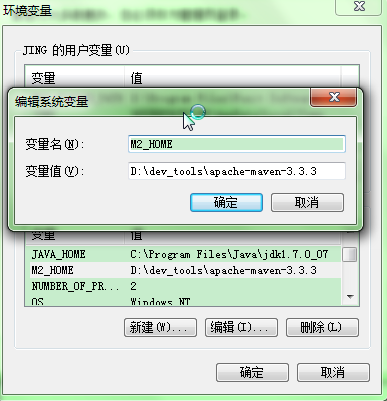
Eclipse配置maven环境笔记

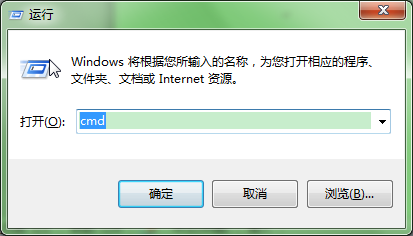
1. 下载maven <http://maven.apache.org/download.cgi> ，选择的是最新的apache-maven-3.3.3-bin.zip ;
2. 解压后放在指定目录，如D:\\dep\_tools\ ;
3. 配置maven的环境变量
4. 开始 -> 计算机 ->右击，属性 –> 高级系统设置 –> 环境变量 ->在系统变量那一栏中，新建填上变量名(M2\_HOME)和变量值(D:\dev\_tools\apache-maven-3.3.3),确定，确定，确定。

如图：

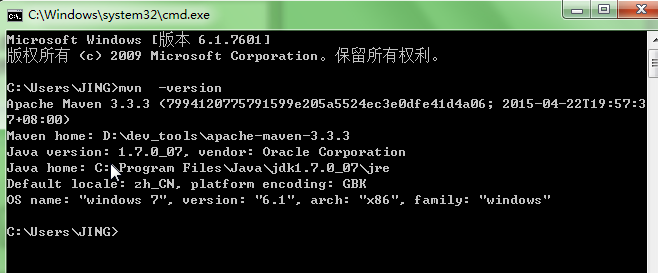


1. 验证mvn是否安装成功

快捷键win+R –> 输入cmd –> 确定



1. 在弹出的输入框中输入mvn -version；出现如图所示的内容maven就安装成功了



1. 配置maven

修改默认的依赖包仓库位置

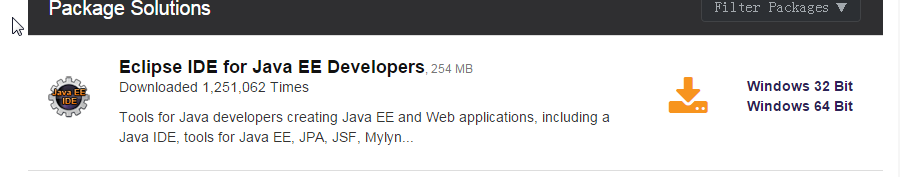
Maven 目录(D:\dev\_tools\apache-maven-3.3.3) -> conf -> settings.xml

Maven默认的本地仓库位置是${user.home}/.m2/repository，使用过程中这个目录里的文件会比较多，占用空间越来越大。一般建议更换到其它磁盘目录下。如下配置，就把默认的本地仓库更改到D:/dev\_tools/DepLibs.m2/repository这个目录(这个目录结构需要自己创建好)。

并把settings.xml 放在D:/dev\_tools/DepLibs/.m2/目录下。

1. 下载Eclipse javaEE版本 <http://www.eclipse.org/downloads/>

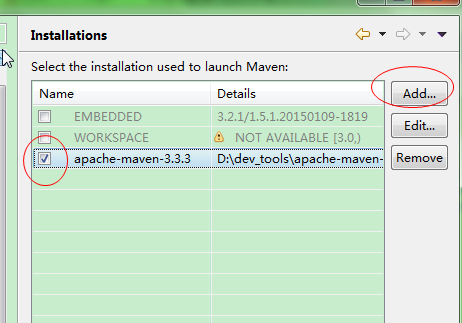
选择32位或64位版本



该版本是2015年4月最新版本，已经自带了m2Eclipse插件

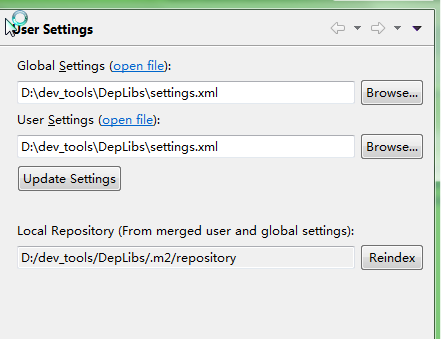
1. 启动Eclipse，配置m2Eclipse插件关联maven程序

Eclipse菜单栏 –> window -> preferences -> maven -> installations -> 右侧的add -> directory –> 选择maven的安装主目录(D:\dev\_tools\apache-maven-3.3.3) –> finish -> 在关联的maven前打钩 -> apply；



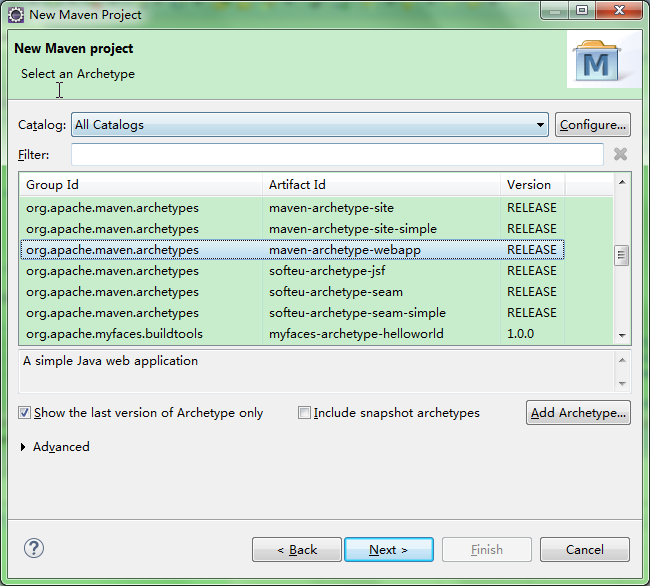
继续下一步，设置自定义的本地仓库位置

点击user settings -> 右侧，global settings 和user settings都设置为D:\dev\_tools\DepLibs\settings.xml -> apply -> OK 完成配置;

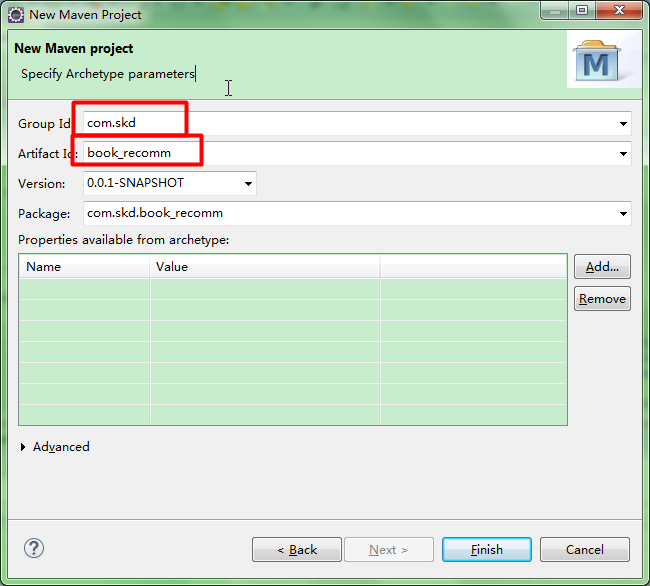


1. 创建一个maven的web项目

菜单栏 File ->New -> project ->选择maven项 -> maven project -> next ->不选create a simple project,next 🡪 GroupId选择 org.apache.maven.archtypes ,Artifact Id 选择 maven-archetype-webapp -> next



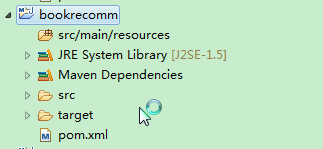
点Next > group id 处输入包名，在Artifact Id输入项目名



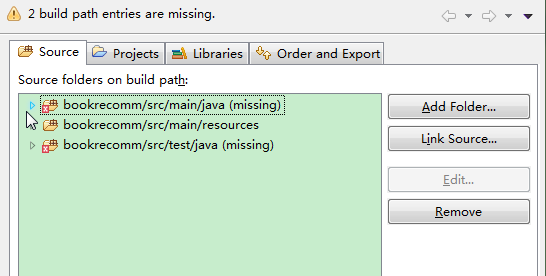
–> 点击finish。Eclipse就会自动创建一个maven项目，并且会在中央仓库中下载所需要的包。如果本地仓库位置没有包，会从网上下载所需要的包，有一段时间的等待

项目后续配置

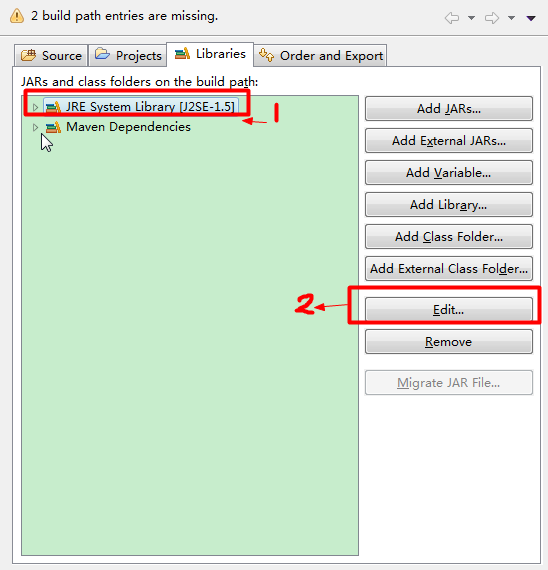
1. 项目创建完成后默认只为我们创建了 src/main/resources，而没有src/main/java，src/test/java和src/test/resources。



打开项目属性 -> java build path -> source 看到 src/main/java和src/test/java 显示 missing：



解决办法：打开上图的Libraries，修改为自己的JDK（1.7 以上版本）



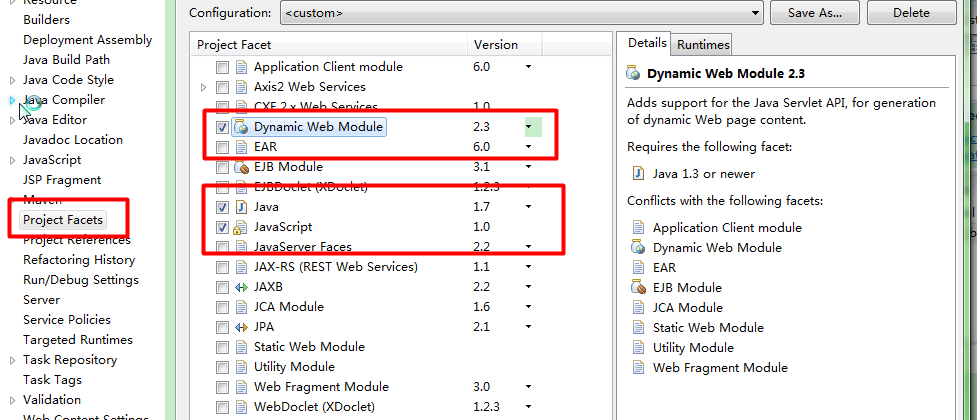


自行添加src/test/resources

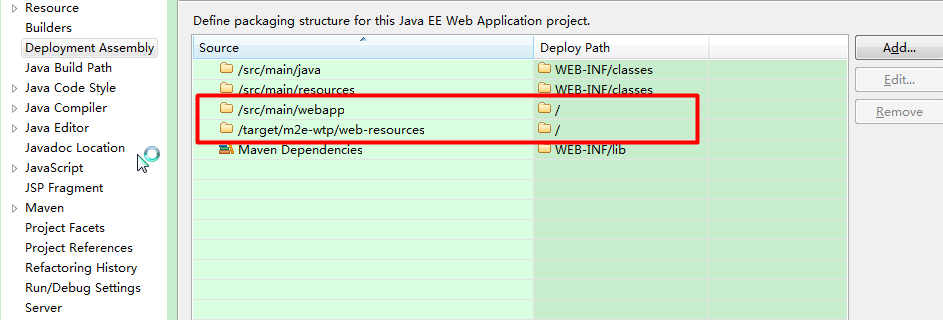
右击项目名 –> new -> Source folder -> 在Folder Name 填写：src/test/resources

1. 设置 Project Facets，选择 Java 1.7 版本和 Dynamic Web Module 2.3 版本

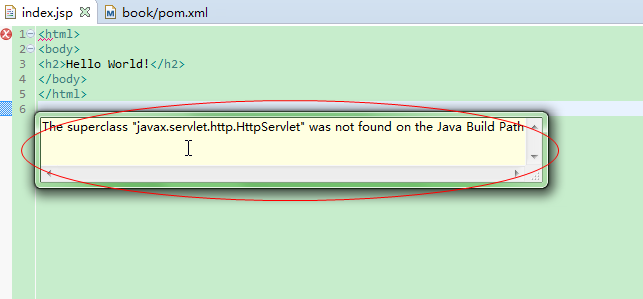
右击项目 -> Properties –> Project Facets



1. 设置部署打包结构如下，删除测试相关目录,点击add添加红框中的两项



1. 还需要在pom.xml添加依赖包解决图中问题，添加 Servlet-api 的依赖包



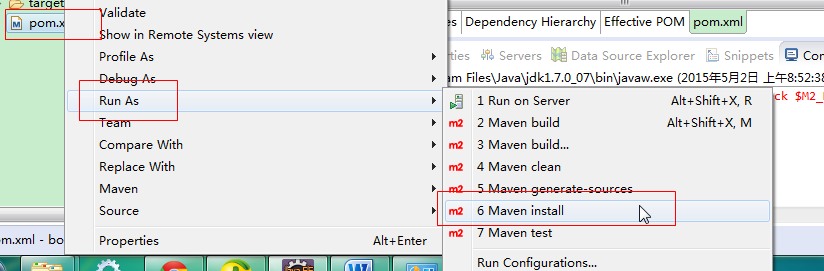
添加好的包如下：



保存后项目会自动去寻找并下载不存在的jar包。

1. 构建和部署项目

右击整个项目的pom.xml –> Run As -> Maven install



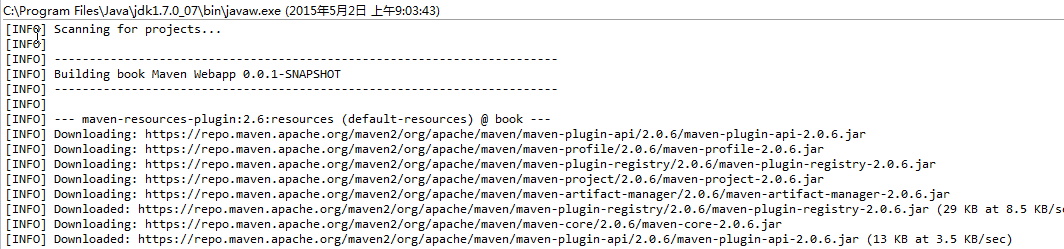
但出现了这样的错误

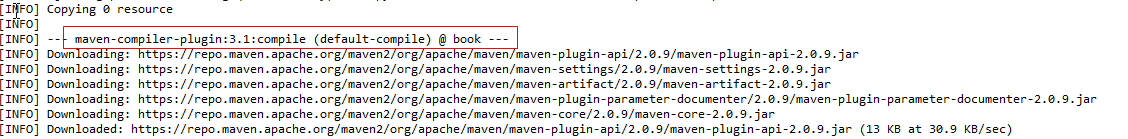


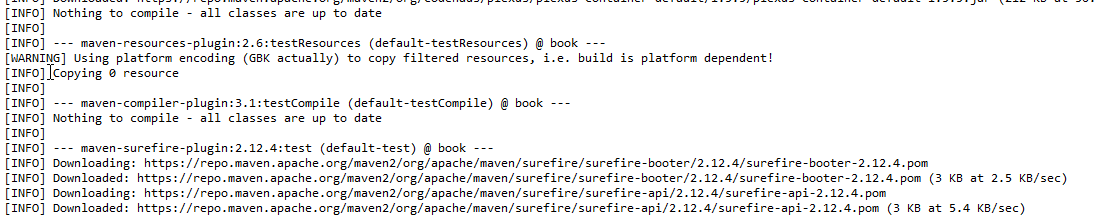
解决办法：菜单栏window –> preperences -> java ->installed JREs -> 右侧选择使用的JRE环境，点击Edit，将Default VM arguments处填上 -Dmaven.multiModuleProjectDirectory=$M2\_HOME –> finish。

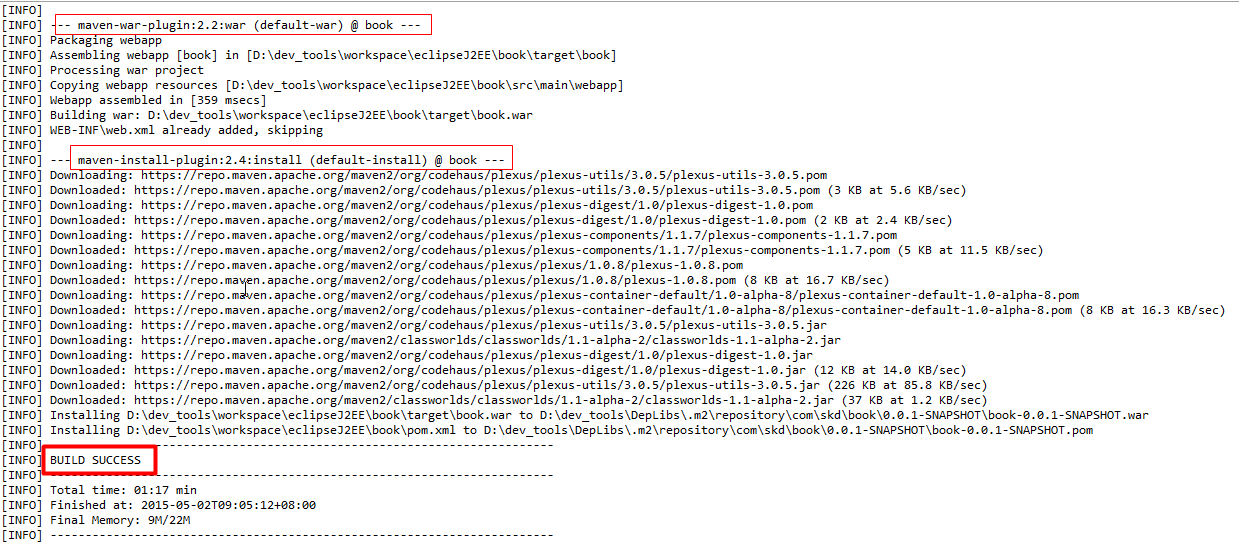
控制台输出结果：

会下载一些依赖包



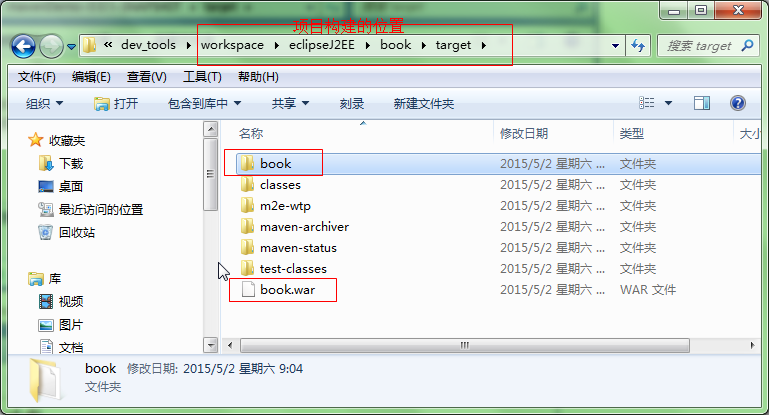






这样就构建完成了

1. 查看已经生成的项目部署文件



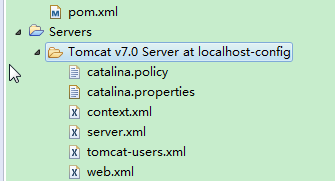
1. 因为项目还未链接tomcat，就暂时将war包放在tomcat下，启动tomcat并访问该war包



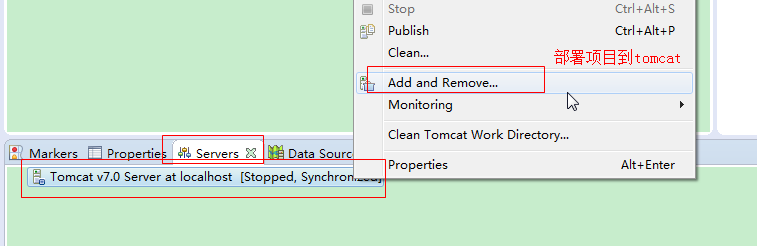
1. 配置Tomcat，让项目自动部署到tomcat中运行

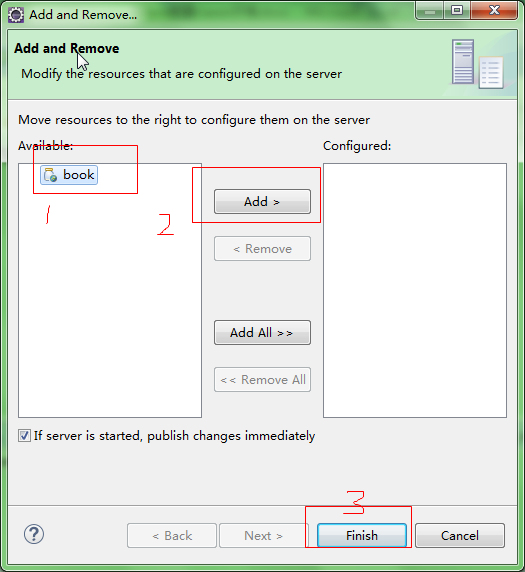
菜单栏window -> preperences ->Server Runtime Environment -> 右侧，点击Add –>选择tomcat 对应版本(apache tomcat v7.0) -> next ->点击browser ，选择tomcat 的安装位置 ，Name处随便填，JRE选择你刚才配置Default VM arguments的那个 –> finish;

左侧会出现servers目录

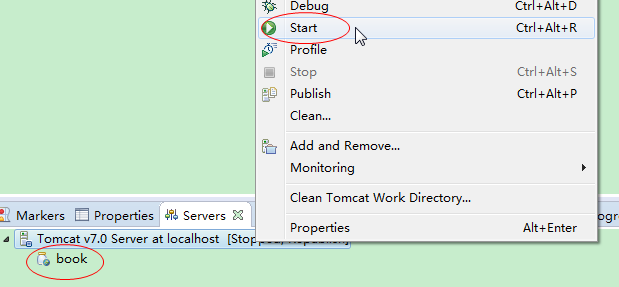


1. 看中间下部是否有Servers标签，没有的话点菜单栏window -> show view -> servers





1. 现在项目已经部署到tomcat中，点击start启动tomcat



1. 在浏览器中访问

