

JEECG 微云开发平台

Maven 版本手把手入门手册

2013/11/17

scott@jeecg.org

张代浩

官方标准开发工具：

1.IDE

Eclipse Java EE IDE for Web Developers.

Version: Helios Release

Build id: 20100617-1415

2.JDK1.7

3.Tomcat6

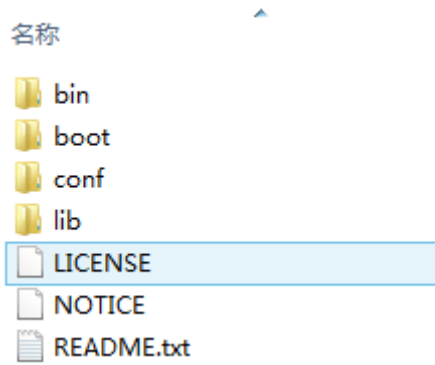
4.Maven 项目构建

5.JEECG 采用 oschina 的仓库存储依赖的 jar

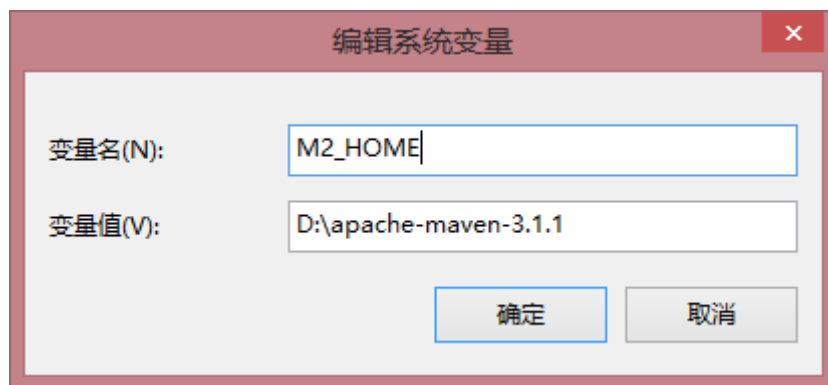
一. Maven 环境搭建

1. 下载：到 Maven 官方网站 <http://maven.apache.org/download.cgi> 下载目前的最新版本 [apache-maven-3.1.1-bin.zip](#)
2. 解压：下载到本地的 zip 文件，只需解压成 X:\apache-maven-3.1.1 即完成安装，目录结构

为如下



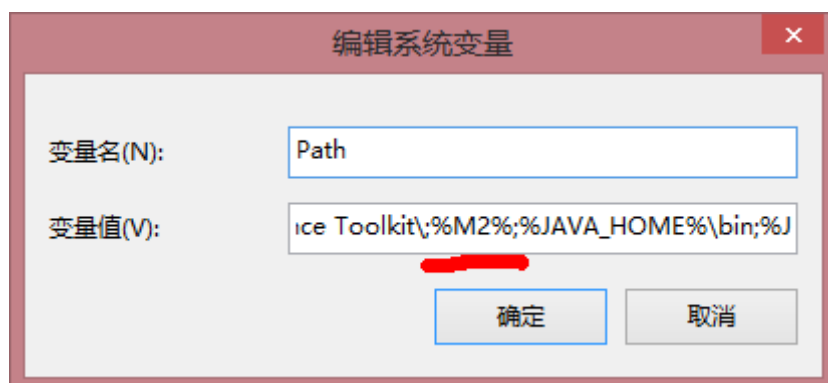
3. 配置环境变量：这台电脑->鼠标右键->属性->高级环境设置->环境变量-系统变量框下新增变量名 M2_HOME ，值为 X:\apache-maven-3.1.1



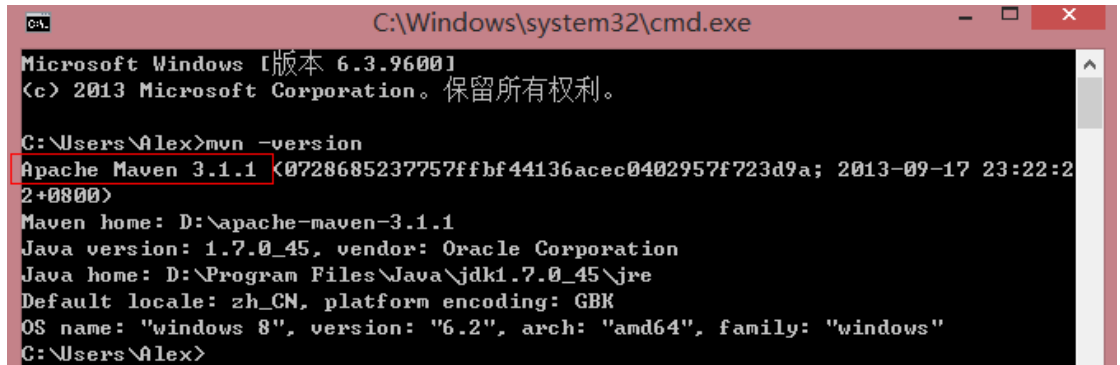
再新建变量 M2 ，值为 %M2_HOME%\bin



编辑变量名 Path ，在变量值中后面追加“;%M2%”



4. 测试安装：win+R ->cmd ->在命令行中输入 mvn -version 显示 mvn 版本信息为 Apache Maven 3.1.1 则为安装成功。



```

C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.3.9600]
(c) 2013 Microsoft Corporation。保留所有权利。

C:\Users\Alex>mvn -version
Apache Maven 3.1.1 (0728685237757ffbf44136acec0402957f723d9a; 2013-09-17 23:22:22+0800)
Maven home: D:\apache-maven-3.1.1
Java version: 1.7.0_45, vendor: Oracle Corporation
Java home: D:\Program Files\Java\jdk1.7.0_45\jre
Default locale: zh_CN, platform encoding: GBK
OS name: "windows 8", version: "6.2", arch: "amd64", family: "windows"
C:\Users\Alex>

```

5. 修改 maven 仓库存放位置：找到 maven 下的 conf 下的 settings.xml 配置文件，我是在 D:\apache-maven-3.1.1\conf\settings.xml。maven 的仓库默认是放在本地用户的临时文件夹下面的 .m2 文件夹下的 repository 下，这很影响 C 盘的空间。现在我们来修改将它指定到我们自己的路径下，我现在要将仓库指定到 D:\Maven\repo\m2（可以自己定义位置）目录下。

```

|-->
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://maven.apache
<!-- localRepository
| The path to the local repository maven will use to store artifacts.
|
| Default: ${user.home}/.m2/repository
<localRepository>/path/to/local/repo</localRepository>
-->
<localRepository>D:/Maven/repo/m2</localRepository>
<!-- interactiveMode
| This will determine whether maven prompts you when it needs input. If set to false
| maven will use a sensible default value, perhaps based on some other setting, for

```

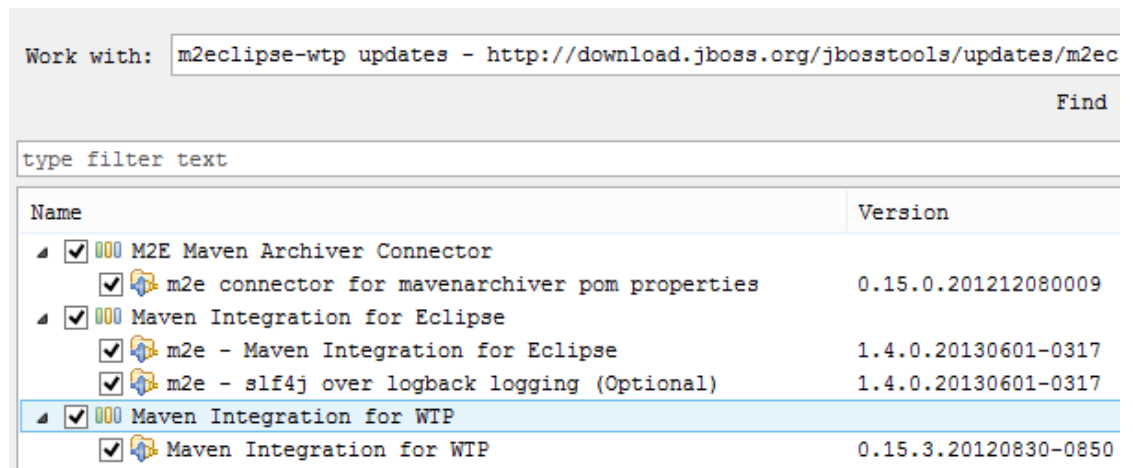
6. OK，先来体会一下 maven，在 cmd 中敲并回车执行：mvnhelp:system 这时候 maven 就会从远程仓库开始下载一大堆的东西，没事，让它下载着，迟早都要下载的。

二. MyEclipse 或 Eclipse 环境搭建

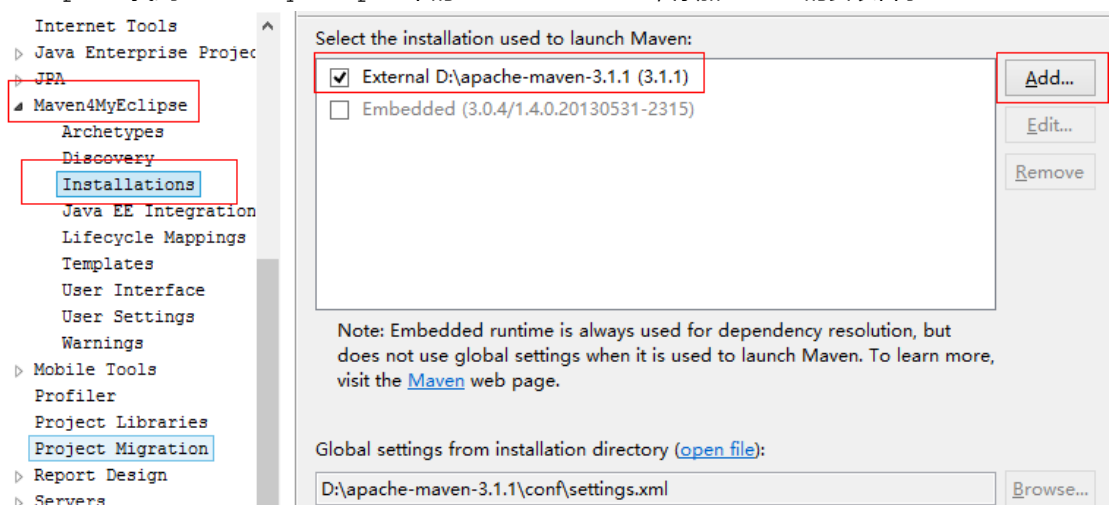
1. 插件安装：MyEclipse8 - MyEclipse2014 已经自带 maven 插件，此步可以略过。
2. 在线安装：选择 Help --> Install New Software 输入地址

<http://download.jboss.org/jbosstools/updates/m2eclipse-wtp/>，把选项

勾上，然后等待它下载安装，完成之后重启 eclipse 即可

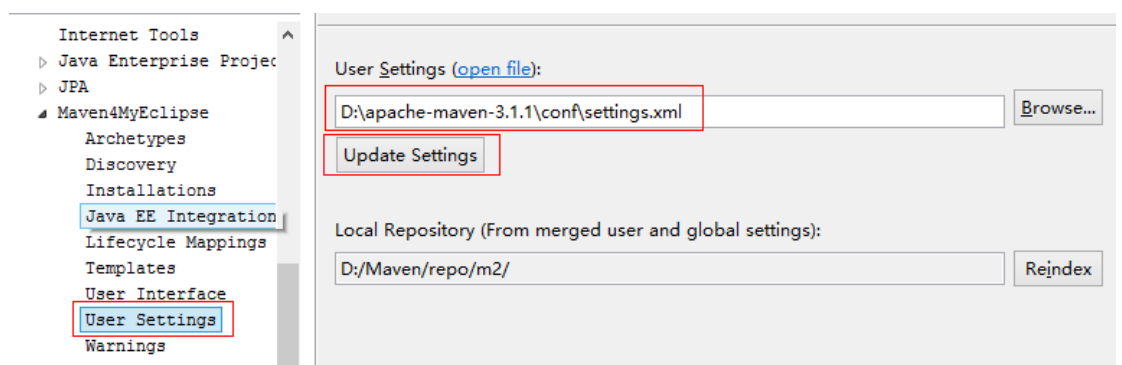


3. 配置 eclipse : 找到 Maven4MyEclipse 下的 Installations , 添加 maven 的安装目录



找到 User Settings 选择 maven 安装目录下的 settings.xml , 单击 Update

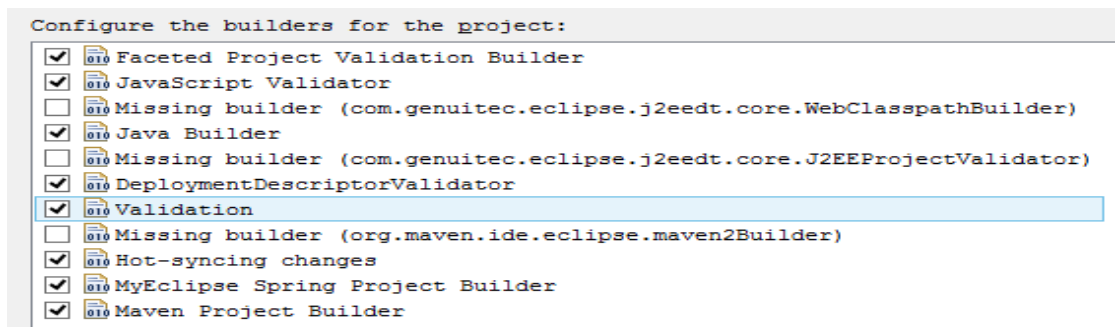
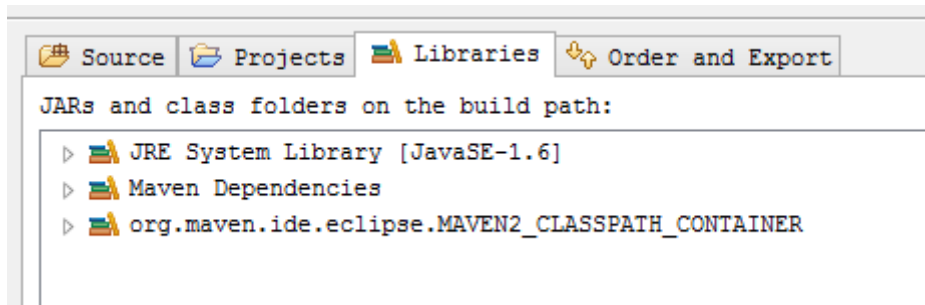
Settings.



OK , eclipse 配置已经完成了。

三. 加载 jeecg-maven 项目

1. 加载项目 : 加载 jeecg-maven 项目到 MyEclipse 后 , 选择项目名右键 properties , 选择 Java Build Path , 把红叉的 JRE 删除 , 添加自己的 JRE 库。选择 builders 把一些红叉的和带有 miss 的选项取消勾选 , 确定后退出

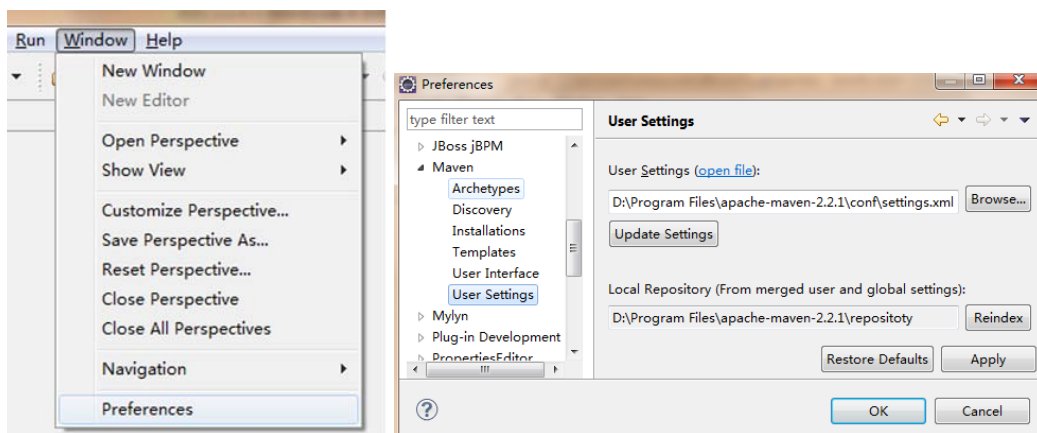


2. 下载 maven 所包含的包：选择项目名右键选择 RunAS，再选择 Maveninstall，控制台就开始下载项目所需要的包。

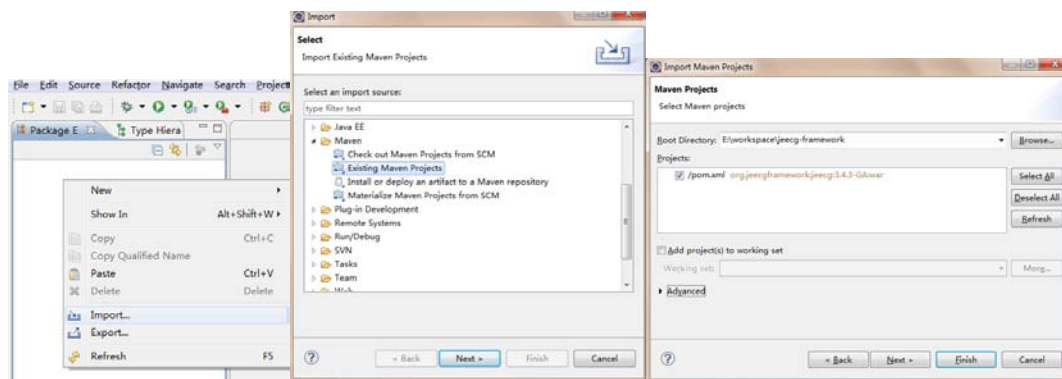
四. 配置 jeecg 的 maven 项目

1. 准备

导入 jeecg 项目之前，打开 eclipse/myeclipse，配置本地的 maven，这样当你导入 maven 项目后，会自动下载 maven 相关的依赖 jar。

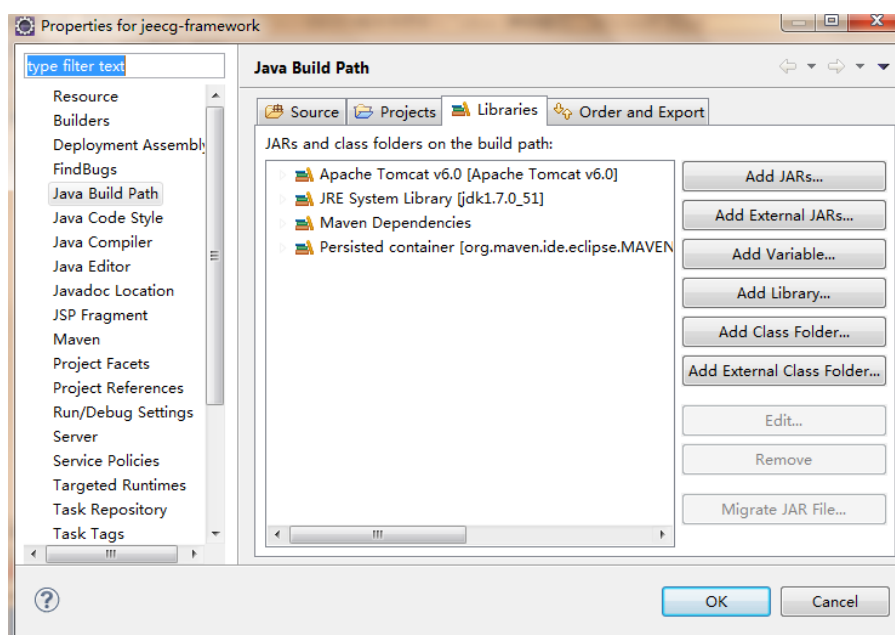


2. 导入 jeecg 的 maven 项目

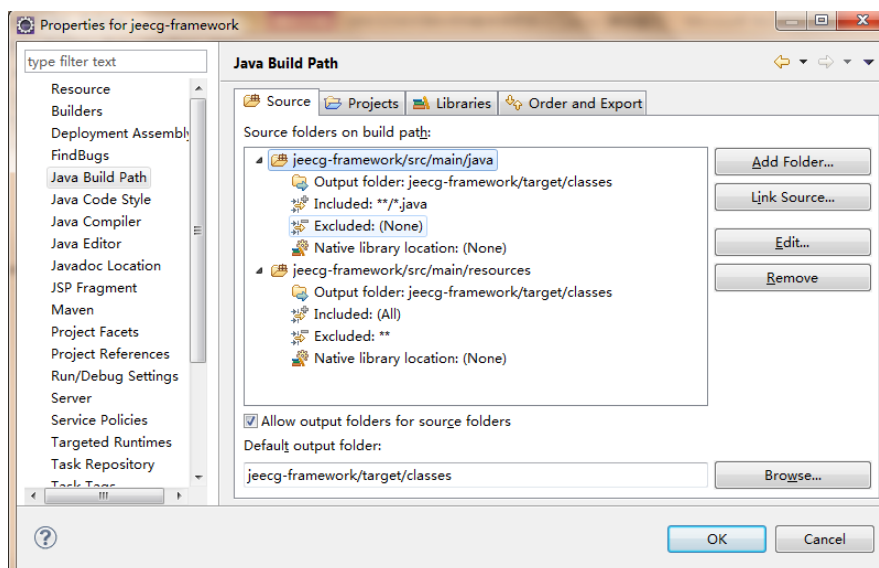


3. Java Build Path

Libraries 如下

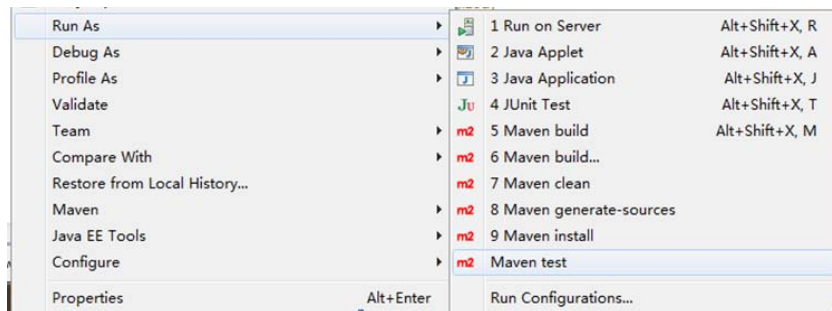


Source 如下，（MyEclipse 这里可能会按照 Maven 的标准结构显示，即多出 test 目录并以红叉显示，请将其移除）



4. 运行 maven 测试

工程上右键：

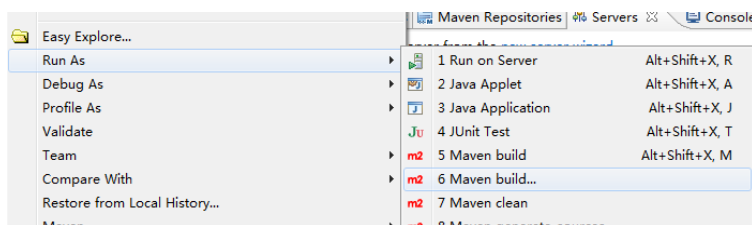


控制台没有错误出现就 ok 啦，下一步介绍 maven 项目的启动。

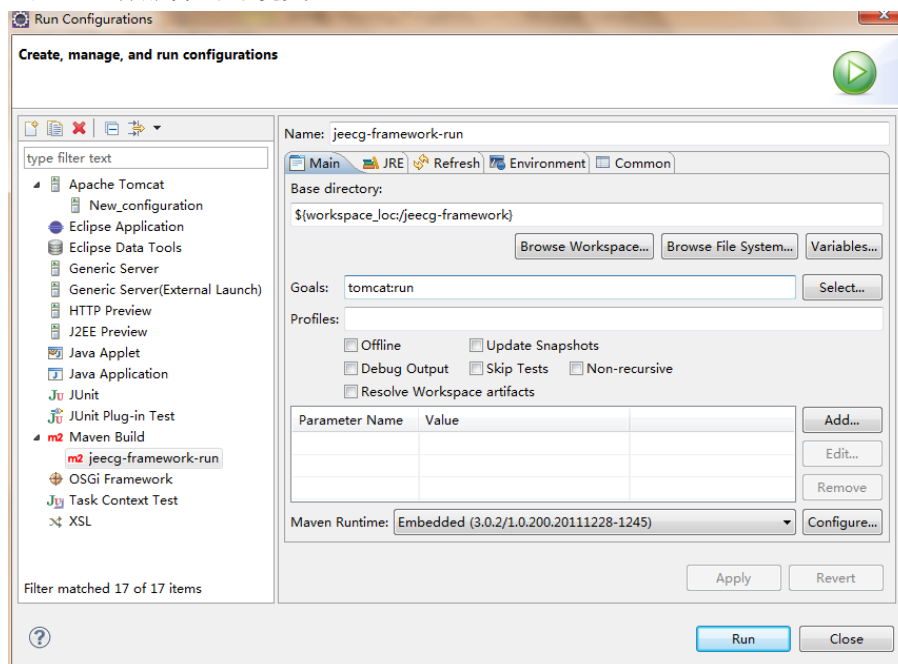
五. 启动 maven 项目的 3 种方式

1. 使用 maven 的 build 的方式启动

工程右键，运行 mavenbuild：



配置运行服务的环境变量：



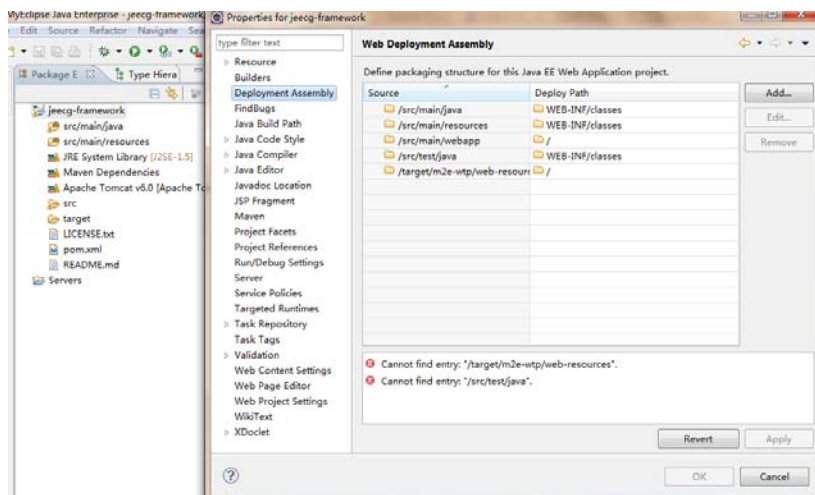
点击 Run 按钮即可以启动应用。

浏览器访问：<http://localhost:8080/jeecg>。

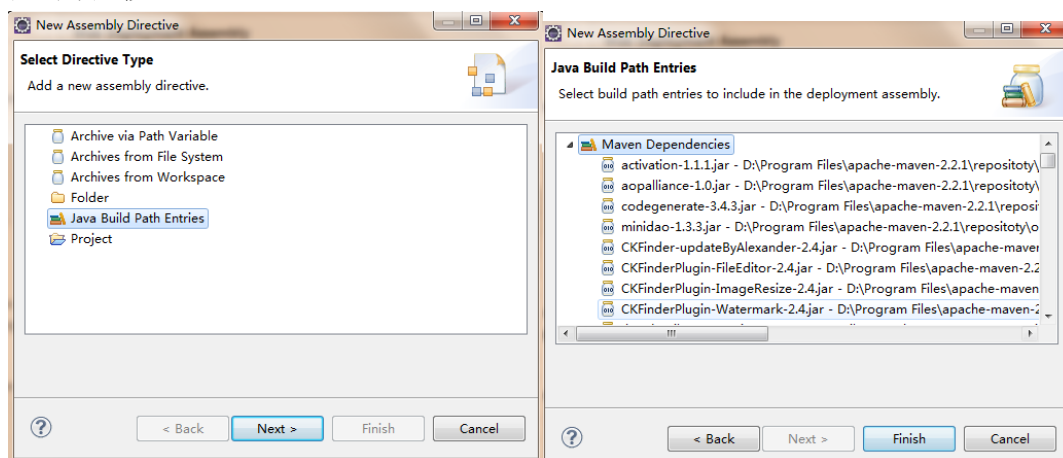
2. 使用 Eclipse 的 server 启动

工程右键，设置部署程序集（如果没有 Deployment Assembly 这个属性，在项目所在的工作区有一个 .project 文件，打开该文件增加如下黄色的这一行，用于 enable Deployment Assembly 属性：

```
<natures> ..... <nature>org.eclipse.wst.common.modulecore.ModuleCoreNature</nature> </natures> ):
```

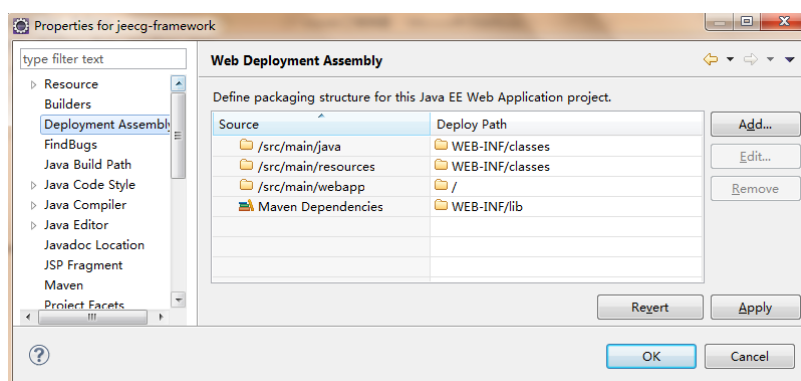
点击添加按钮:



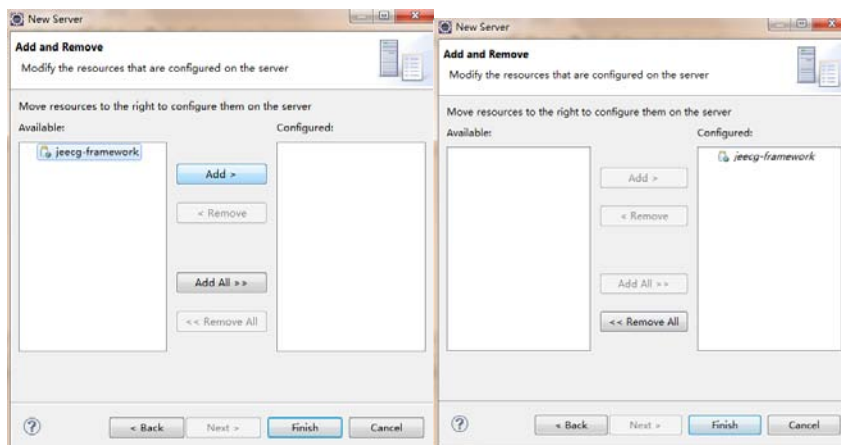
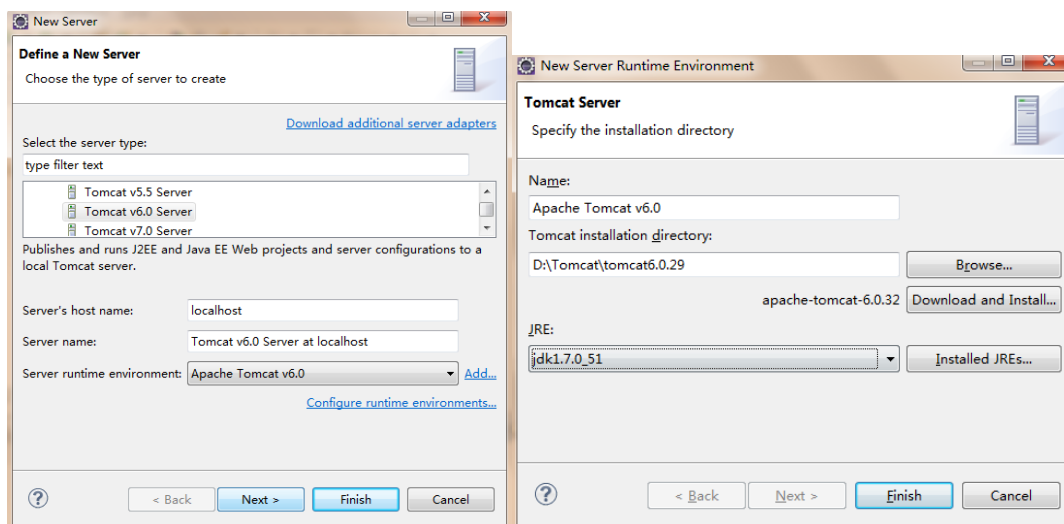
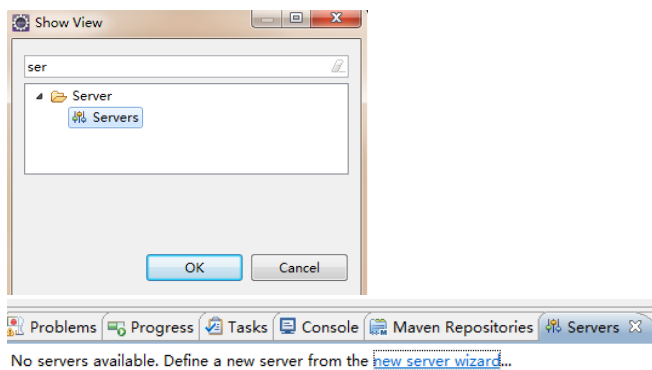
移除无用的选项

Cannot find entry: */target/m2e-wtp/web-resources*.
Cannot find entry: */src/test/java*.

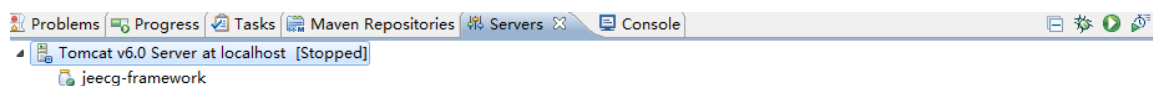
后的最终页面:



配置本地 Server 并启动:



右键 **start**，启动应用



浏览器访问：<http://localhost:8080/jeecg-framework>。

（如果访问页面时报：

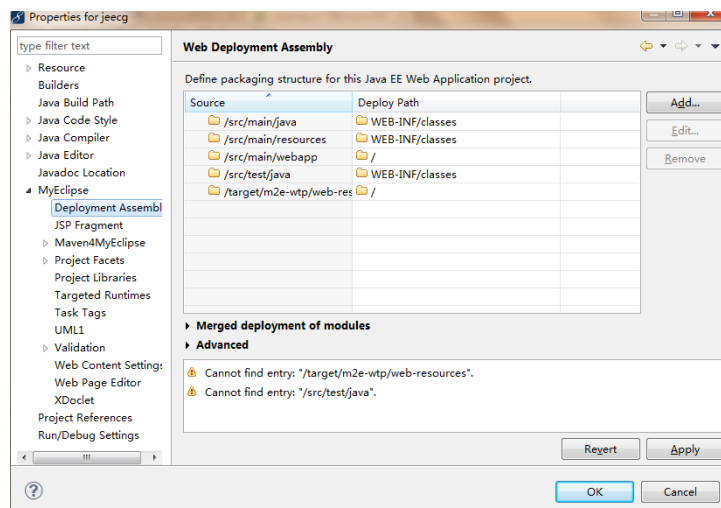
Failed to load or instantiate TagLibraryValidator class: org.apache.taglibs.standard.tlv.JstlCoreTLV

请查看你的工程目

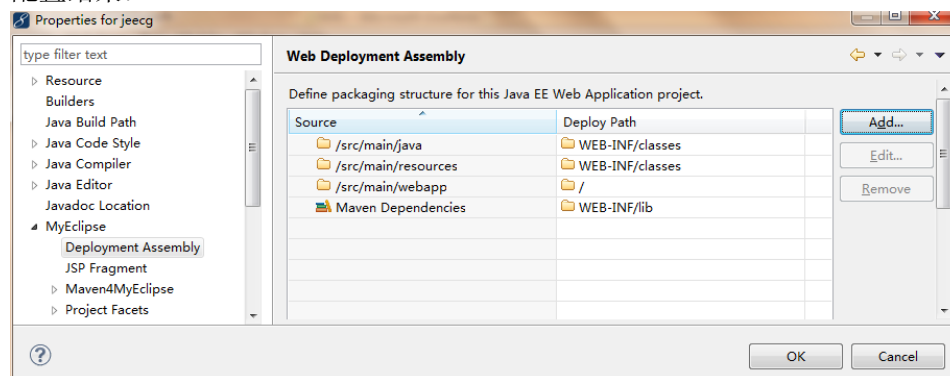
录.metadata\plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\jeecg-framework\WEB-INF\lib 下面是否有这两个 jsp-api.jar 和 servlet-api.jar 文件，将其删掉，重新启动 tomcat。

3. 使用 MyEclipse 的 server 启动

运行 build



配置结果:



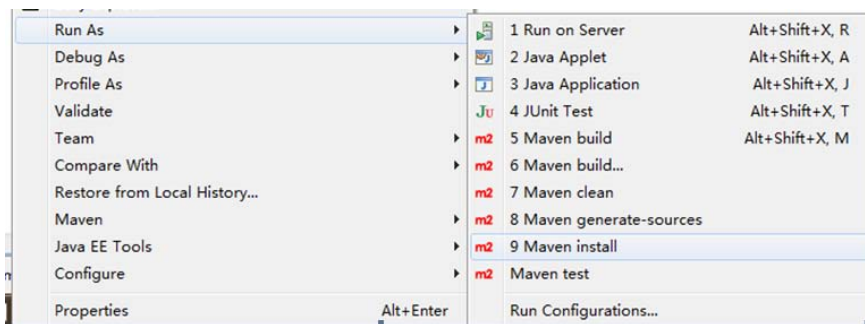
右键 start，启动应用:

Server	Status	Mode	Location
MyEclipse Derby	Stopped		
MyEclipse Tomcat	Running	Run	
jeecg	OK	Exploded	E:\zhanggm_project\.metadata\me_tc\t\wtpwebapps\jeecg-f...
MyEclipse Tomcat 7	Stopped		

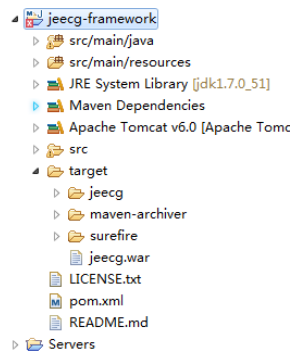
浏览器地址: <http://localhost:8080/jeecg-framework>。

六. 发布 war 包

工程右键:



编译通过后，target 目录下回生成 jeecg.war 包：



拷贝 jeecg.war 到 tomcat 的 webapps 下面，然后运行 tomcat 即可。

七. 错误信息

1. 控制台报以下错：

```
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @ jeecg ---
[INFO] Compiling 409 source files to E:\workspace\jeecg-framework\target\classes
[INFO] -----
[ERROR] COMPILATION ERROR :
[INFO] -----
[ERROR] \workspace\jeecg-framework\src\main\java\org\jeecgframework\core\annotation\config\AutoMenuOperation.java:[33,52] 不兼容的类型
找到: org.jeecgframework.core.annotation.config.MenuCodeType
需要: org.jeecgframework.core.annotation.config.MenuCodeType
[ERROR] \workspace\jeecg-framework\src\main\java\org\jeecgframework\core\common\service\impl\CommonServiceImpl.java:[106,28] 无法确定<T>T
的类型参数: 对于上限为T,java.lang.Object的类型变量 T, 不存在唯一最大实例
[ERROR] \workspace\jeecg-framework\src\main\java\org\jeecgframework\core\common\service\impl\CommonServiceImpl.java:[144,31] 无法确定<T>T
的类型参数: 对于上限为T,java.lang.Object的类型变量 T, 不存在唯一最大实例
[ERROR] \workspace\jeecg-framework\src\main\java\org\jeecgframework\core\common\service\impl\CommonServiceImpl.java:[304,29] 无法确定<T>T
的类型参数: 对于上限为T,java.lang.Object的类型变量 T, 不存在唯一最大实例
[ERROR] \workspace\jeecg-framework\src\main\java\org\jeecgframework\core\util\ToEntityUtil.java:[22,21] 无法确定<T>T 的类型参数: 对于上限为
T,java.lang.Object的类型变量 T, 不存在唯一最大实例
[INFO] 5 errors
```

这是因为你的编译器低于 1.7，请修改你的编译器为 1.7。

)

