

# JEESZ 分布式框架开发环境部署

作者：Jeesz

更新日期：2016-01-02

## 1. 环境准备

1. 开发工具： Eclipse IDE（建议使用高一点的版本）
2. JDK 版本： JDK1.7
3. 项目管理： Maven3.1.1

## 2. 安装步骤

1. 安装 jdk1.7 并配置环境变量（相信大家都会做，这里不再做重复的说明）
2. 安装 maven： maven 安装过程就是解压缩的过程（我已经帮大家解压了，目录： `E:\myprj\jeesz-maven\apache-maven-3.1.1`），为了方便大家，这边已经提供了相关的安装包，会在打包的目录下共享。
3. mavan 配置环境变量，右击我的电脑→属性→高级系统设置→环境变量→找到（系统变量 path），在最前面输入你的 maven 安装包路径，声明：因为上一步解压了指定的目录，所以配置如下：

`E:\myprj\jeesz-maven\apache-maven-3.1.1\bin;`

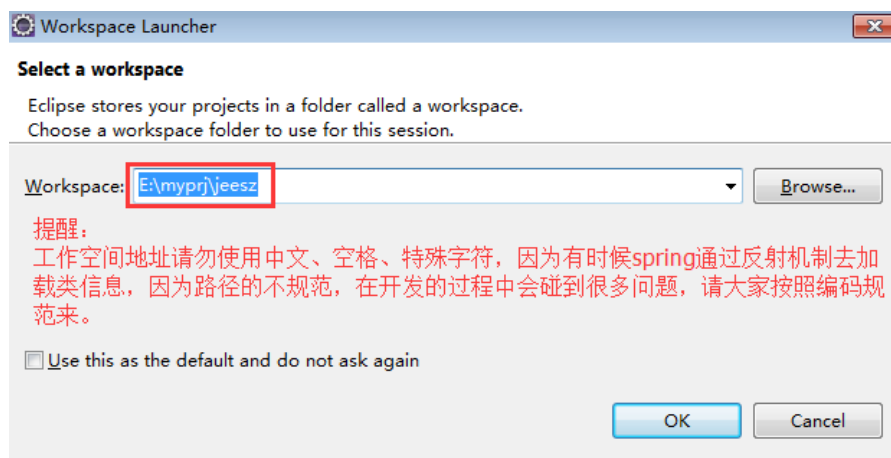
相信大家都有配过环境变量，如果您不是很清楚，请查询相关资料，了解一下 maven 的相关知识。

4. 打开 cmd，在里面敲： `mvn -version`，出现以下结果说明你 maven 安装成功了

```
C:\Users\Administrator>mvn -version
Apache Maven 3.1.1 (0728685237757fbbf44136acec0402957f723d9a; 2013-09-17 23:22:22+0800)
Maven home: D:\myprj\thinkgem-jeesite-e744ebc\doc\maven\maven3\bin\..
Java version: 1.6.0_43, vendor: Sun Microsystems Inc.
Java home: C:\Program Files\Java\jdk1.6.0_43\jre
Default locale: zh_CN, platform encoding: GBK
OS name: "windows 7", version: "6.1", arch: "amd64", family: "windows"
C:\Users\Administrator>
```

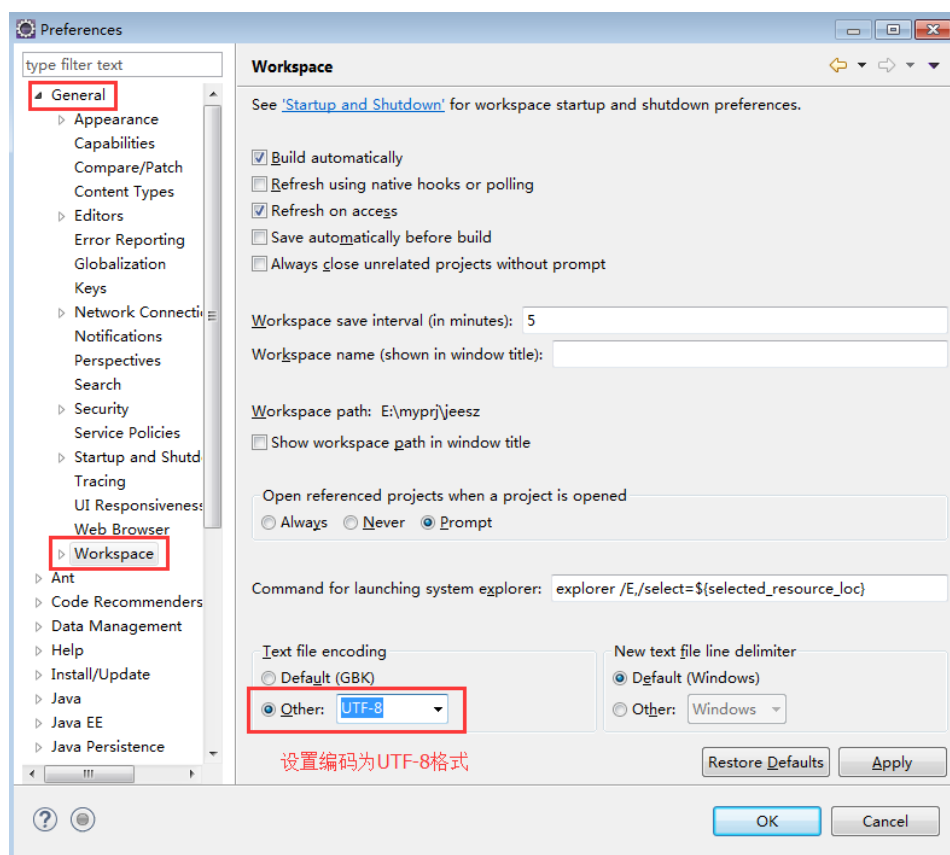
mvn 是 maven 的一个指令, mvn -version 是查看版本信息, 我的操作系统是 64 位的 WIN7, 安装的 maven 是 3.1.1, 如果能打印如上信息, 说明到此 Maven3 已经在你的电脑上安装完成。

## 5. 创建工作空间(进入后选中 j2ee 视图)



## 6. 配置 Eclipse 编码为 UTF-8

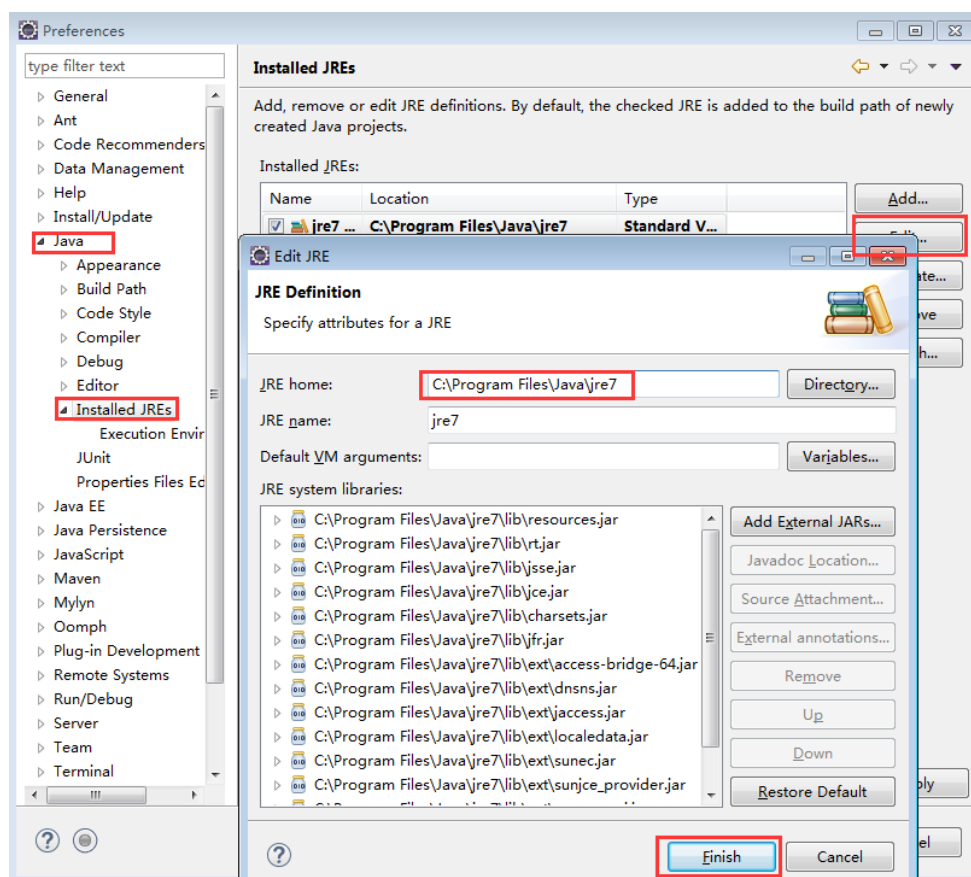
Eclipse 工具栏中选择 **Window→Prefrences→General→Workspace**



## 7. 配置 jre 环境(之前大家需要安装了 jdk1.7 版本, 这里就使用安装好的 jdk 进行配置):

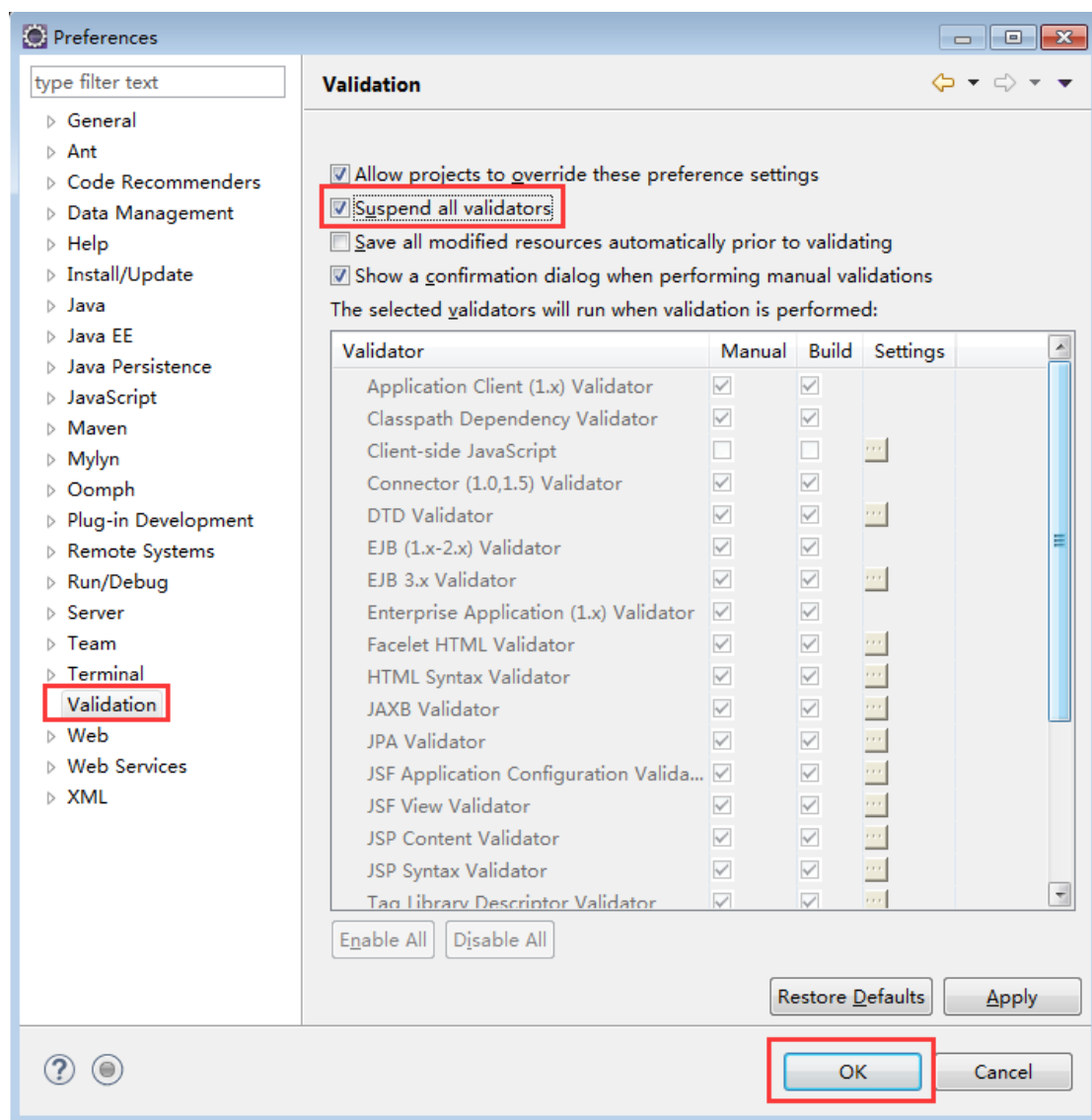
选择 **Window→Prefrences→java→Installed JREs→Edit→选择自己本**

地 jdk 安装路径→Finished→OK



8. 去除 JavaScript 的验证等信息（因为项目中大量使用 js、css、xml 等其他不同格式的文件，考虑到每次编译的执行时间，所以这边临时将一些验证信息去除）

Window→Prefrences→Validation→Suspend all validators→ok



9. 修改刚刚安装的 maven settings.xml 配置文件，路径如下：

E:\myprj\jeesz-maven\apache-maven-3.1.1\conf, 添加本地库地址：

<localRepository>E:/myprj/jeesz-maven/apache-maven-3.1.1/repository

</localRepository>

修改中央库路径（在网络正常的情况下，可以从远程中央库下载缺失的 jar 包,我这里使用了 Ochina 的中央库，如果网络有问题，大家可以使用自己公司的私服进行加载相关 jar 包），考虑到大家包有时候下载不下来，我这里也给大家共享了项目所需要的 jar 包，全部放到了以下目录：

E:/myprj/jeesz-maven/apache-maven-3.1.1/repository

```

<mirrors>
  <!-- mirror
  | Specifies a repository mirror site to use instead of a given repository
  | this mirror serves has an ID that matches the mirrorOf element of the repository
  | for inheritance and direct lookup purposes, and must be unique across the entire
  | configuration
  |
  <mirror>
    <id>mirrorId</id>
    <mirrorOf>repositoryId</mirrorOf>
    <name>Human Readable Name for this Mirror.</name>
    <url>http://my.repository.com/repo/path</url>
  </mirror>
  -->
  <!--
  <mirror>
    <id>mirrorId</id>
    <mirrorOf>repositoryId</mirrorOf>
    <name>Human Readable Name for this Mirror.</name>
    <url>http://repo1.maven.org/maven2/</url>
  </mirror>-->
</mirrors>

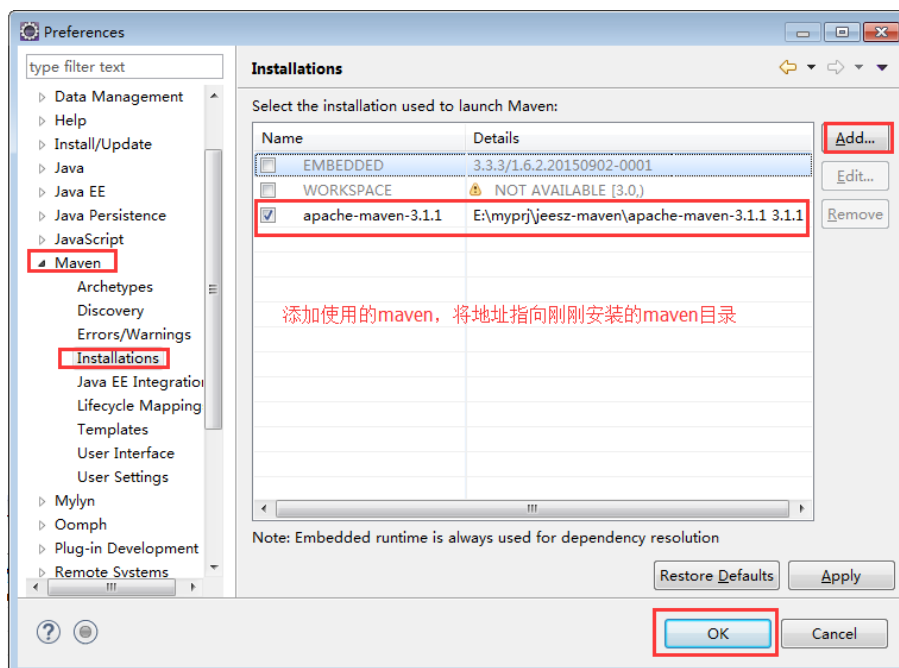
```

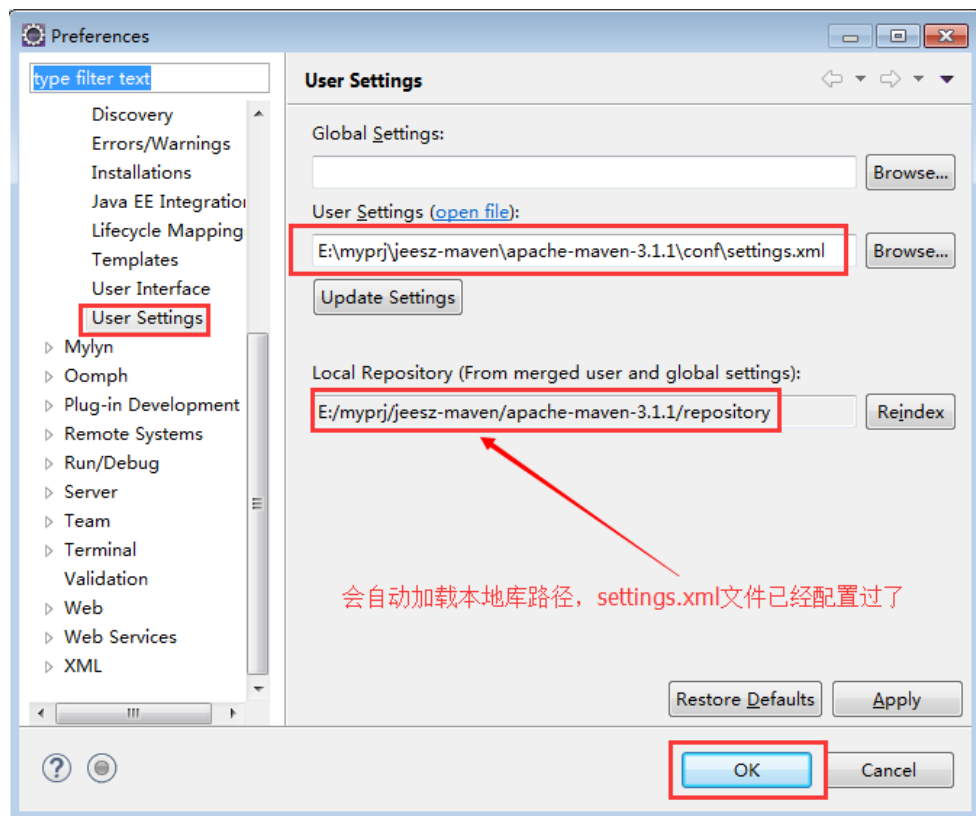
```

<mirror>
  <id>CN</id>
  <name>OSChina Central</name>
  <url>http://maven.oschina.net/content/groups/public/</url>
  <mirrorOf>central</mirrorOf>
</mirror>
</mirrors>

```

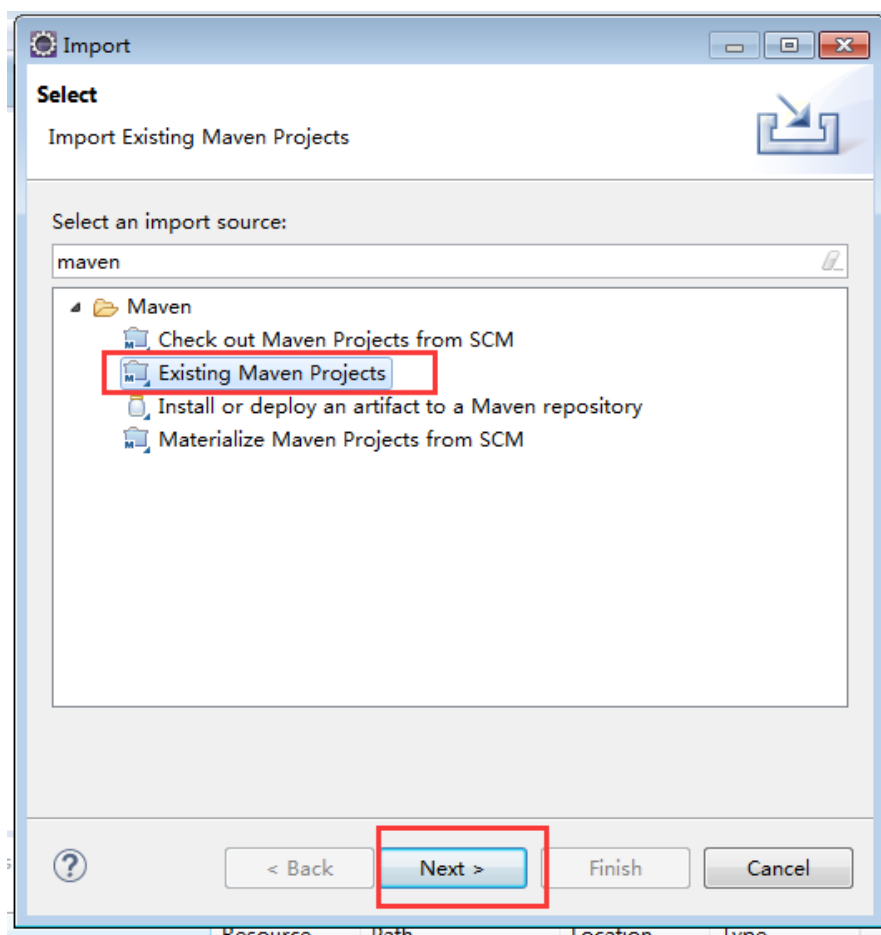
#### 10. Eclipse 中配置 maven 安装目录和 Setting 文件加载的本地库目录



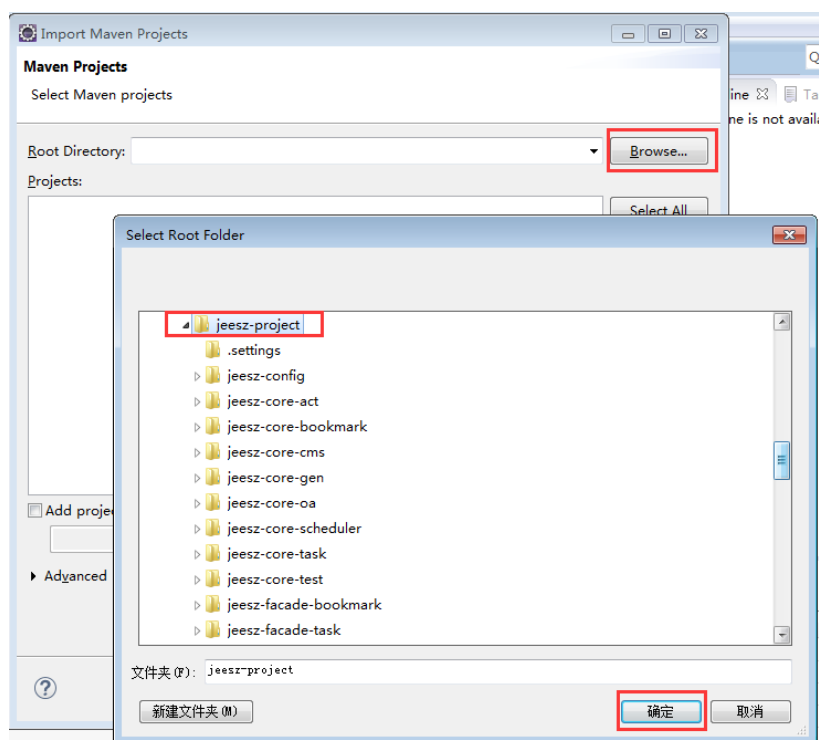


## 11. 导入 Maven 项目

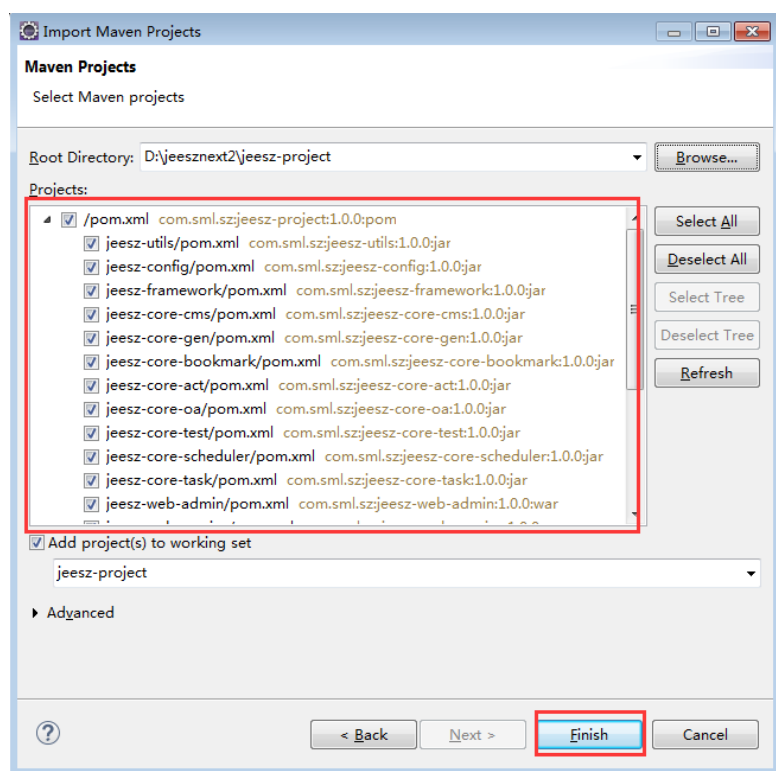
File→Import→Existing Maven Projects→下一步



选择你本地的 maven 项目（请选中 jeesz-project 根目录），点击确定：



加载出来所有项目后，点击 finish 完成

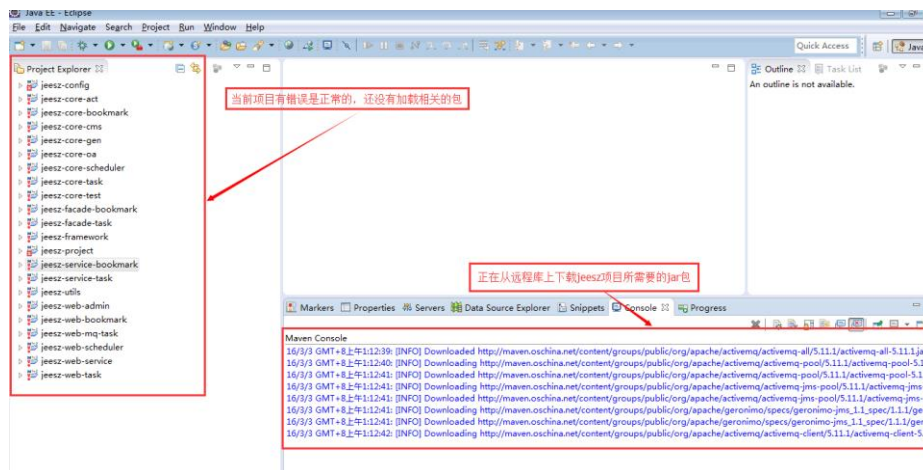


提醒：这个导入的过程要一段时间，因为我本地连接网络，所以会从远程中央库下载项目所需要的所有的 jar 包，请大家耐心等待。

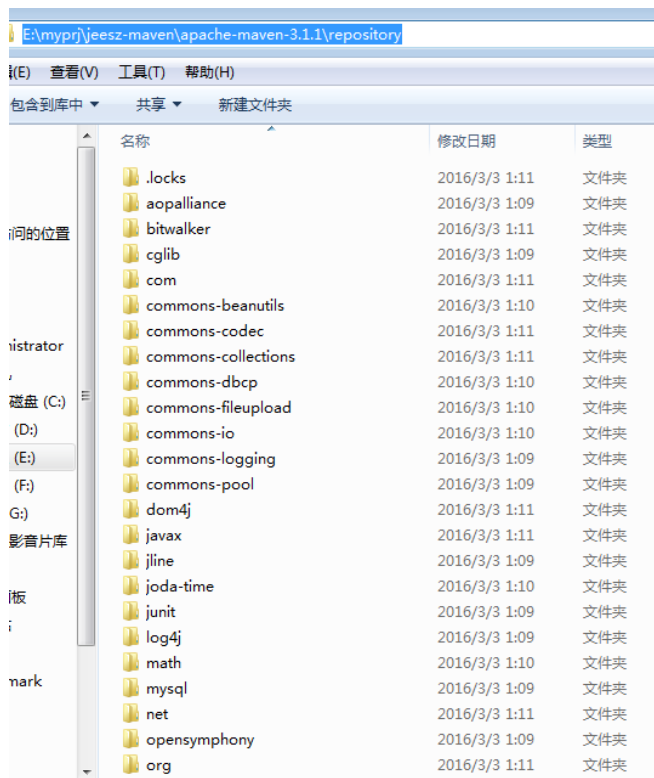
如果你使用了本地库则不需要等待很长时间，如果是使用公司私服，则等待时间也不会很长。



导入后的项目信息：

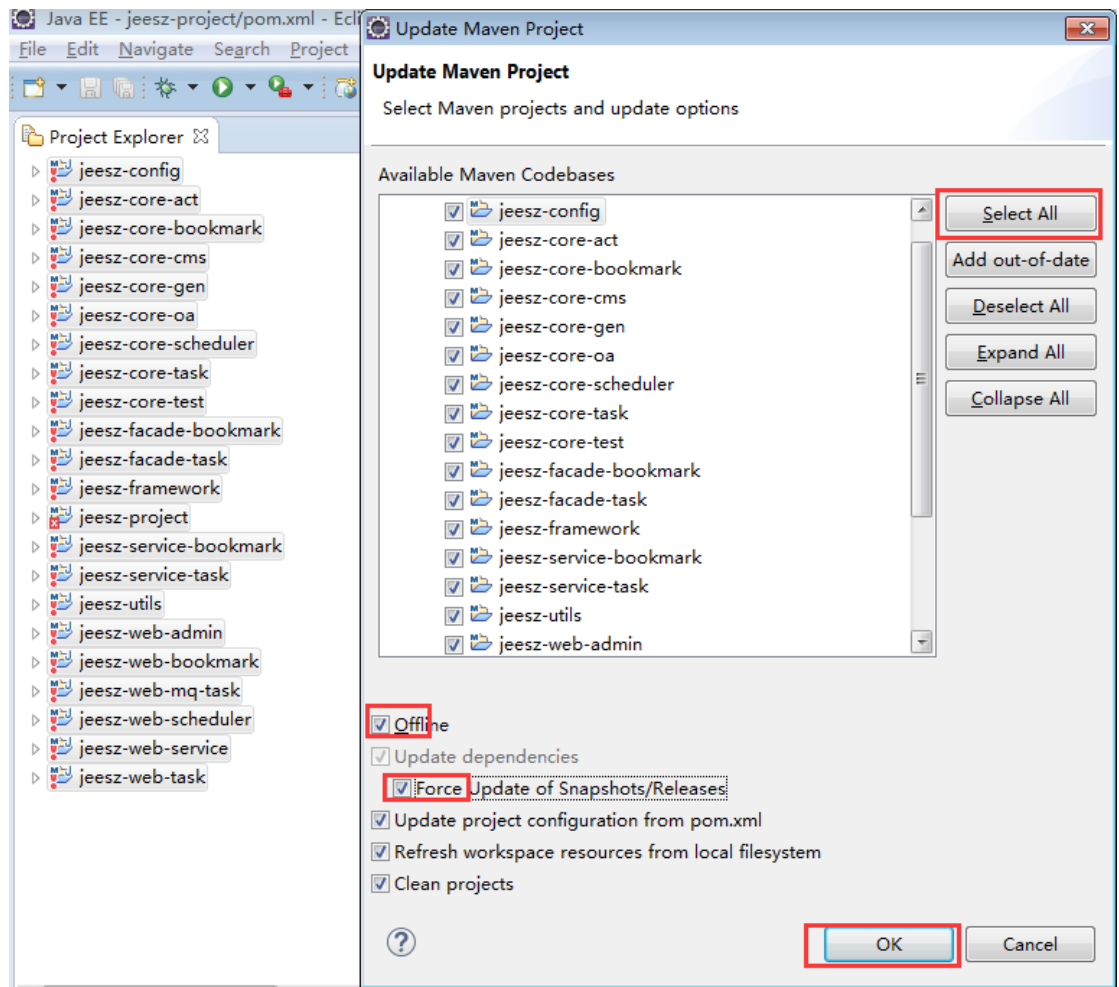


本地库的包已经下载了：



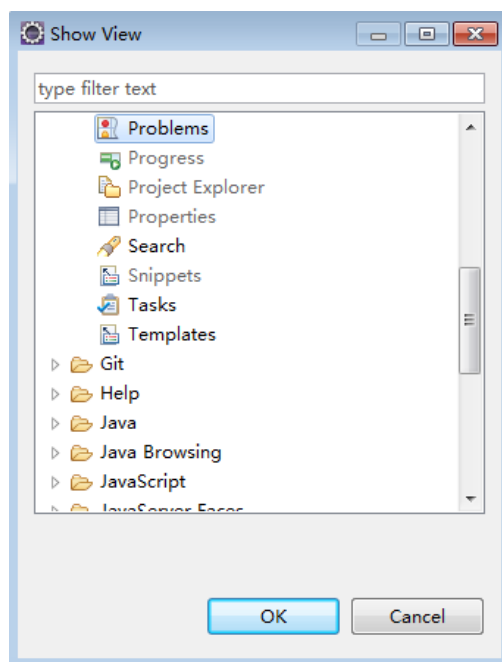
12. 下载完所有 jar 包，更新 maven 项目：

- 1) 选中所有项目
- 2) 右键选择 **maven**→**Update project...**→**选择所有**→**勾选关联项**→**点击 ok**



如果项目还出现红 X 错误，则必定是缺少相关的 jar 包，可以使用：

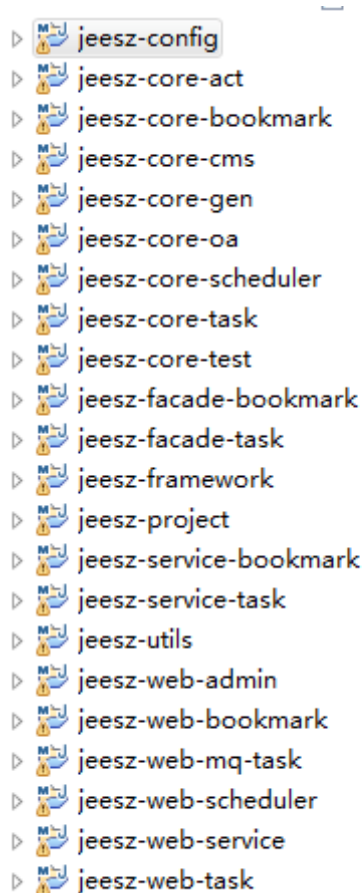
Window→show View→problem 窗口查看异常信息：



如果缺少 jar 包，可以将我们提供的 jar 包直接 copy 到你指定目录即可。

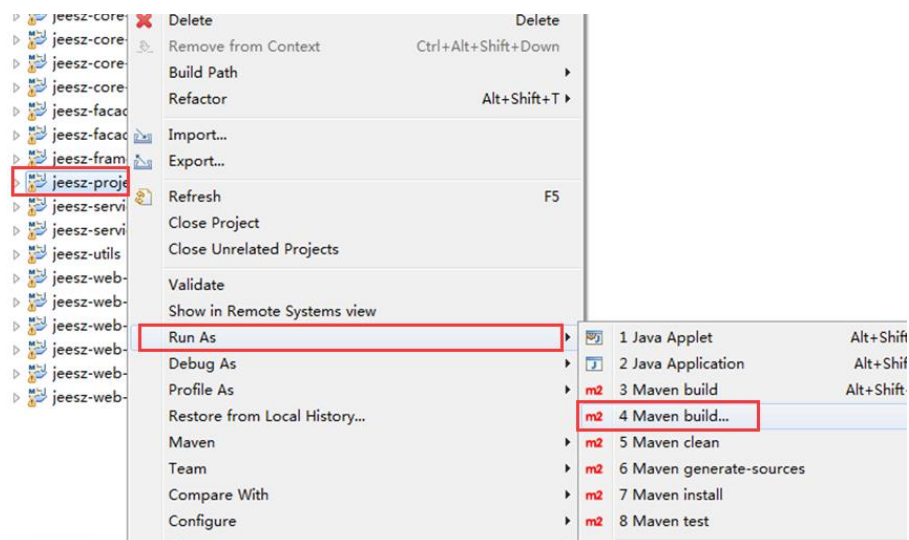
待项目里面的所有错误全部清零后才能进行下一步工作（如下图所示）

选中所有项目，对项目进行清理：Project→clean



### 13. 尝试对项目进行编译、打包看是否有问题：

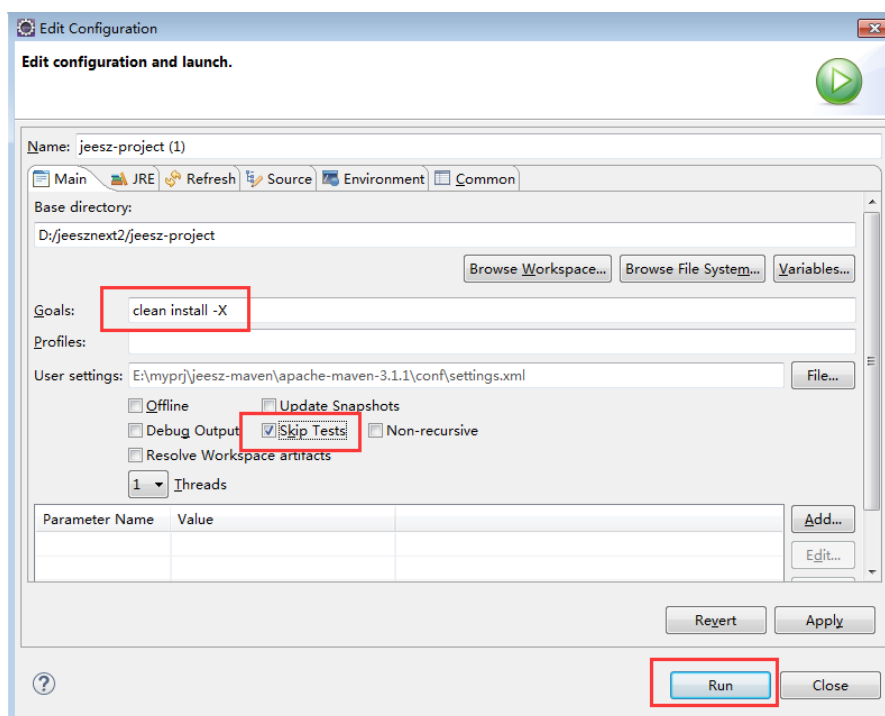
注意：一般我们打包编译是针对于根项目，因为里面配置了所有的依赖关系。右键 jeesz-project→Run As→Maven build...



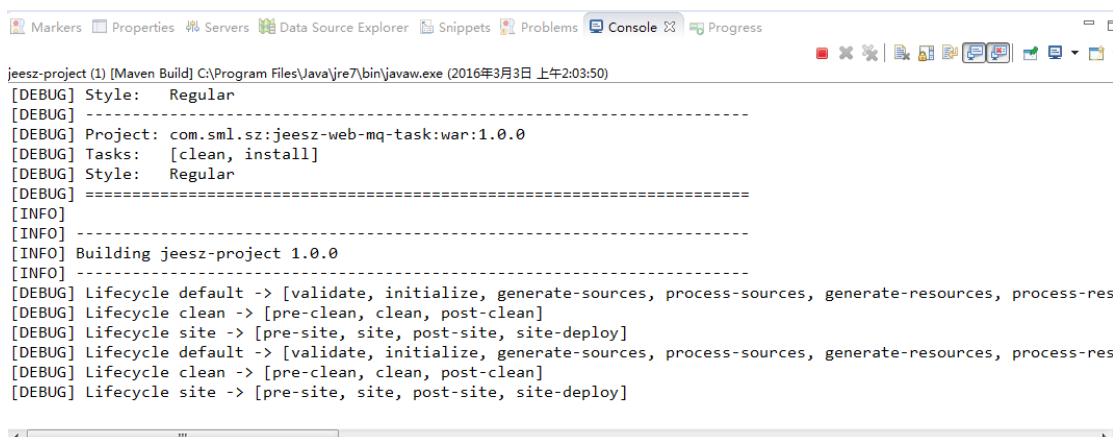
在弹出的对话框中录入 `clean install -X` 命令，勾选 **Skip tests**，Run...

提醒：如果对 `maven` 命令不是很熟的，可以去查下相关的资料

这边简单做下解释：`clean install -X` 表示清理所有项目并将 `jar` 包装载到本地库，其中 `-X` 表示输出的信息（一般在出现 `maven` 问题的时候可以使用 `-X` 来定位问题。）



此时控制台开始编译装载所有的包，这个过程有点漫长，请大家耐心等待。



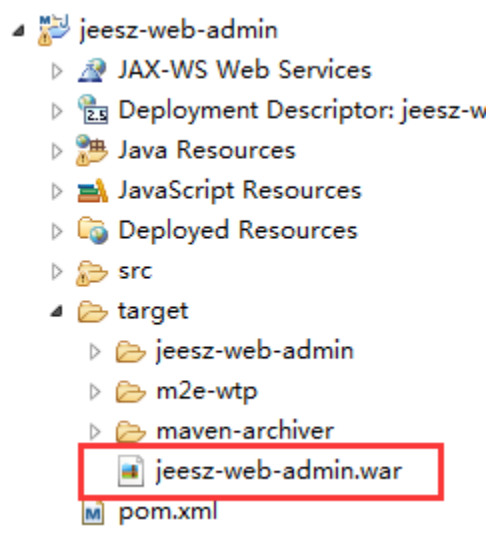
出现以下结果说明项目是正常的。

```

[INFO] Reactor Summary:
[INFO]
[INFO] jeesz-project ..... SUCCESS [1.109s]
[INFO] jeesz-utils ..... SUCCESS [11.922s]
[INFO] jeesz-config ..... SUCCESS [0.109s]
[INFO] jeesz-framework ..... SUCCESS [5.447s]
[INFO] jeesz-core-cms ..... SUCCESS [2.013s]
[INFO] jeesz-core-gen ..... SUCCESS [1.951s]
[INFO] jeesz-core-bookmark ..... SUCCESS [1.499s]
[INFO] jeesz-core-act ..... SUCCESS [2.169s]
[INFO] jeesz-core-oa ..... SUCCESS [2.045s]
[INFO] jeesz-core-test ..... SUCCESS [1.529s]
[INFO] jeesz-core-scheduler ..... SUCCESS [2.341s]
[INFO] jeesz-core-task ..... SUCCESS [1.515s]
[INFO] jeesz-facade-bookmark ..... SUCCESS [1.498s]
[INFO] jeesz-web-admin Maven Webapp ..... SUCCESS [2:42.334s]
[INFO] jeesz-web-service Maven Webapp ..... SUCCESS [59.934s]
[INFO] jeesz-web-scheduler Maven Webapp ..... SUCCESS [1:36.274s]
[INFO] jeesz-facade-task ..... SUCCESS [7.364s]
[INFO] jeesz-web-task Maven Webapp ..... SUCCESS [2:07.646s]
[INFO] jeesz-web-bookmark ..... SUCCESS [3:36.369s]
[INFO] jeesz-service-bookmark Maven Webapp ..... SUCCESS [12.407s]
[INFO] jeesz-service-task Maven Webapp ..... SUCCESS [6.027s]
[INFO] jeesz-web-mq-task Maven Webapp ..... SUCCESS [32.145s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----

```

打好的包(包含 war、jar),都在 target 文件夹下,这边只以 jeesz-web-admin 为实例截图如下:



#### 14. 使用 Tomcat 发布 jeesz 项目（发布过程不做详细介绍，大家都会做）：

下面对各个不同的项目进行介绍：

Jeesz-web-admin: 可以独立部署（**后台管理平台**）

Jeesz-web-service:可以独立部署（**REST 服务平台**，客户可以根据自己的业务去拓展 Rest 服务，里面已经写好了相关的实例）

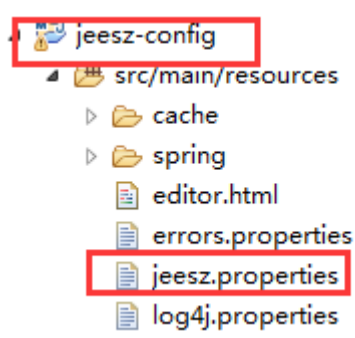
Jeesz-service-bookmark、jeesz-service-task 为 dubbo 服务提供者（**这边只通过收藏和待办项目来对 dubbo 服务项目开发做相关介绍，客户可以根据自己的业务去拓展 dubbo 服务，可以同时部署，也可以拆分部署**）

Jeesz-web-bookmark、jeesz-web-task 为 dubbo 消费者平台，其中 dubbo 服务对外暴露的服务接口为：Jeesz-facade-bookmark、jeesz-facade-task（这边只通过收藏和待办项目来描述 dubbo 客户端怎样调用 dubbo 服务提供者的过程，客户可以将 dubbo 消费端植入到不同的项目中，只需要做一些简单配置就可以了）

部署条件：

- 1) 执行 mysql 数据库脚本：jeesz.sql

数据库 jdbc 的配置项可以参考



```
jdbc.type=mysql
jdbc.driver=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://localhost:3306/jeesz?useUnicode=true&characterEncoding=utf-8
jdbc.username=root
jdbc.password=root
```

- 2) Jeesz-web-admin 可以直接通过 tomcat 部署

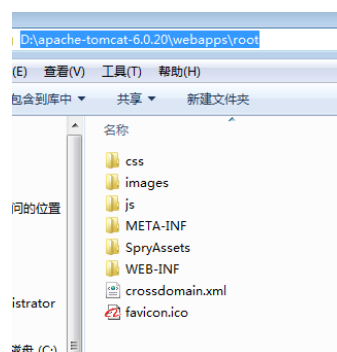
登陆用户名密码：jeesz/admin

- 3) Jeesz-web-service:可以通过 tomcat 独立部署

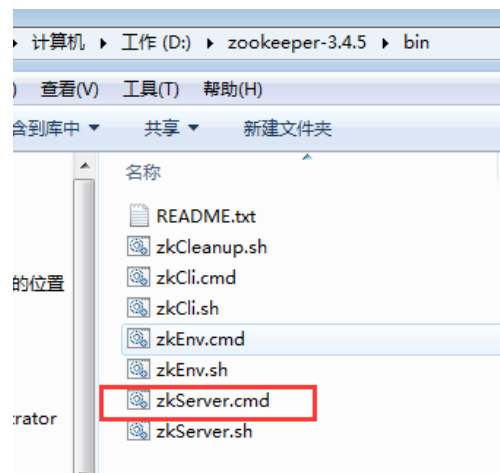
登陆用户名密码：jeesz/admin

- 4) 部署 dubbo 相关的项目需要具备以下条件：

- 安装 dubbo 管控平台，将 war 包解压到 tomcat，并将目录替换为根目录，直接启动 tomcat 即可（开发环境提供 window）



- Zookeeper 注册中心安装，我这边提供了 zookeeper-3.4.5 版本（开发环境提供 window），进入安装目录启动即可



提醒：我这边已经给提供了相关的程序，其中 **dubbo 管控平台**要独立部署在 **Tomcat** 服务器上，这个不需要进行高可用配置，直接启动 **tomcat** 就可以启动。**Zookeeper** 注册中心有相关的命令可以直接运行启动。如果想部署在 **linux** 系统上，可以看我们提供的相关文档即可。

- 5) 待 dubbo 管控台和注册中心启动 ok，则将 Jeesz-service-bookmark、jeesz-service-task 项目同时（可以独立部署）部署到 tomat 中，为 dubbo 服务提供者
- 6) 登陆 dubbo 管控台查看相关的服务信息，可以看到你发布的服务已经由管控台进行管控了：

<http://localhost:8080>

直接通过地址可以访问是因为将包名改为了 Root

用户名密码：root/root

- 7) 分部部署 Jeesz-web-bookmark、jeesz-web-task dubbo 消费者项目，目的是让开发人员了解服务是怎样调用的。

到此整个项目的安装过程先到这里，请各位开发人员安装完项目后投入开发，所有的业务直接服务化，因为项目是基于驱动开发的，故将你们的表设计好，通过 jeesz-web-admin 项目的 gen 代码生成模块自动生成所有的 controller、

service、dao、xml、jsp 界面等，直接进行业务编写处理。近期会不断的完善文档，请时刻关注。