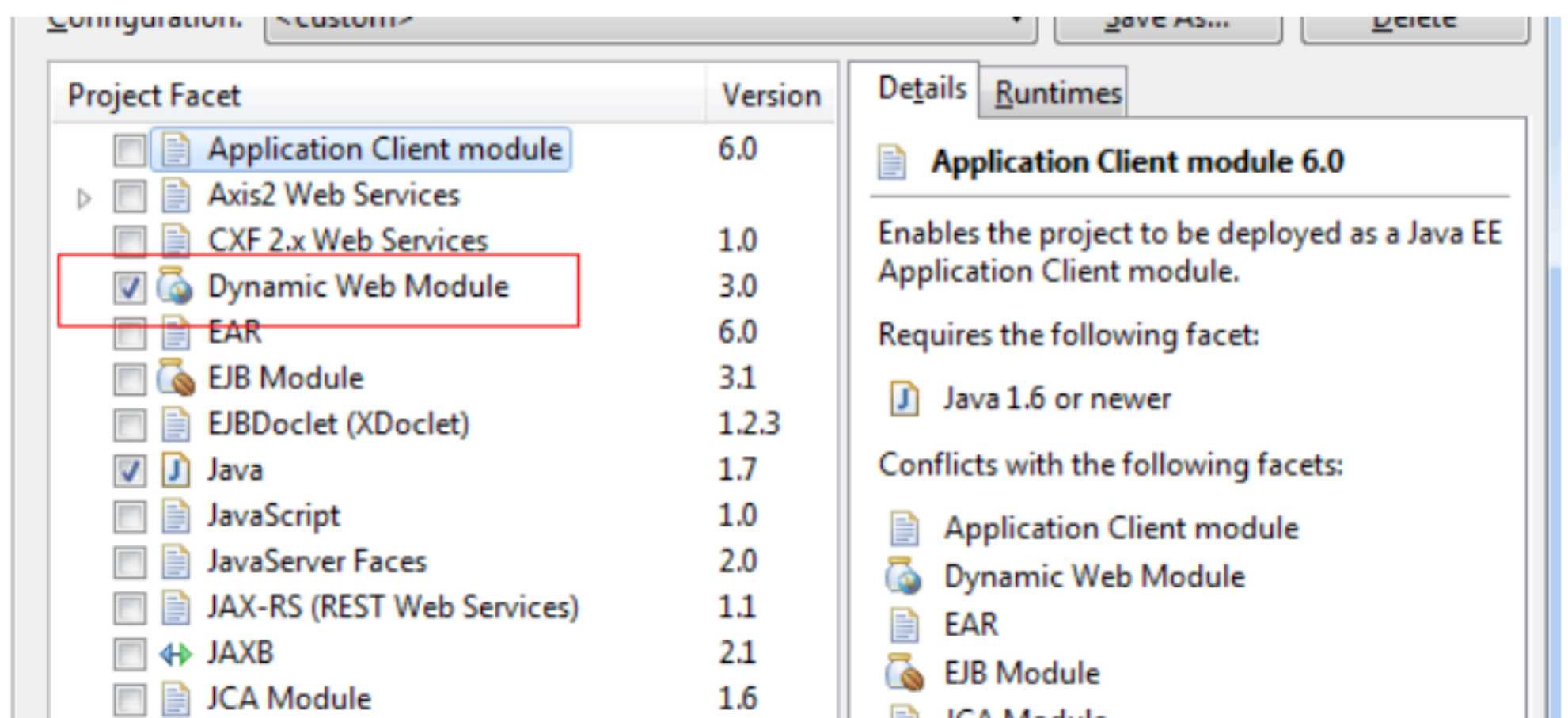


由于我们的项目要使用 eclipse 发布到 tomcat 下面，这里我们需要先把项目转换成 dynamic web project

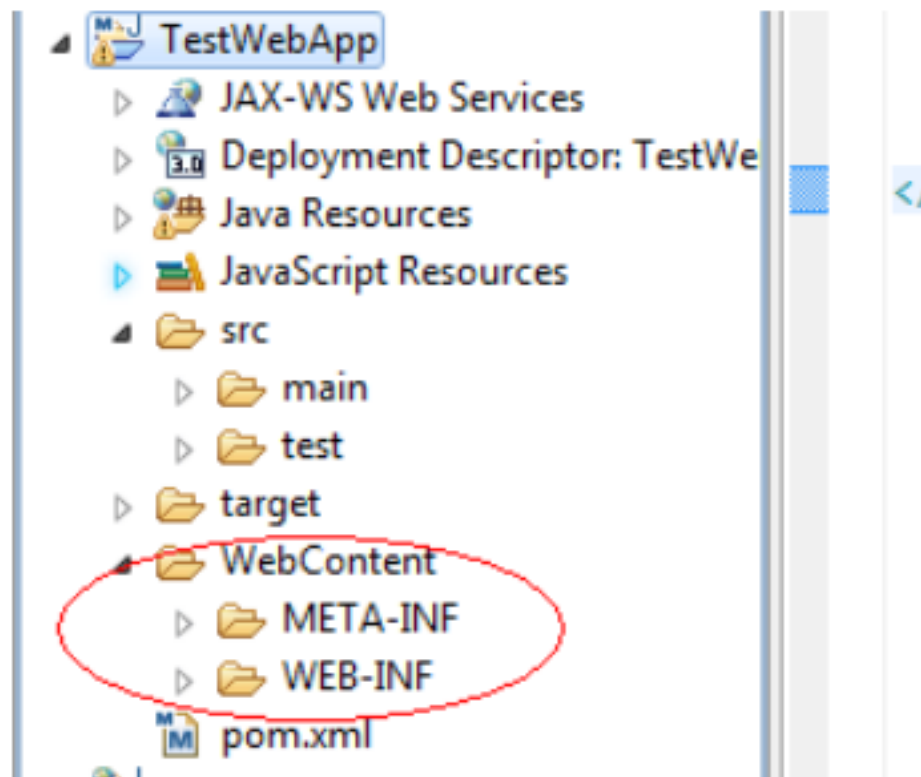
在我们的项目上点击右键，选择 properties 并找到 Project Facets，并点击 Convert to faceted form ..如下图：



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

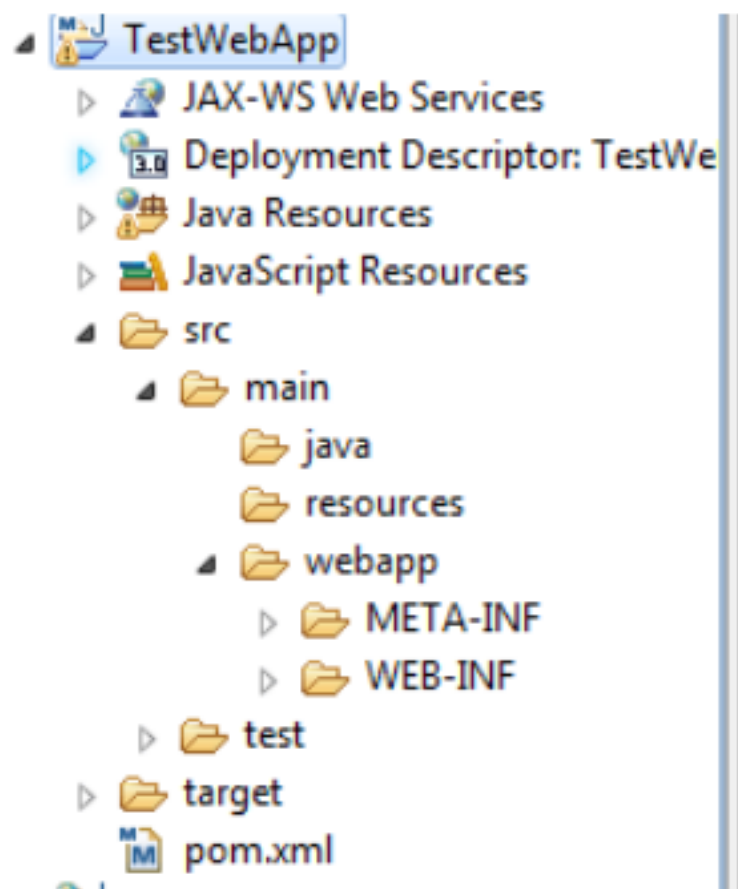


接下来观察我们的项目结构，多了一个 web content 目录

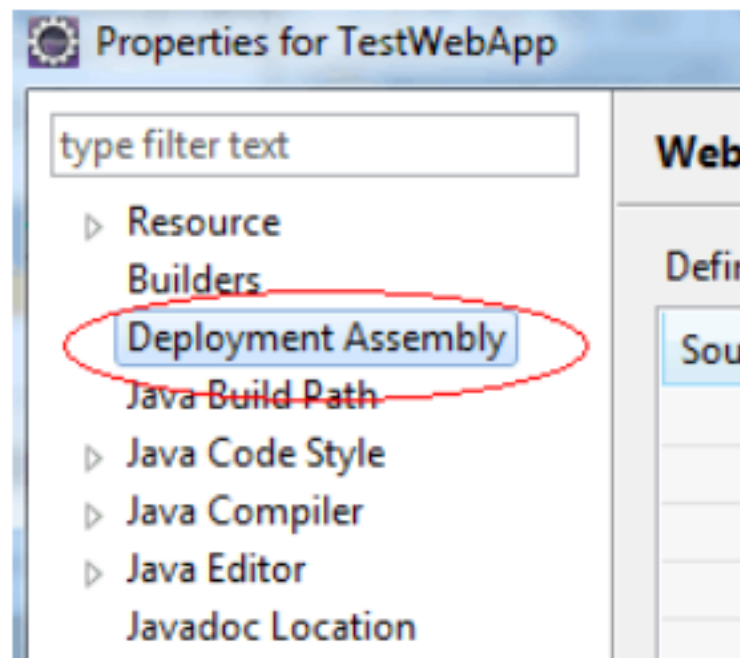


虽然此时我们可以发布到 tomcat 中，但这不符合 maven 的结构，我们还要做如下修改

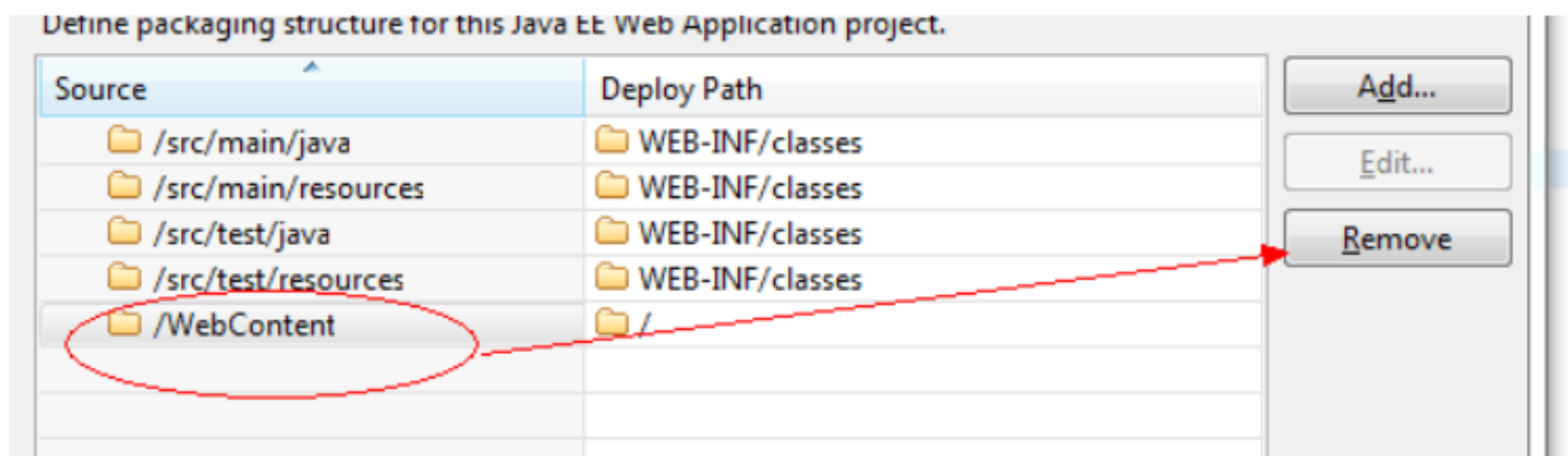
把上图 WebContent 下面两个目录 META-INF ， WEB-INF 直接剪切到 src/main/webapp 目录下，并删掉 WebContent 目录，那么现在的项目结构如下图：



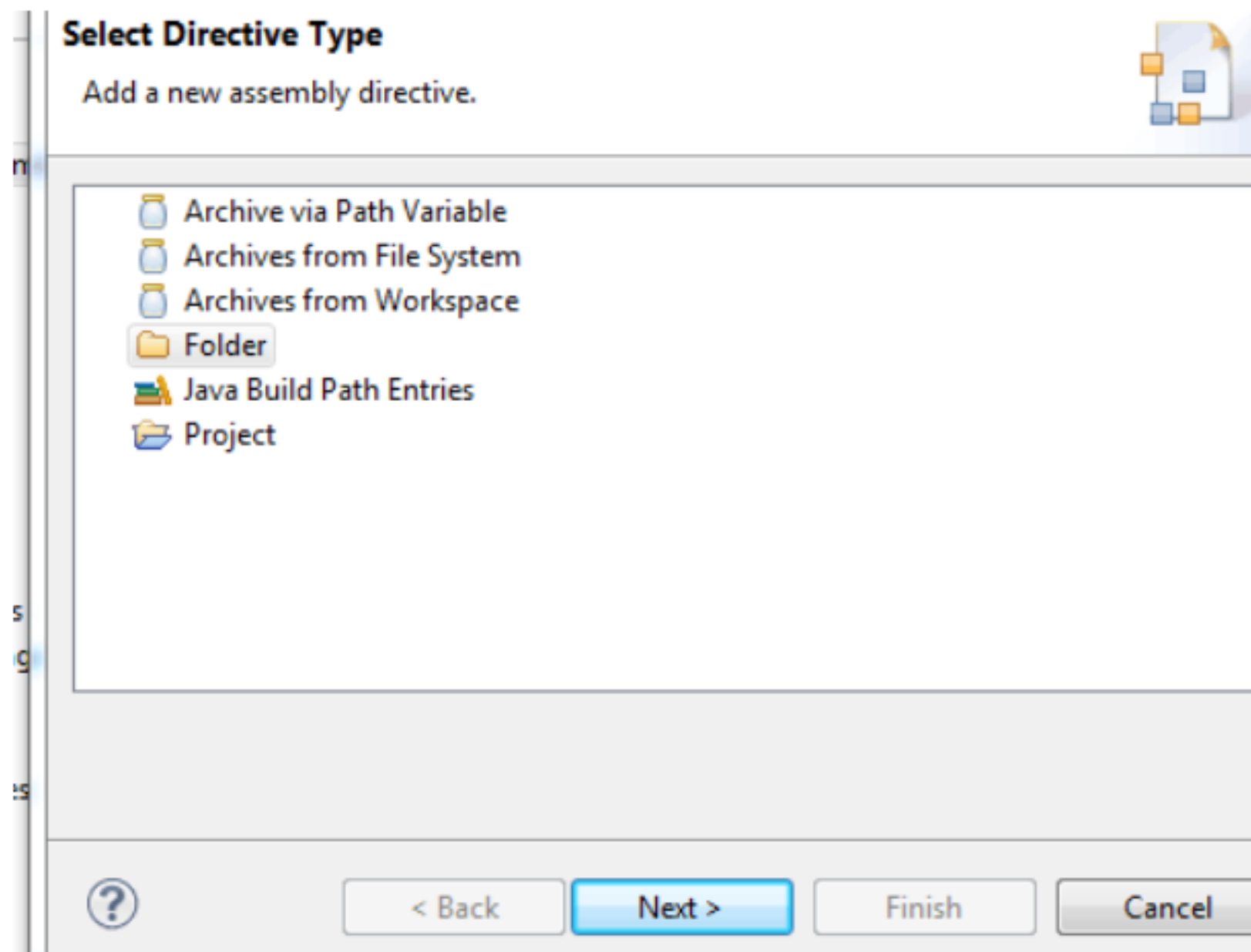
然后我们要修改发布规则，右键点击项目，选择 Deployment Assembly



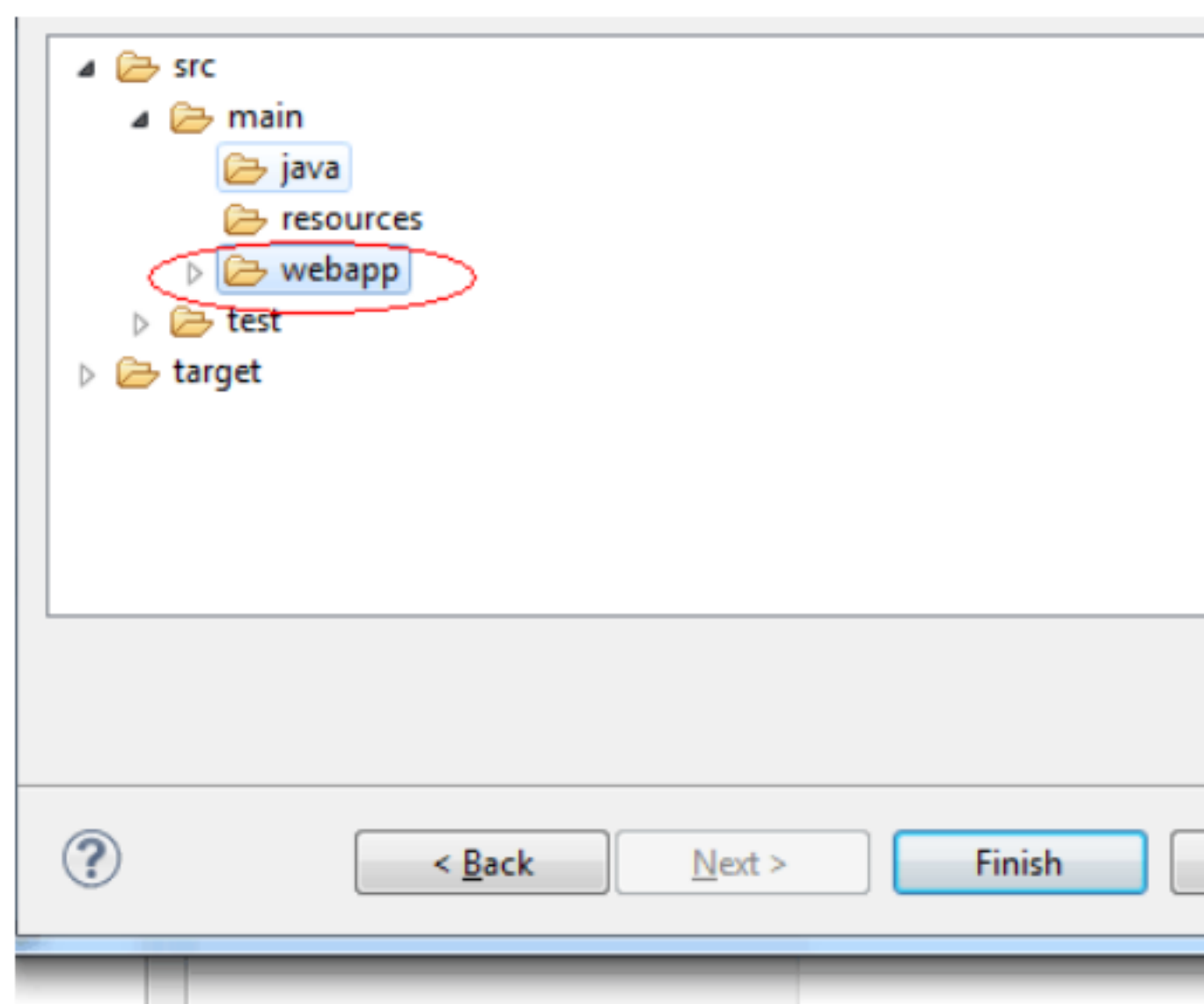
选择 WebContent ，把它 remove 掉，测试类我们也不需要发布， test 的两个目录页可以 remove



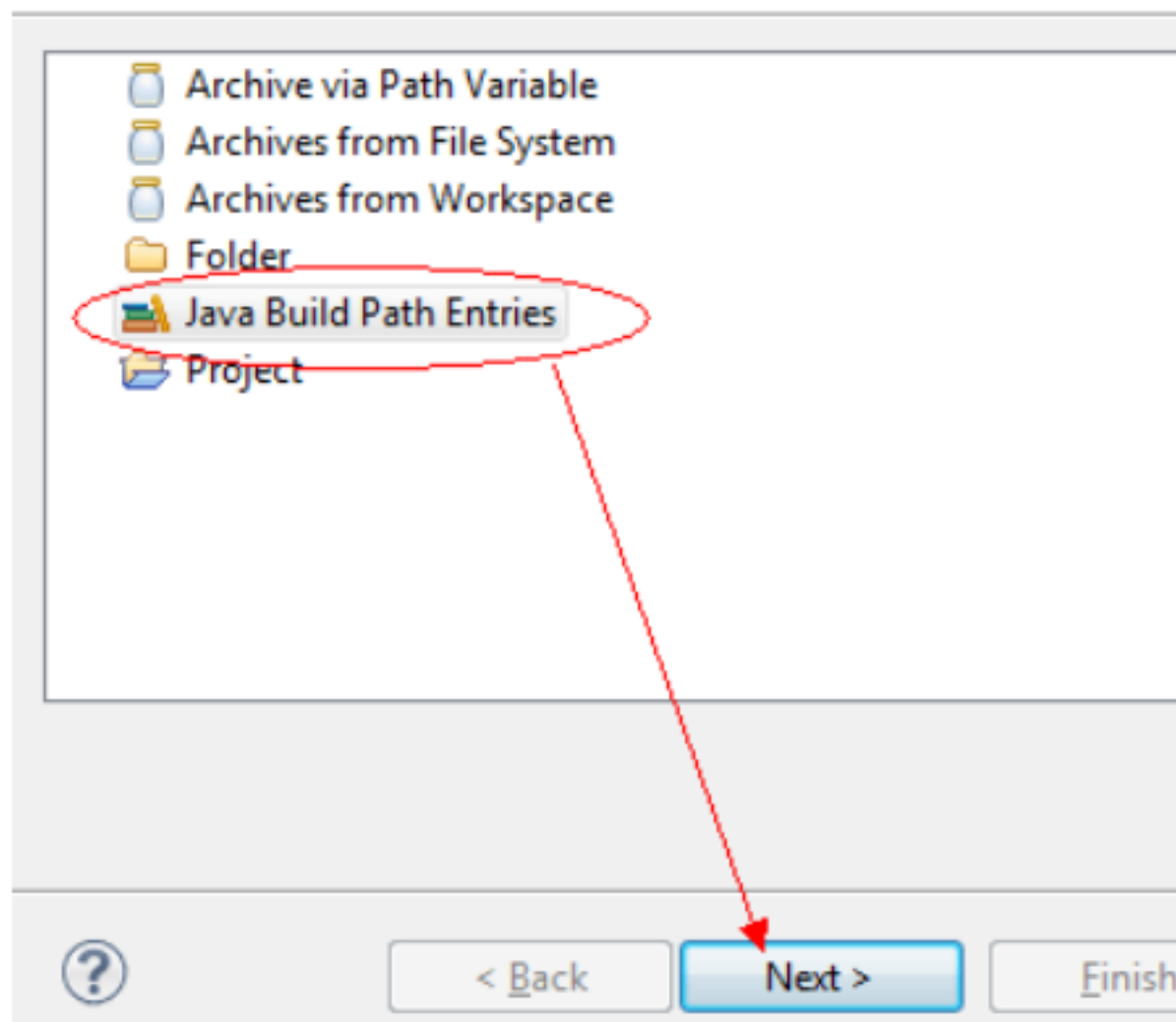
接着重新指定一个 web 的路径，点击 add ，选择 Folder -- next



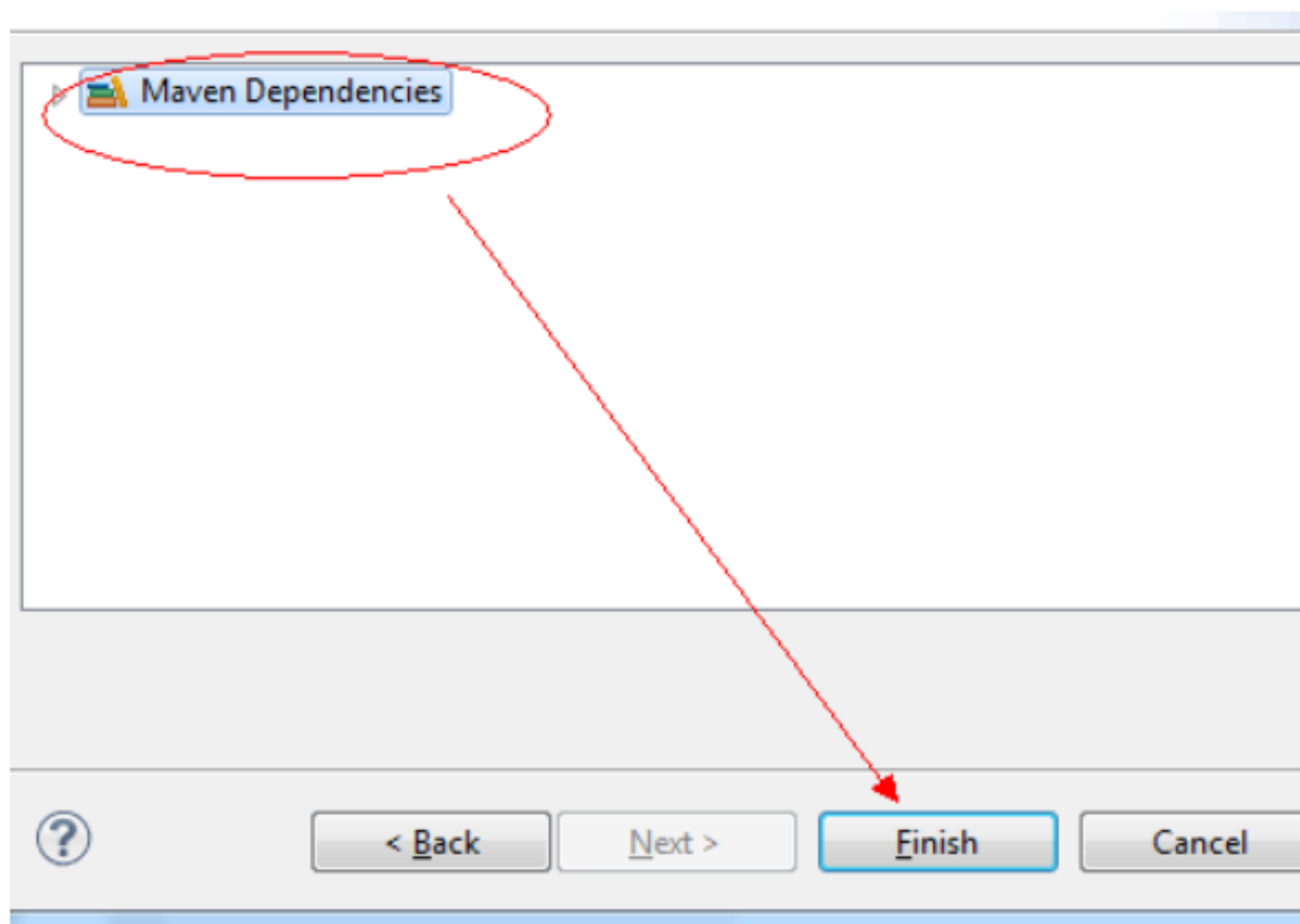
在 src 下找到 webapp 目录，然后 finish



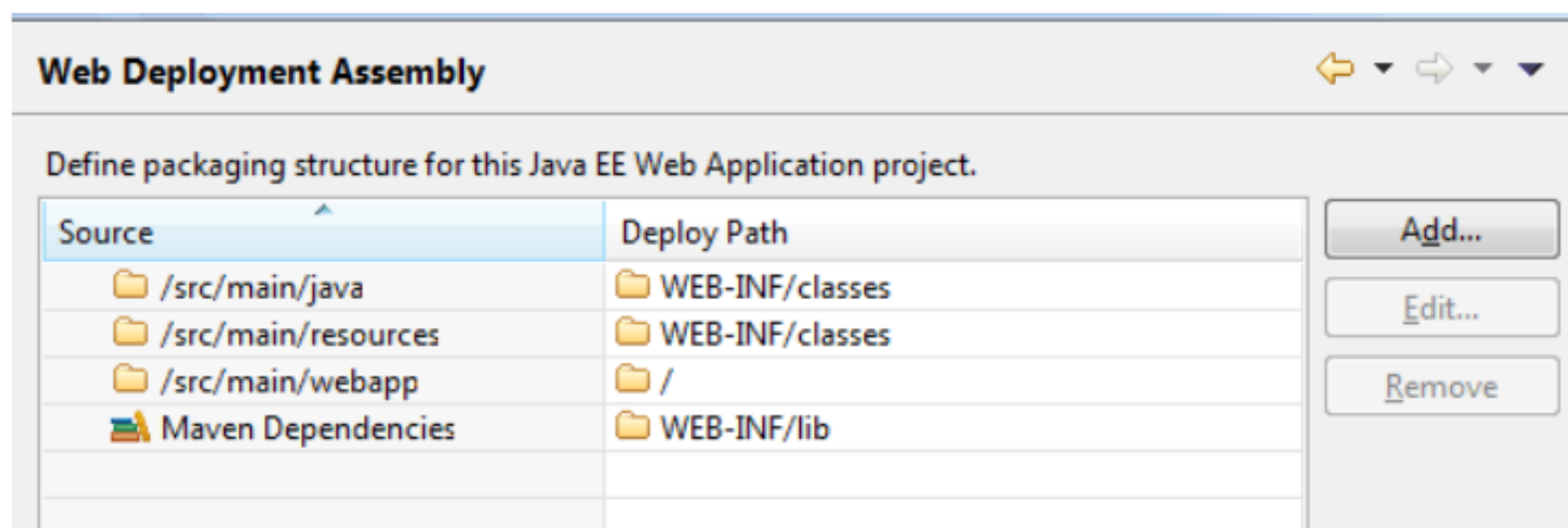
最后一步，我们要把当前的 build path 指向 Maven Dependency ，直接点击 add ，选择 Java Build Path Entries 然后 next



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

补充：我们需要在 src/main/webapp/WEB-INF 下面创建一个 web.xml

导入我们的 Spring mvc 依赖 jar 包

```
<dependencies>
```

```
<dependency>
```

```
<groupId>org.springframework</groupId>
```

```
<artifactId>spring-core</artifactId>
```

```
<version>3.0.7.RELEASE</version>
```

```
</dependency>
```

```
<dependency>
```

```
<groupId>org.springframework</groupId>
```

```
<artifactId>spring-web</artifactId>
```

```
<version>3.0.7.RELEASE</version>
```

```
</dependency>
```

```
<dependency>
```



```
<groupId>org.springframework</groupId>
<artifactId>spring-webmvc</artifactId>
<version>3.0.7.RELEASE</version>
</dependency>

<dependency>
  <groupId>org.codehaus.jackson</groupId>
  <artifactId>jackson-mapper-asl</artifactId>
  <version>1.7.1</version>
</dependency>

</dependencies>
```

直接保存，maven 就会自动为我们下载所需 jar 文件