



# 智慧开源基础开发引擎(ZRpower)

Version 7.0

(SpringBoot+Mybatis+Thymeleaf+Vue+Element 版本)

(环境安装及项目结构技术手册)

|      |                                |      |                                                                     |
|------|--------------------------------|------|---------------------------------------------------------------------|
| 文件名称 | 智慧开源基础开发引擎( 环境安装及项目结构手册 ) V7.0 | 文件版本 | V1.2                                                                |
| 文件编号 | ZR-JSWD-VE01                   | 受控状态 | <input checked="" type="checkbox"/> 受控 <input type="checkbox"/> 不受控 |
| 编制部门 | 技术研发部                          | 文件页数 |                                                                     |
| 编制人  | 付天龙、梁文楷                        | 编制日期 | 2021-09-08                                                          |
| 批准人  | 徐仲学                            | 批准日期 | 2021-09-08                                                          |



网址: <http://www.zrpower.cn>

# 目 录

|                                        |    |
|----------------------------------------|----|
| 第一章 版权说明.....                          | 3  |
| 第二章 前端 VUE 环境搭建.....                   | 4  |
| 2.1 前端 Vue 各软件版本.....                  | 4  |
| 2.2 安装 Node.js.....                    | 4  |
| 2.3 安装 Visual Studio Code.....         | 7  |
| 第三章 后台开发环境搭建.....                      | 7  |
| 3.1 基础开发引擎项目的各软件版本.....                | 7  |
| 3.2 用 Eclipse+jdk1.8 搭建 Java 开发环境..... | 7  |
| 3.3 Eclipse 开发工具配置 jdk.....            | 21 |
| 3.4 Eclipse 开发工具配置 tomcat.....         | 24 |
| 第四章 VUE 配置及说明.....                     | 34 |
| 4.1 框架整体.....                          | 34 |
| 4.2 config 目录.....                     | 34 |
| 4.2.1 dev.env.js.....                  | 34 |
| 4.2.2 index.js.....                    | 35 |
| 4.2.3 prod.env.js.....                 | 35 |
| 4.3 src 目录.....                        | 35 |
| 4.3.1 assets 资源目录.....                 | 35 |
| 4.3.2 components 组件目录.....             | 35 |
| 4.3.3 router 静态路由目录.....               | 36 |
| 4.3.4 store 动态路由目录.....                | 36 |
| 第五章 框架配置及说明.....                       | 37 |
| 5.1 项目的软件架构图.....                      | 37 |
| 5.2 项目结构说明.....                        | 38 |
| 5.3 项目的核心技术说明.....                     | 38 |
| 5.4 Eclipse 开发工具的 Maven 配置.....        | 43 |
| 5.5 打包部署操作说明.....                      | 45 |
| 5.6 只集成引擎时说明.....                      | 54 |
| 5.7 创建数据表及演示数据.....                    | 55 |
| 5.8 Mysql 数据库乱码处理.....                 | 55 |
| 第六章 开发技术文档捐赠.....                      | 57 |

## 第一章 版权说明

本项目为免费和开源项目, 本文档为付费文档, 版权归贵州智慧开源软件有限公司所有, 并保留一切权利, 本文档及其描述的内容受有关法律的版权保护, 对本文档和开源代码以任何形式的非法复制、泄露或散布到网络提供下载, 都将导致相应的法律责任。

文档更新

本文档由贵州智慧开源软件有限公司于 2021 年 9 月最后修订。

## 第二章 前端 VUE 环境搭建

Vue.js (读音 /vju:/, 类似于 view) 是一个构建数据驱动的 web 界面的渐进式框架。Vue.js 的目标是通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件。

### 2.1 前端 Vue 各软件版本

Node.js 版本: 12.22.6

Npm 版本: 6.11.3

Vue 版本: 2.5.21

Element-ui 版本: 2.15.5

### 2.2 安装 Node.js

前端开发框架和环境都是需要 Node.js, 先安装 node.js 开发环境, vue 的运行是要依赖于 node 的 npm 的管理工具来实现, 下载 <https://nodejs.org/en/>, 安装完成之后, 打开 cmd 开始输入命令。(我用的是 win10 系统, 所以需要管理员权限, 右键点击以管理员身份运行 cmd), 不然会出现很多报错。

下载

最新的长期支持版本: 10.16.0

| v10.16.0<br>推荐大多数用户使用                | v12.4.0<br>最新的特性               |                             |
|--------------------------------------|--------------------------------|-----------------------------|
| Windows 安装包<br>node-v10.16.0-x64.msi | macOS 安装包<br>node-v10.16.0.pkg | 源代码<br>node-v10.16.0.tar.gz |

阿里云 高性能云服务器2折起 0.73元/日, 节省80%运维成本

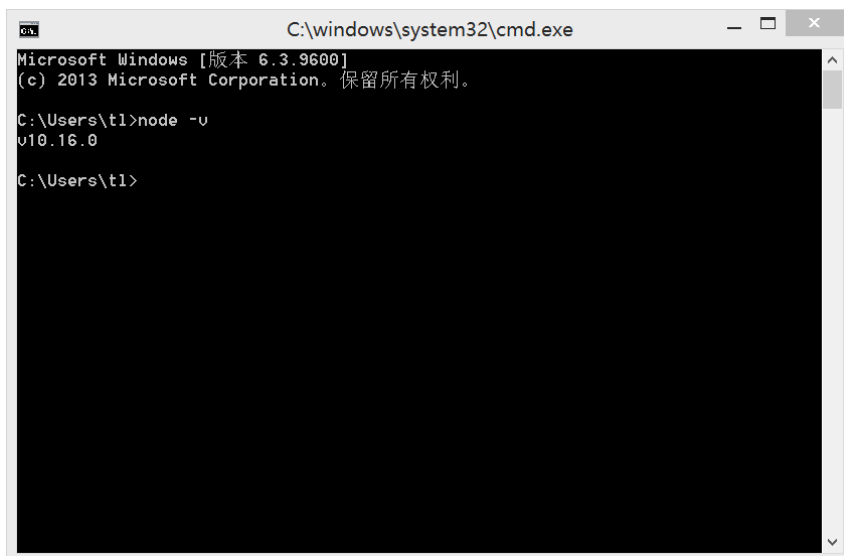
| Windows 安装包 (.msi)   | 32 位  | 64 位  |       |
|----------------------|-------|-------|-------|
| Windows 二进制文件 (.zip) | 32 位  | 64 位  |       |
| macOS 安装包 (.pkg)     | 64 位  |       |       |
| macOS 二进制文件 (.pkg)   | 64 位  |       |       |
| Linux 二进制文件 (x64)    | 32 位  |       |       |
| Linux 二进制文件 (ARM)    | ARMv6 | ARMv7 | ARMv8 |
| Docker 镜像            | 官方镜像  |       |       |
| 全部安装包                | 阿里云镜像 |       |       |

[https://blog.csdn.net/YH\\_ang\\_eng](https://blog.csdn.net/YH_ang_eng)

第 1 步, 下载完毕后, 可以安装 node, 建议不要安装在系统盘 (如 C: )。



第 2 步，安装完 Node.js 后，打开电脑终端查看 Node 版本。以 Windows 为例，打开 cmd 命令窗口输入：node -v，查看 node 版本。



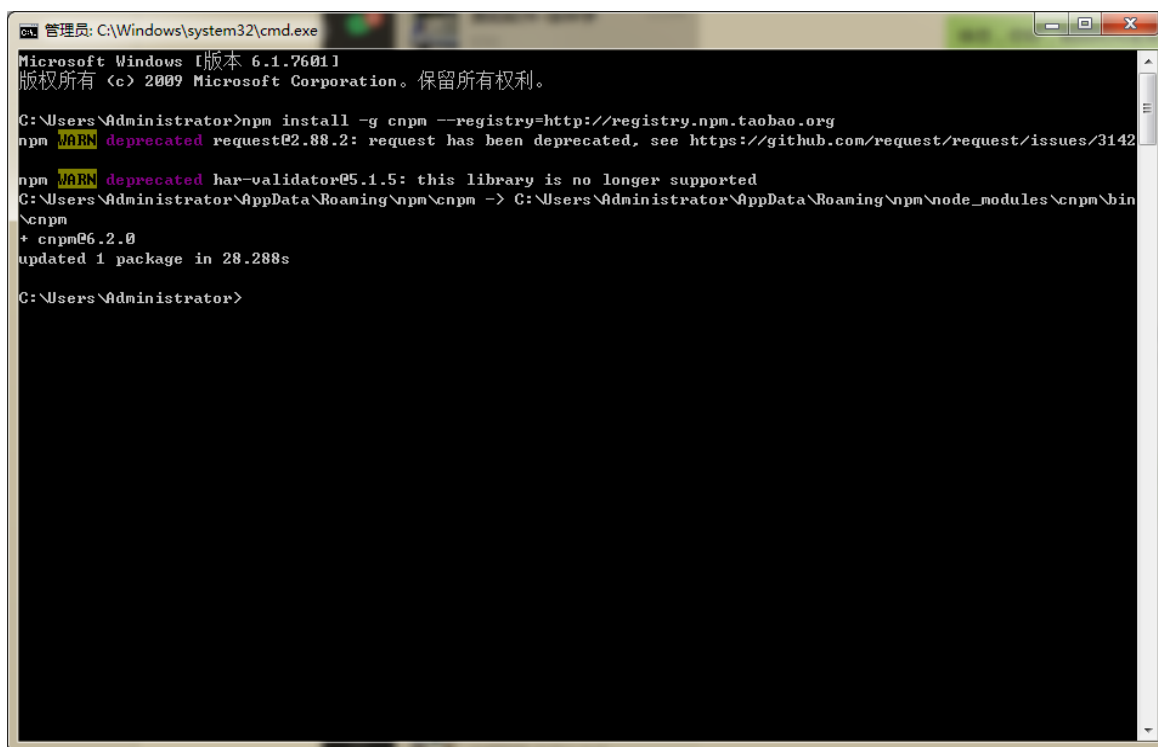
```
C:\windows\system32\cmd.exe
Microsoft Windows [版本 6.3.9600]
(c) 2013 Microsoft Corporation. 保留所有权利。

C:\Users\tl>node -v
v10.16.0

C:\Users\tl>
```

第 3 步, 安装 cnpm 淘宝镜像。淘宝镜像的 cnpm 来代替 npm 的安装, 速度会快很多。打开终端窗口, 并输入以下命令:

```
npm install -g cnpm --registry=http://registry.npm.taobao.org。
```



```
管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation. 保留所有权利。

C:\Users\Administrator>npm install -g cnpm --registry=http://registry.npm.taobao.org
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
C:\Users\Administrator\AppData\Roaming\npm\cnpm -> C:\Users\Administrator\AppData\Roaming\npm\node_modules\cnpm\bin\cnpm
+ cnpm@6.2.0
updated 1 package in 28.288s

C:\Users\Administrator>
```

安装过程持续 5 分钟左右, 耐心等待。

第 4 步, 基本环境搭建完成后就可以开始 Vue 开发了。首先需要安装 vue 的脚手架工具: vue-cli, 然后初始化一个基本项目。输入:

```
#安装全局 vue-cli
npm install - global vue-cli
#cmd 进入项目目录
cd D:\ vue-zrshiro
#安装项目依赖
npm install
# 运行测试, 访问 http://localhost:8081
npm run dev
# 项目打包
npm run build
```

## 2.3 安装 Visual Studio Code

去官网上下下载: <https://code.visualstudio.com/>。下载之后, 双击安装, 安装完之后左侧栏那边是英文, 如何变为中文: 按快捷键 `ctrl+shift+p`, 输入配置显示语言或者英文 `Configure Display Language`, 安装完成之后, 重启即可。

# 第三章 后台开发环境搭建

注: 支持 `jdk1.8` 及 `jdk1.8` 以上的所有版本, 满足此条件的 `java` 开发环境都可以集成我们的基础开发引擎, 已经搭建好开发环境的跳过本章。

## 3.1 基础开发引擎项目的各软件版本

JAVA 运行环境: `jdk1.8` 及以上

SpringBoot: `2.5.3`

## 3.2 用 Eclipse+jdk1.8 搭建 Java 开发环境

我们的程序支持所有的 `java` 开发环境, 已经搭建好开发环境的忽略本节。其它开发工具的搭建不再介绍, 请自行查找相关资料。

## 一、前期的准备

### 1. jdk1.8 的下载

(1) 百度搜索 JDK, 点击方框里的第一个。



(2) 下载 JDK。



(3) 选中 Accept License Agreement, 然后在 Winows x86 (32 位系统的下这个) 和 Windows x64 (64 位系统的下这个) 中选择 一个下载, 如果不清楚自己的系统是 32 位还是 64 位, 可以百度, 或者直接下载 32 位的 Windows x86。



## 2. Eclipse 的下载

(1) 百度搜索 Eclipse, 选择第一个。



(2) 选择方框里边的内容进行下载, 这个必须和前面下载的 JDK 版本保持一致, 前边下载的 32 位 JDK 这里就点击 32bit, 前边下载的 64 位 JDK, 这里就点击 64bit。

...or download an Eclipse Package



### Eclipse IDE for Java EE Developers

275 MB | 1,450,069 DOWNLOADS

Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn...

Windows  
32 bit | 64 bit



### Run your Apps on the Cloud

Want Eclipse in the cloud? IBM Bluemix makes it easy. Sign up to begin building today!

Promoted Download



### Eclipse IDE for Java Developers

166 MB | 708,142 DOWNLOADS

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven integration and WindowBuilder...

Windows  
32 bit | 64 bit

(3)继续点击方框中的内容，开始下载。

## Eclipse downloads - Select a mirror

All downloads are provided under the terms and conditions of the [Eclipse Foundation Software User Agreement](#) unless otherwise specified.

Download from: **China - Dalian**  
**Neusoft University of Information**  
(大连东软信息学院) (<http>)

File: eclipse-java-mars-1-win32-x86\_64.zip

Checksums: MD5 SHA1  
SHA-512

**DOWNLOAD**

OR >

### Get It Faster from our Members



#### IBM

Blazingly fast downloads hosted by IBM Bluemix.

**DOWNLOAD**

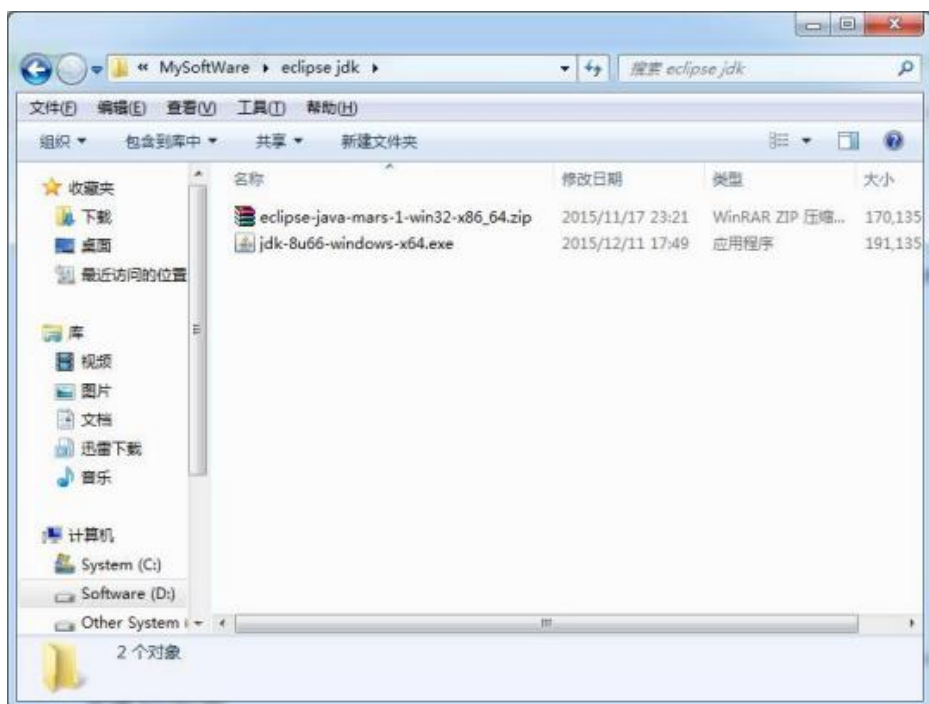


#### Spring by Pivotal

Rapid downloads of Eclipse packages. Free

## 二、JDK 的安装

1. 下载的文件在这里，双击 `jdk-8u66-windows-x64.exe` 开始安装。



2. 平时我们安装软件不要安装在 C 盘，容易导致开机慢，改到 D 盘，点击“更改”。



把 C 盘改成 D 盘:



下一步:



开始安装...



3. 继续安装 Jre, 点击“更改”。



找到刚才我们安装 JDK 的路径, D:\Program Files\Java, 并在此目录下新建一个文件夹 jre1.8.0\_66, 并选中。



下一步:



jre 安装中...

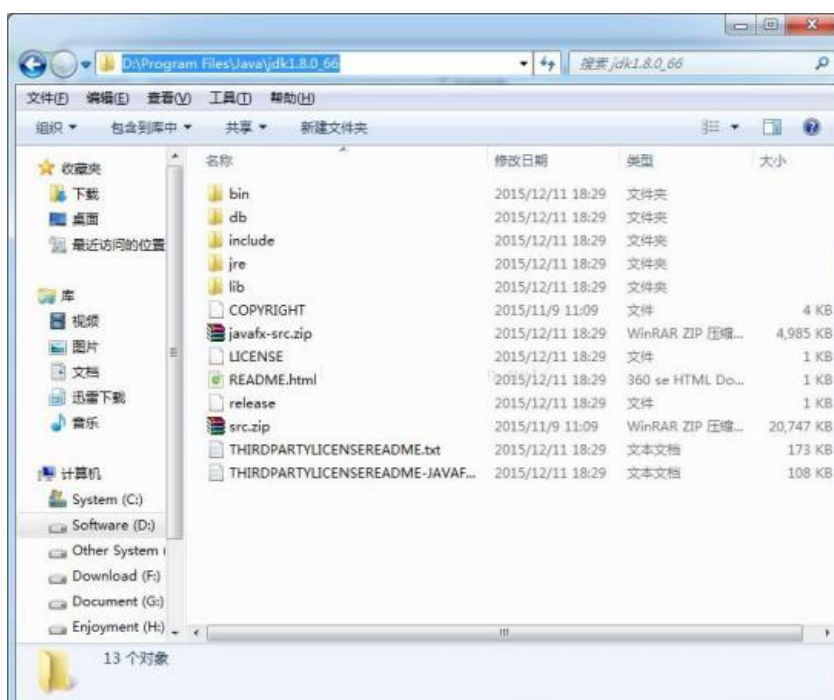


4. 安装完成，点击关闭。

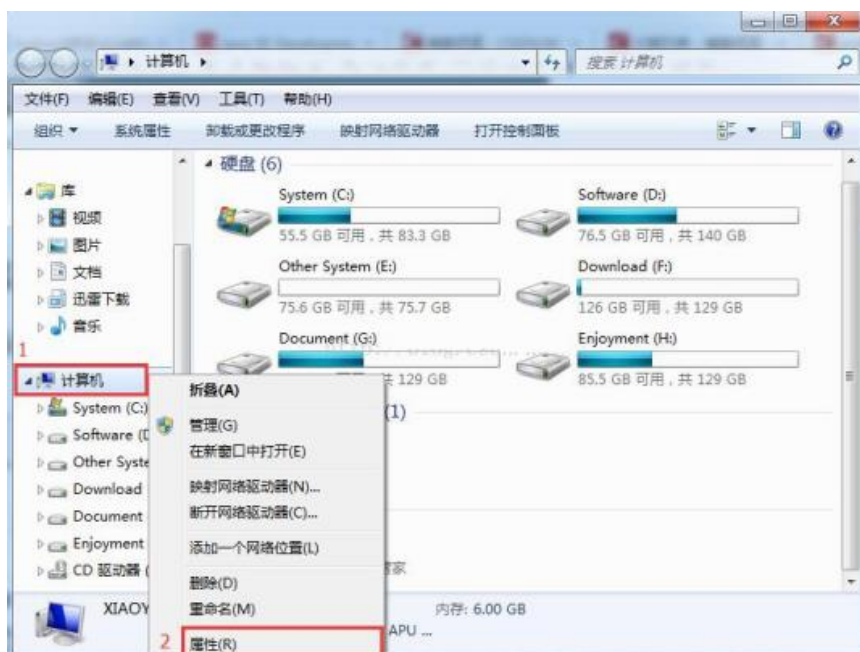


### 三、JDK 环境变量的配置

1. 找到自己 JDK 的安装目录, 复制其安装路径, 我的是 D:\Program Files\Java\jdk1.8.0\_66。



2. 在计算机上右键--属性。

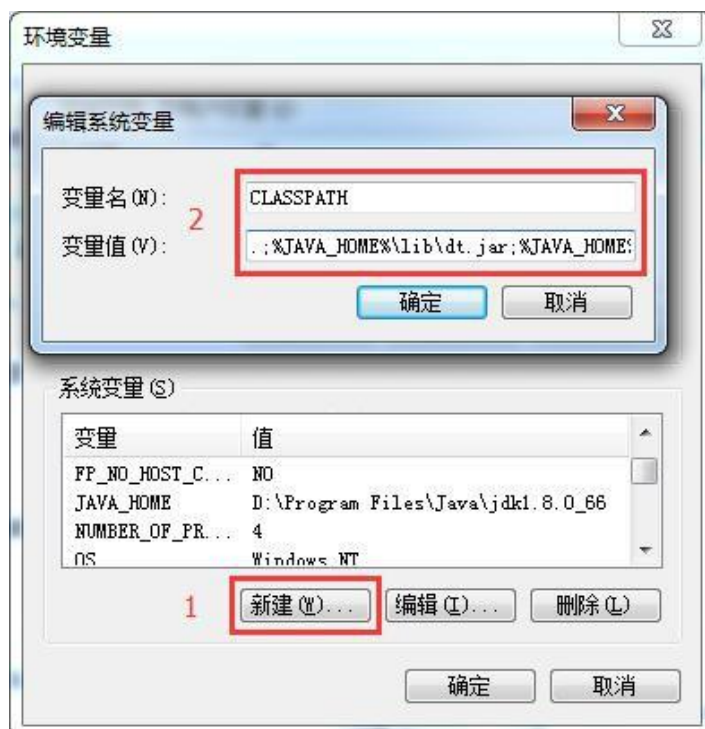


3. 按下图的顺序依次点击高级系统设置—环境变量—新建，在变量名中填写 JAVA\_HOME, 在变量值中填写第 2 步中复制的路径，并确定。



4. 再次点击新建，变量名填 CLASSPATH, 变量值填下边这个，最好是直接复制我的，每一个标点符号都要复制下去，并确定。

.;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar;

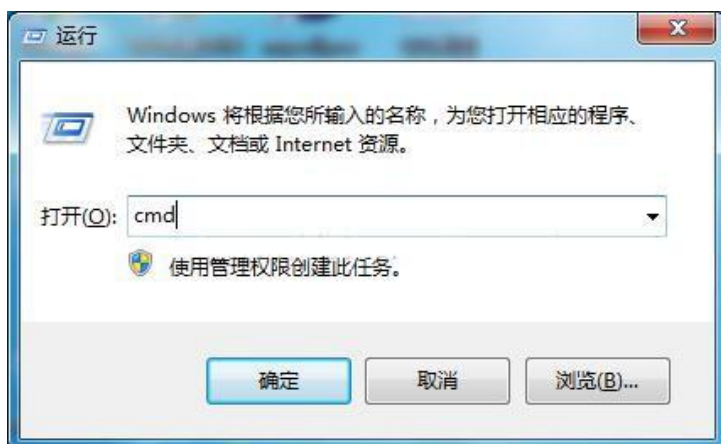


5. 在下方的系统变量中找到 Path, 如方框 1 所示, 选中并点击编辑, 在变量值的最前边添加下面这一行 (注意是添加, 千万别把以前本来有的内容删掉了), 也是最好直接复制我的, 每一个标点符号都要复制下去, 添加完成后一直点确定直到所有窗口关闭。

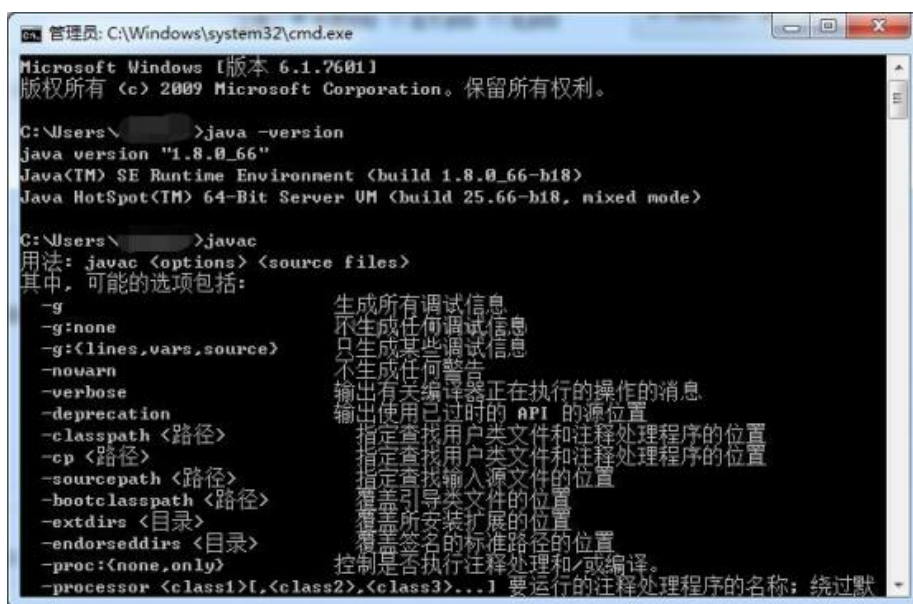
%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin;



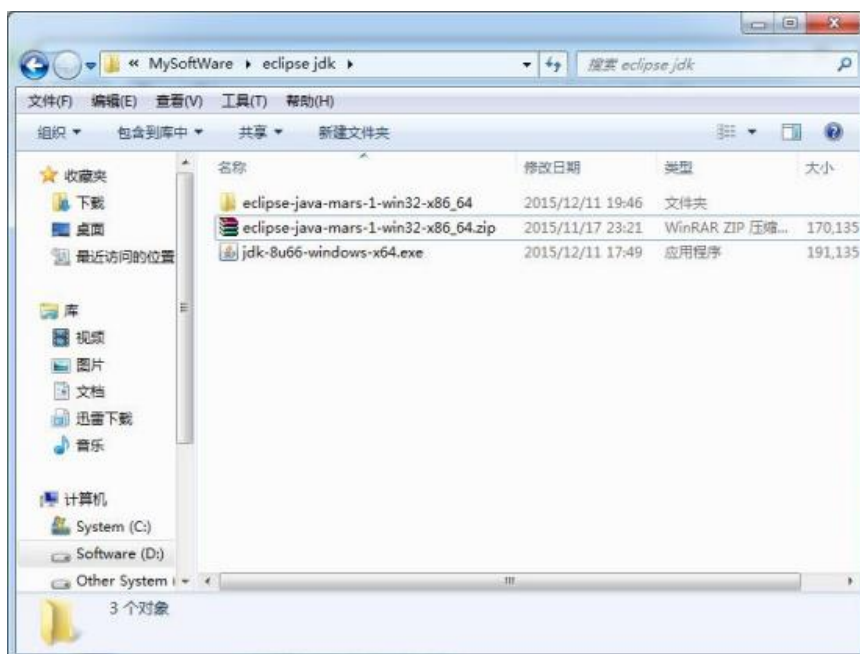
6. 同时按住 Windows 图标键+R 键，弹出如下窗口，输入 cmd，并回车。



7. 依次输入 `java -version` 和 `javac` 并回车，如果出现一下信息则表明 JDK 安装成功！阶段性的胜利！

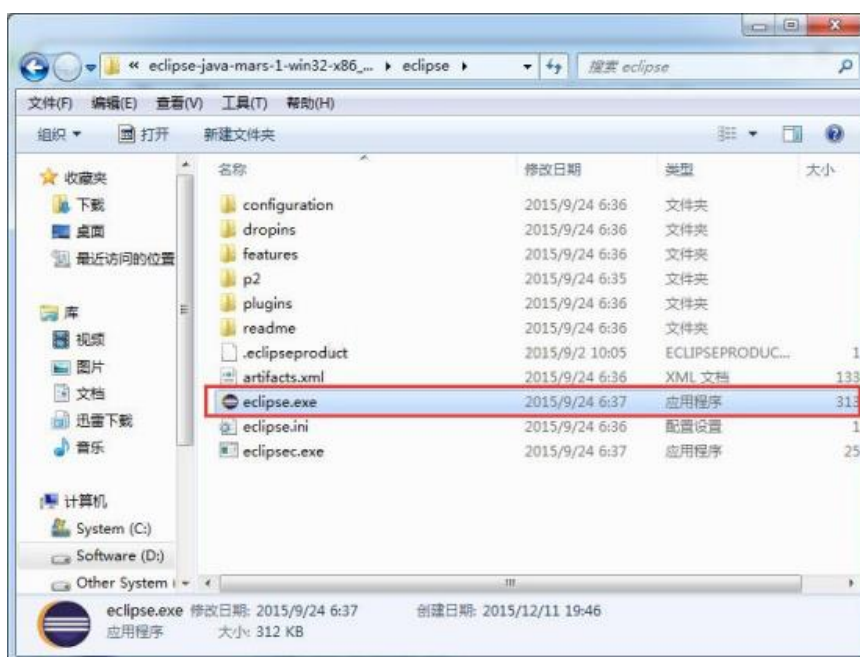


#### 四、Eclipse 的安装配置

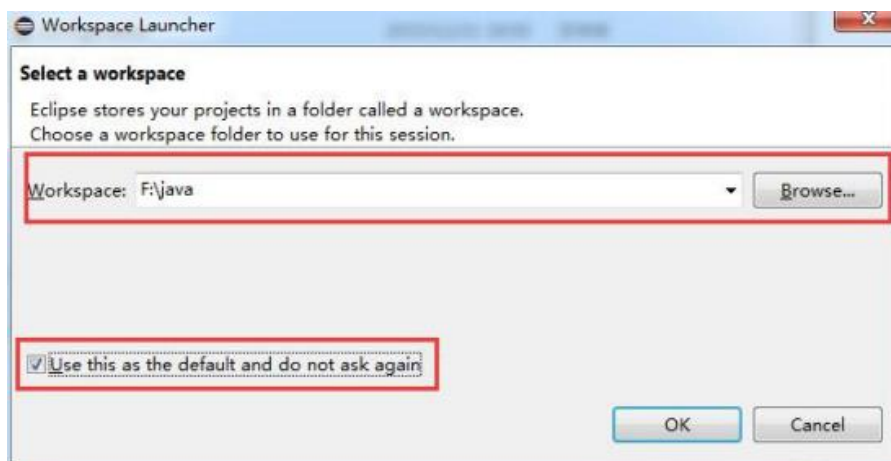


1. 解压开下载好的 Eclipse。

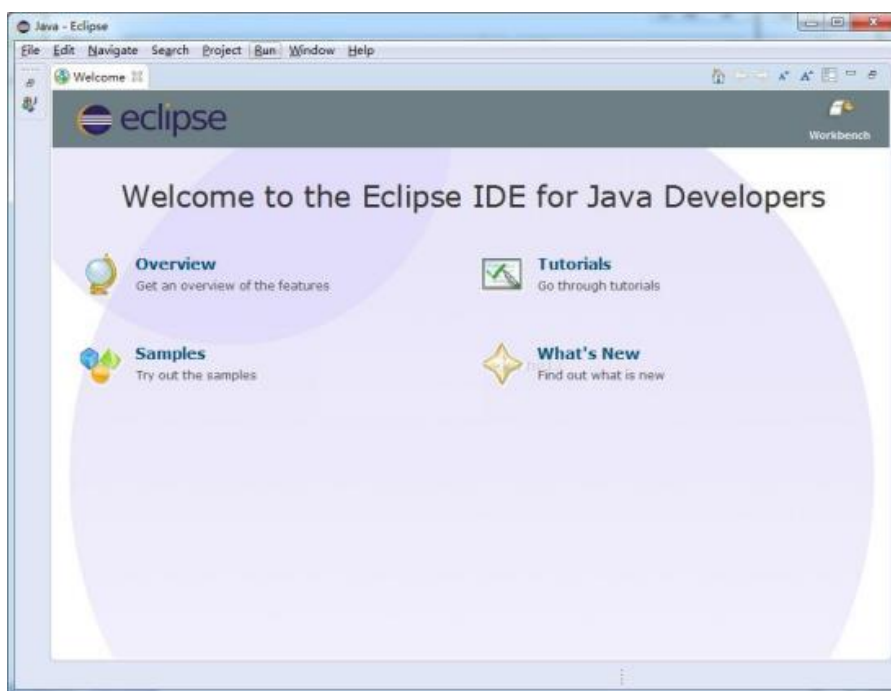
2. 进入到这个目录，并双击 eclipse.exe 启动 Eclipse。



3. 这里是要你选择工作区间，也就是你代码存放的地方，建议在 E 盘、F 盘等位置新建一个文件夹来单独存放代码，并选择左下方的勾，并 OK。



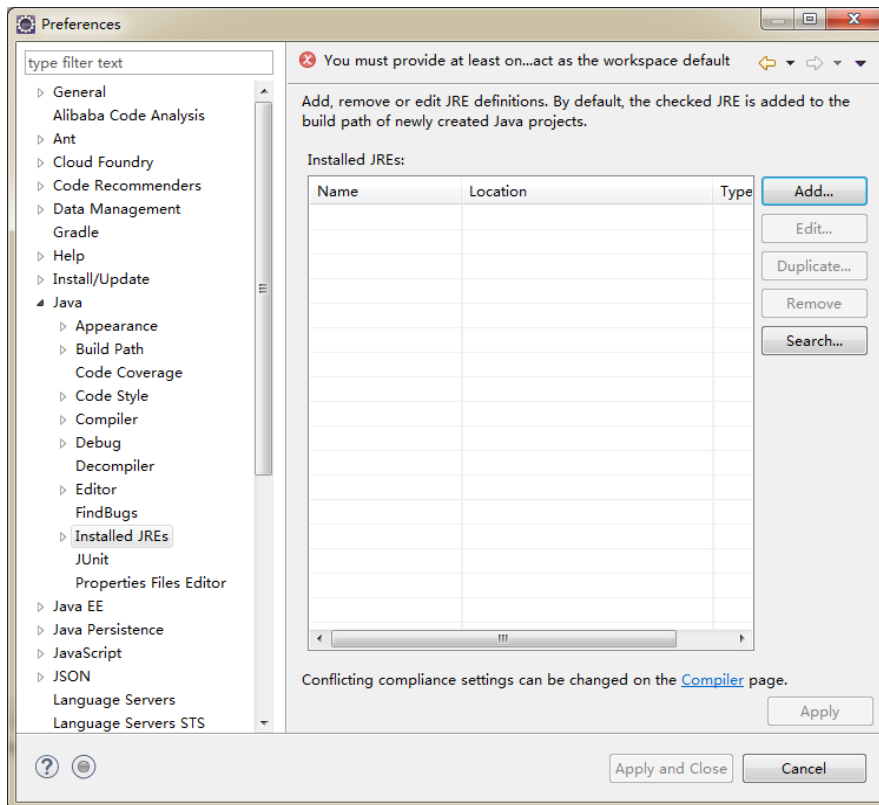
#### 4. 安装成功!



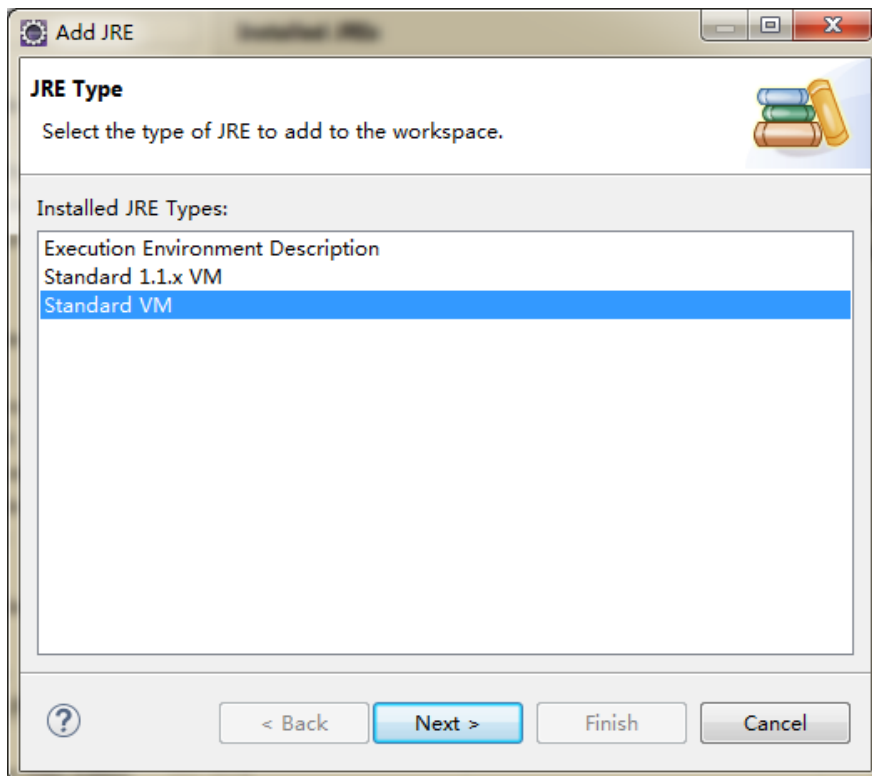
### 3.3 Eclipse 开发工具配置 jdk

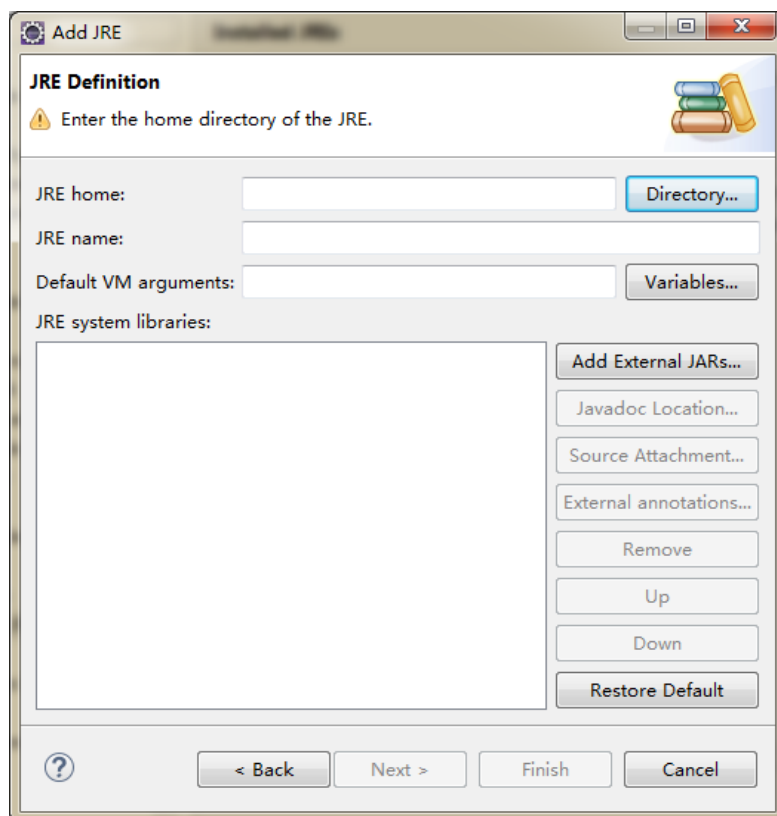
Eclipse 开发工具是要编译并运行 java 项目的,所以需要在 Eclipse 开发工具上配置 jdk 环境。

打开 Eclipse, 单击“Window”菜单, 选择下方的“Preferences”选项。



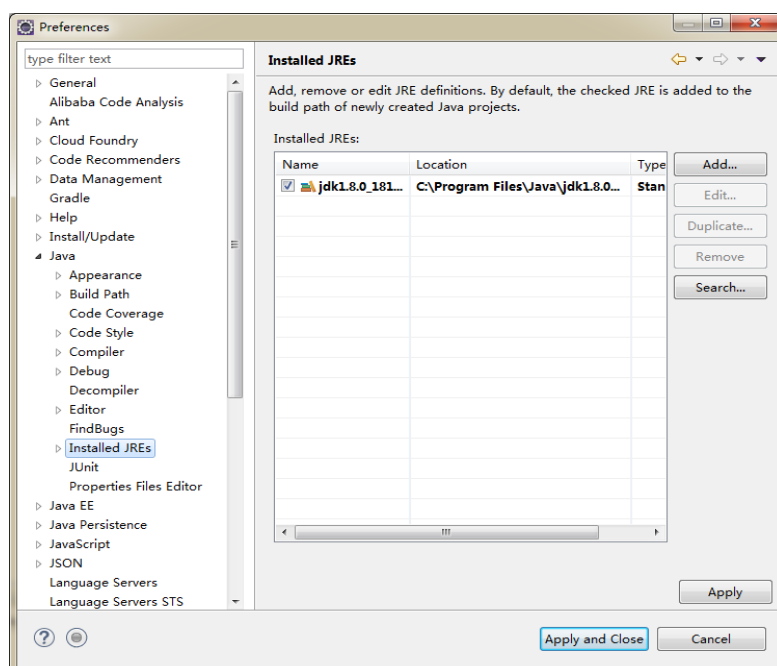
点击 Java -> Installed JREs, 在右侧点击 Add 按钮, 选择 Standard VM, 点击 Next。





JRE home 为当前 jdk 安装的路径, 点击 Directory 选择当前 jdk 安装的路径 (D:\Program Files\Java\jdk1.8.0\_66), JRE name 为当前 jdk 的一个名称, 默认 jdk1.8.0\_66, 然后点击 Finish。

说明: jdk 是带有编译和运行 java 的 \*.class 文件功能的, 而 jre 只有运行 java 的 \*.class 文件功能, 所以建议选择 jdk。



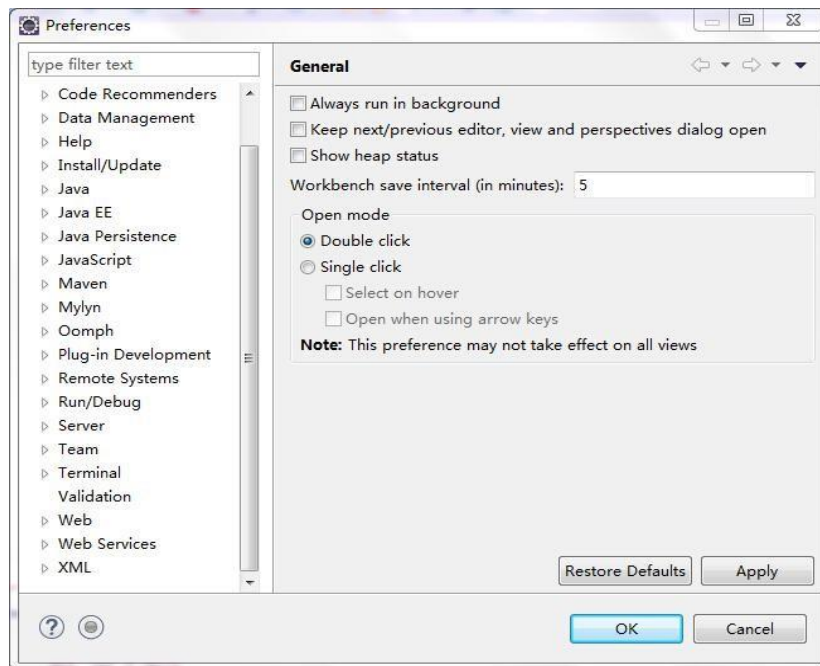
最后点击 Apply and Close 即可完成 Eclipse 开发工具中 jdk 的配置。

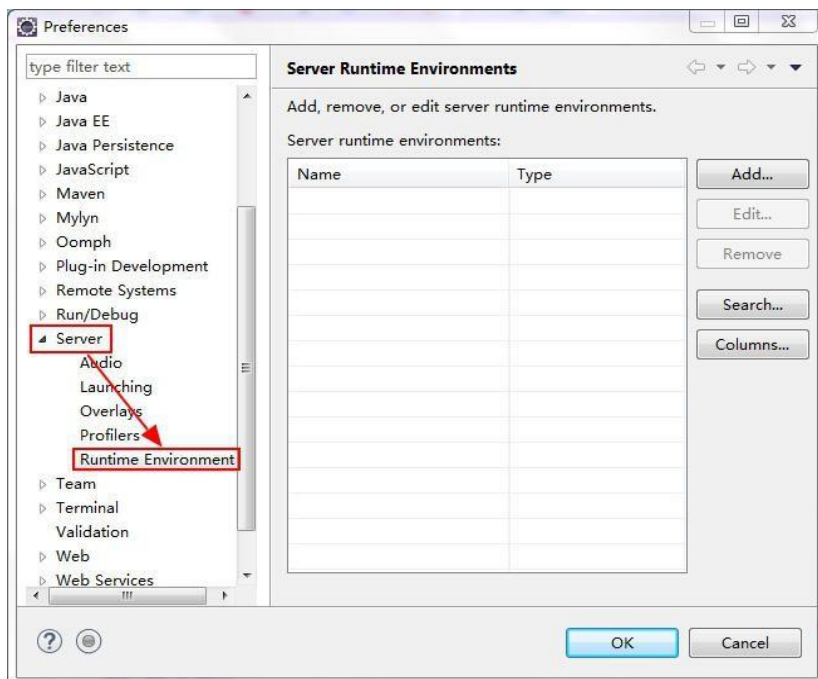
### 3.4 Eclipse 开发工具配置 tomcat

我们的程序支持所有的 java 开发环境，已经搭建好开发环境的忽略本节。其它开发工具的配置不再介绍，请自行查找相关资料。

Eclipse 开发工具配置 tomcat，并且把项目部署到 Tomcat 服务器上。

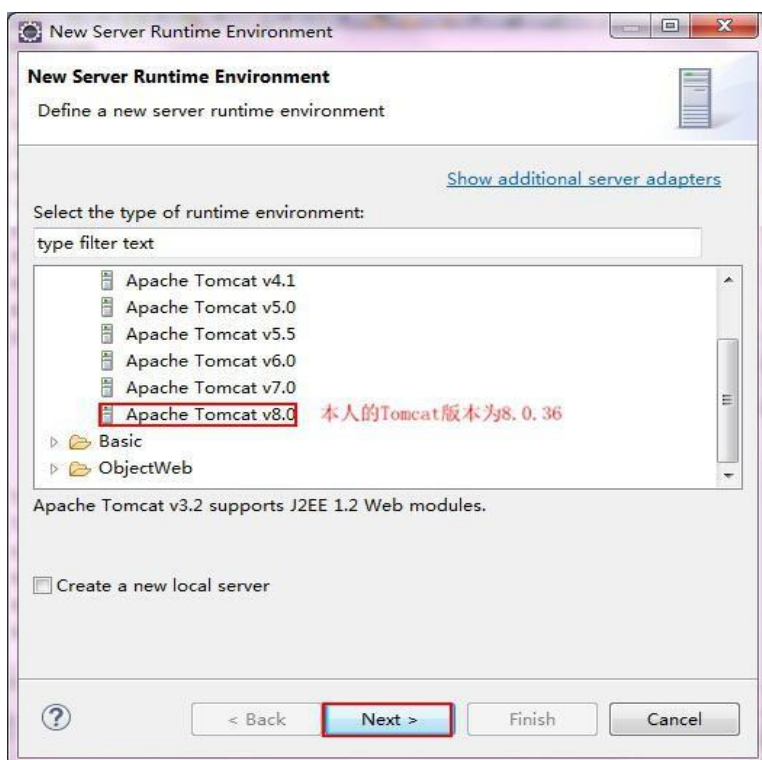
打开 Eclipse，单击“Window”菜单，选择下方的“Preferences”选项。



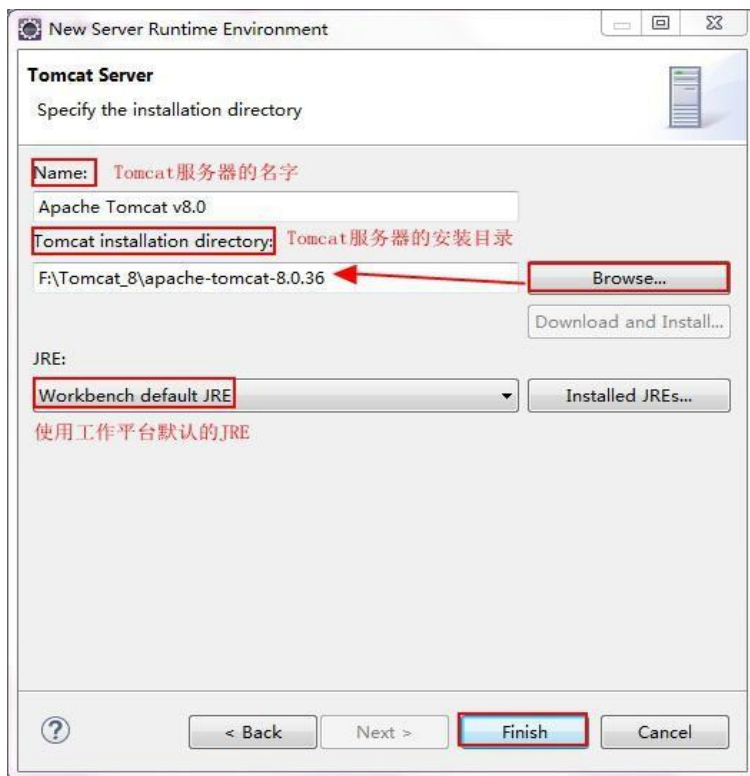


单击“Server”选项，选择下方的“Runtime Environments”。

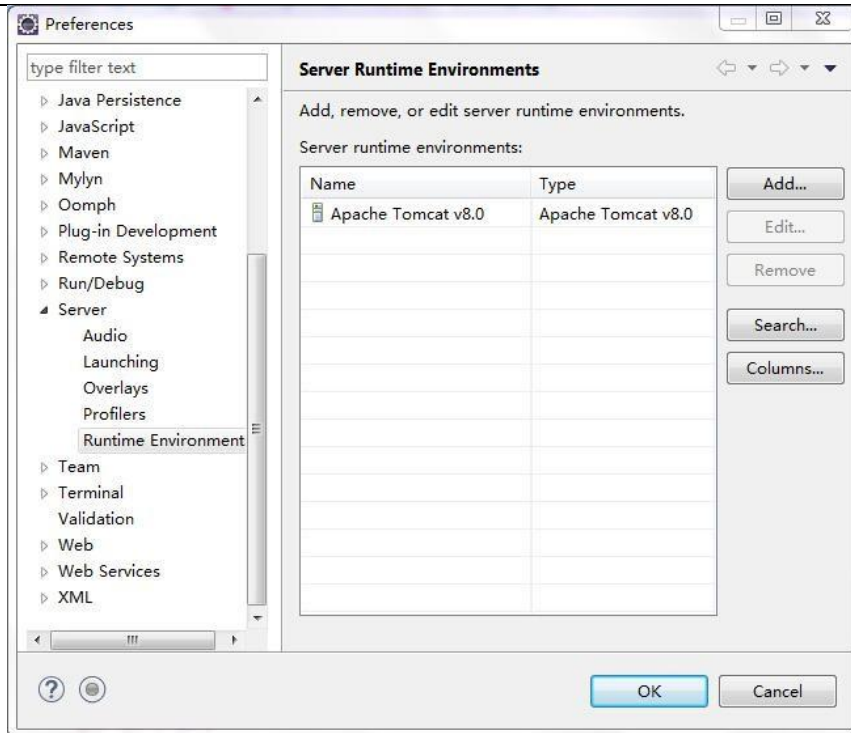
点击“Add”添加 Tomcat。



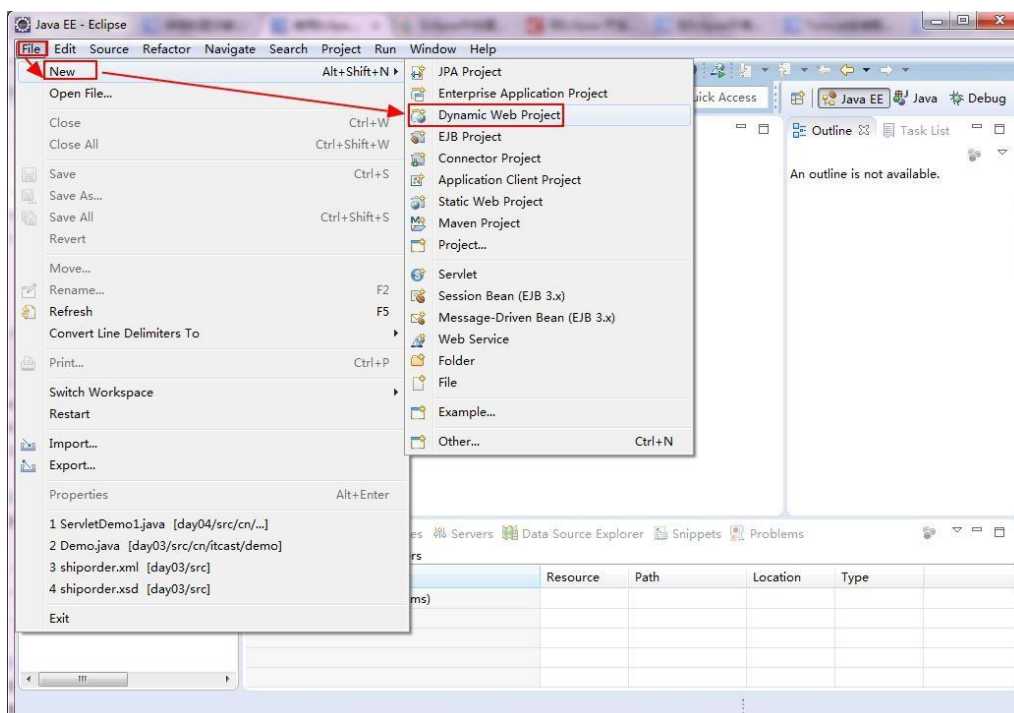
点击“Next”，选中自己安装的 Tomcat 路径。



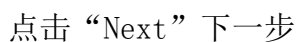
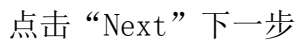
点击“Finish”完成。

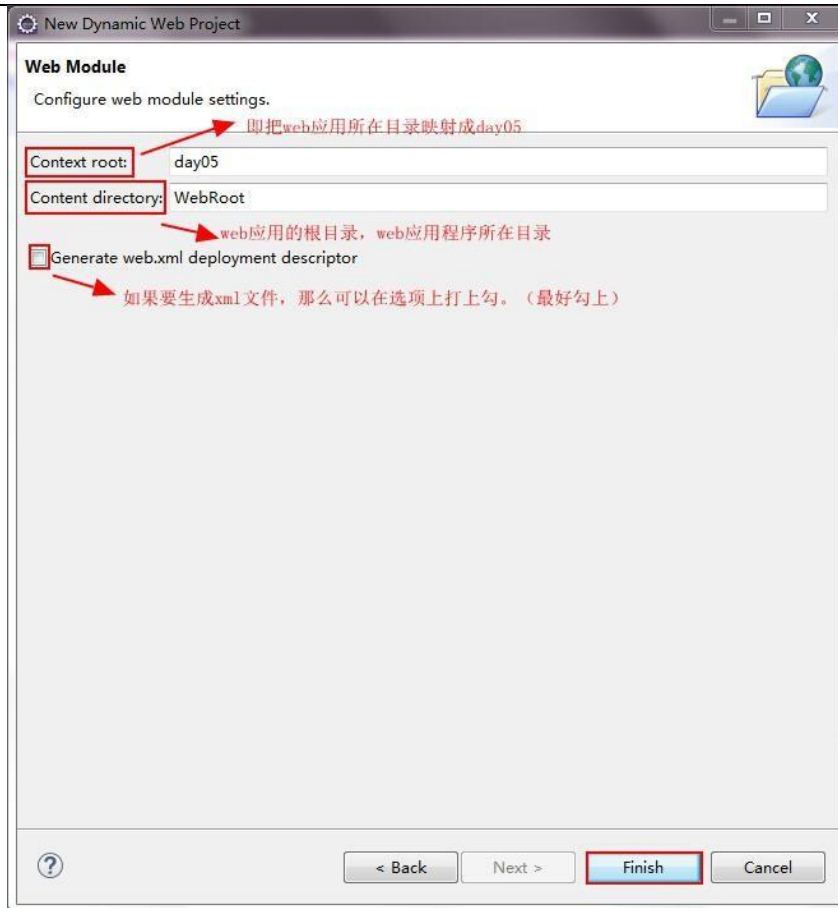


建立一个 Web 应用, File → New → Dynamic Web Project

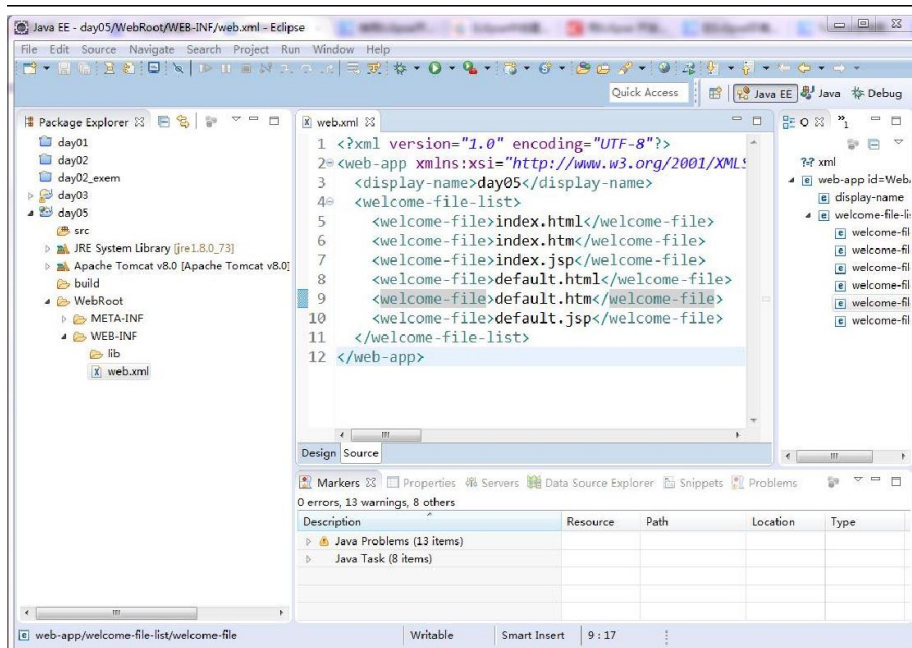


创建一个 Dynamic Web Project



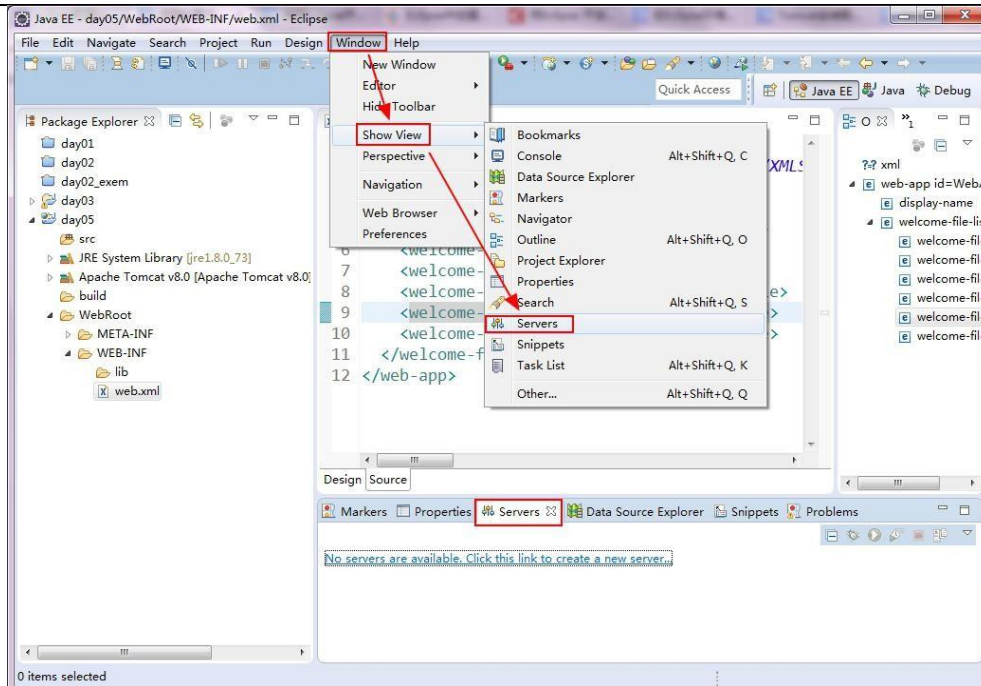


点击“Finish”完成

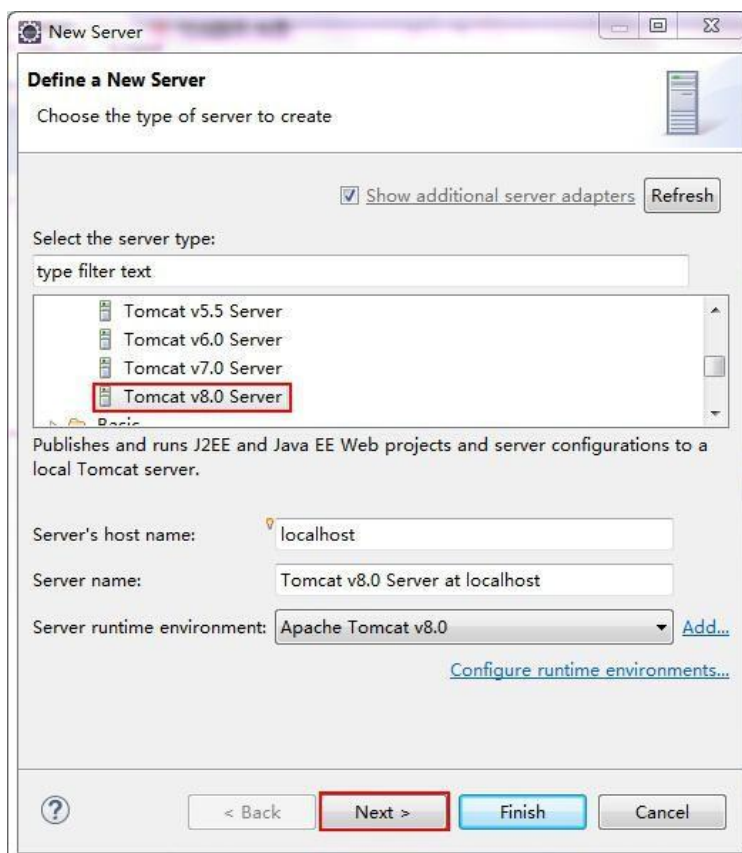


让 Tomcat 服务器显示在控制台上, 将 Web 应用部署到 Tomcat 中。

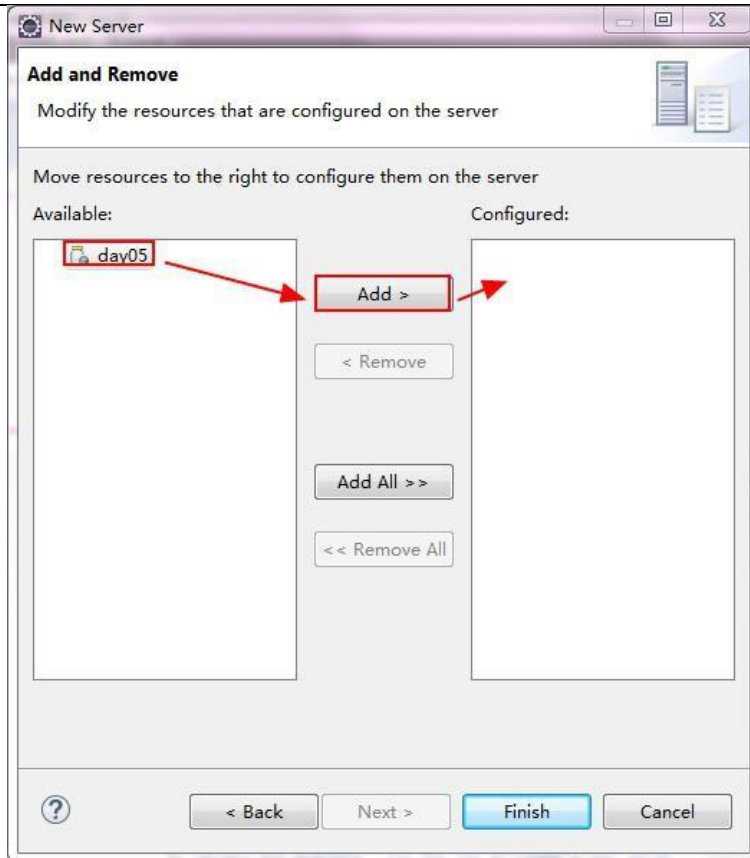
1. Window → Show View → Servers



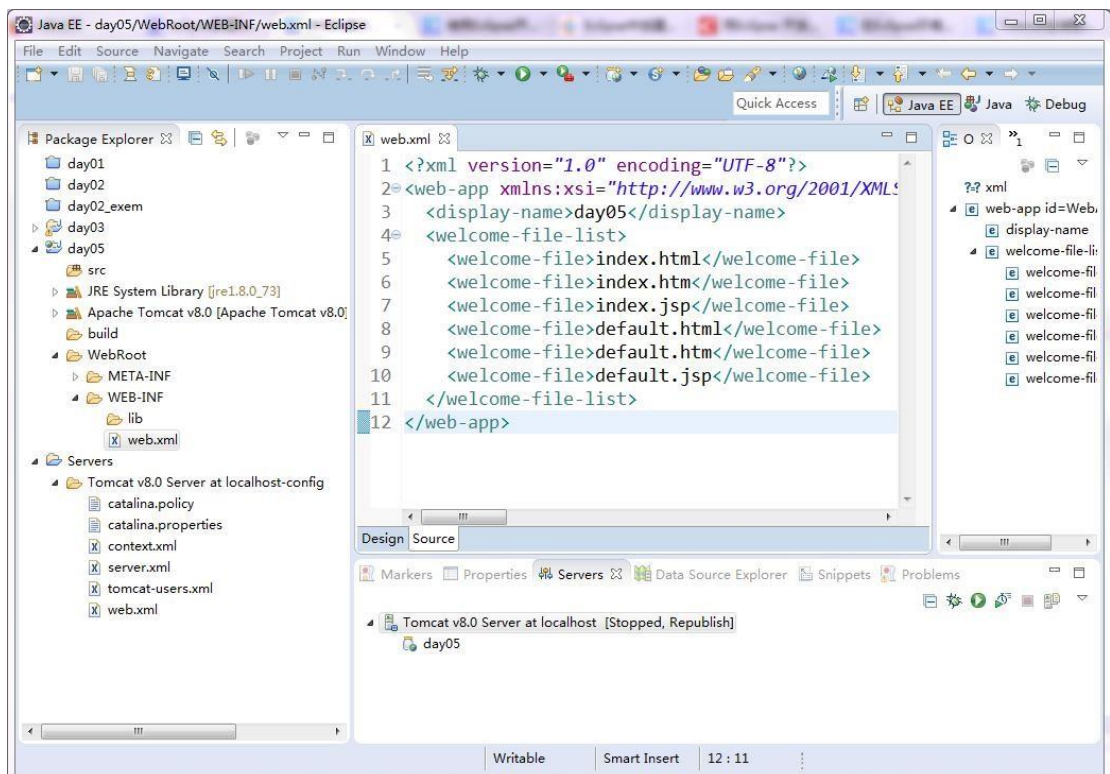
点击链接 No servers are available. Click this link to create a new server. , 在弹出的对话框中选择 Tomcat 版本。



点击“Next”，添加我们的项目。



选中项目并点击 Add，或是双击都可以添加到右边。  
点击“Finish”完成。



返回下方的“Servers”面板，右键单击该面板中的“Tomcat v8.0 Server at localhost”节点，在弹出的快捷菜单中单击“Start”，即可启动指定的 Web 服务器。如果此时直接启动访问 <http://localhost:8080/TomcatTest>，会发现会报 404 的错误。这是因为我们没有添加主页，下面添加主页(index.jsp)的内容：

```
<%@page language="java" contentType="text/html; charset=utf-8"
pageEncoding="utf-8"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

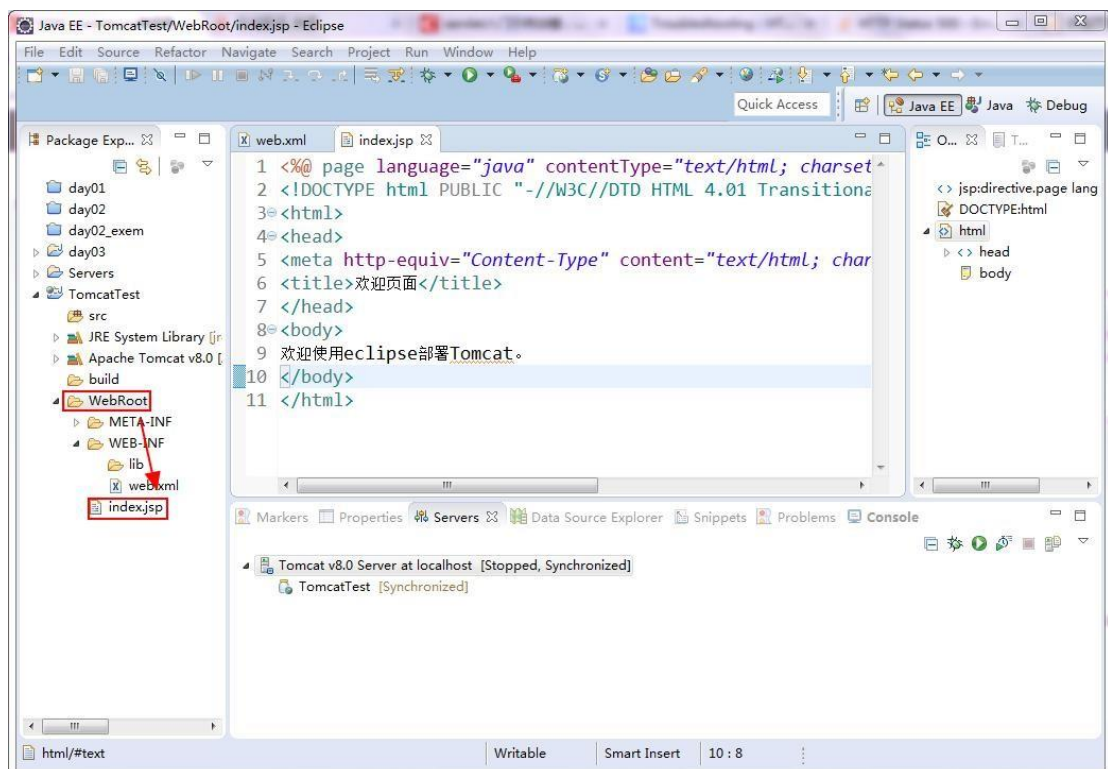
<title>欢迎页面</title>

</head>

<body>
欢迎使用 Eclipse 部署 Tomcat。

</body>

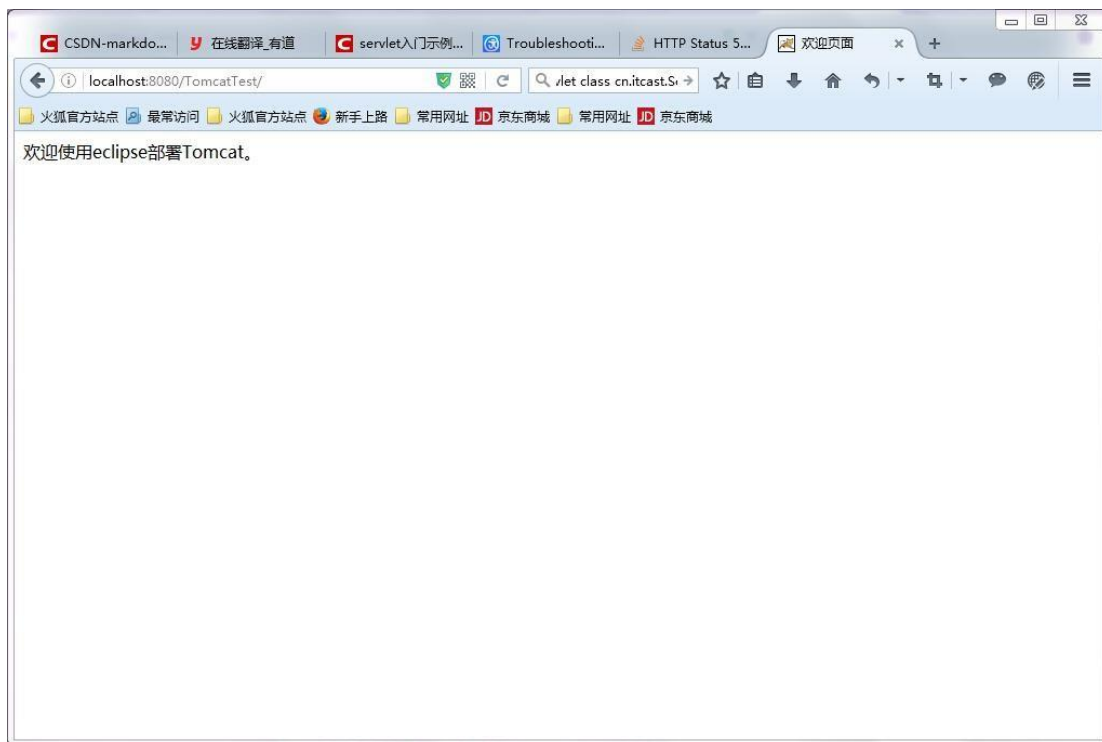
</html>
```



注意：web 资源一定要在 WebRoot 目录下添加。如图：

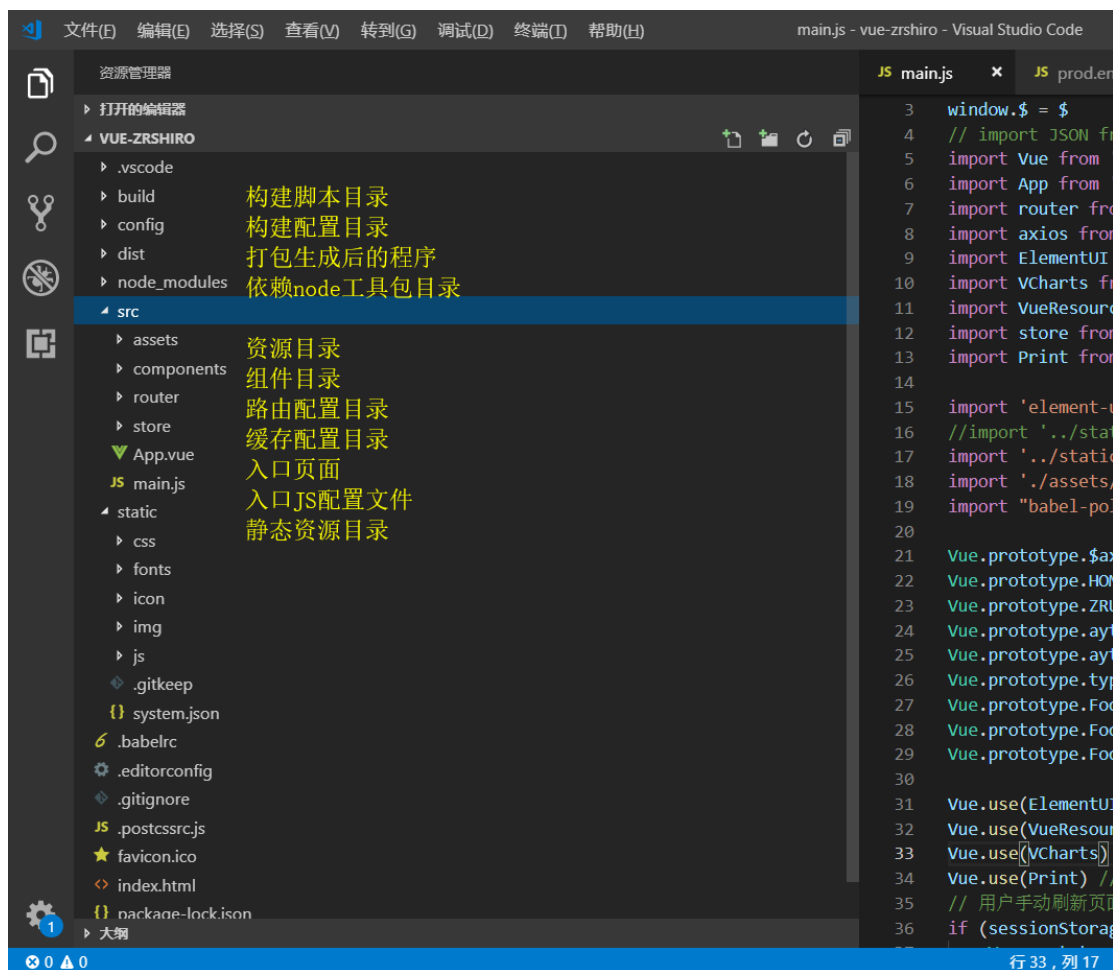
此时, 再一次来访问该链接: <http://localhost:8080/TomcatTest> , 效果如

下:



## 第四章 VUE 配置及说明

### 4.1 框架整体



### 4.2 config 目录

#### 4.2.1 dev.env.js

用于管理开发测试环境的后台接口配置。

## 4.2.2 index.js

```

4
5 const path = require('path')
6
7 module.exports = {
8   dev: {
9
10    // Paths
11    assetsSubDirectory: 'static',
12    assetsPublicPath: '/',
13    proxyTable: {
14      //数据接口地址
15      '/api': { 开发及测试环境下的api接口地址
16        target: 'http://localhost:8083/' //http://localhost:8083/
17        changeOrigin: true,
18        pathrewrite: {
19          '^/api': '/api'
20        }
21      }
22    },
23    // Various Dev Server settings
24    host: 'localhost', // can be overwritten by process.env.HOST
25    port: 8081, // can be overwritten by process.env.PORT, if port is in use, a free one will be determined
26    autoOpenBrowser: false, 开发及测试环境下的访问端口
27    errorOverlay: true,
28    notifyOnErrors: true,
29    poll: false, // https://webpack.js.org/configuration/dev-server/#devserver-watchoptions-
  }
}

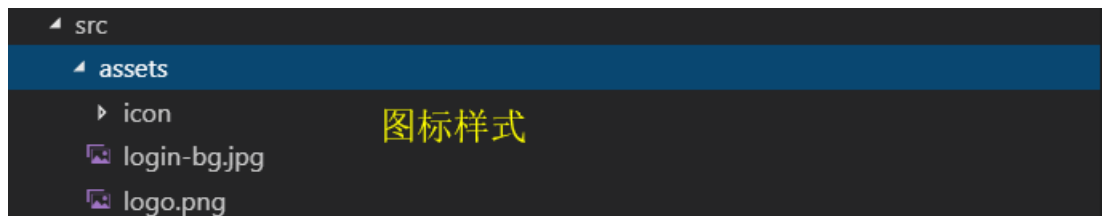
```

## 4.2.3 prod.env.js

用于管理正式环境打包的后台接口配置。

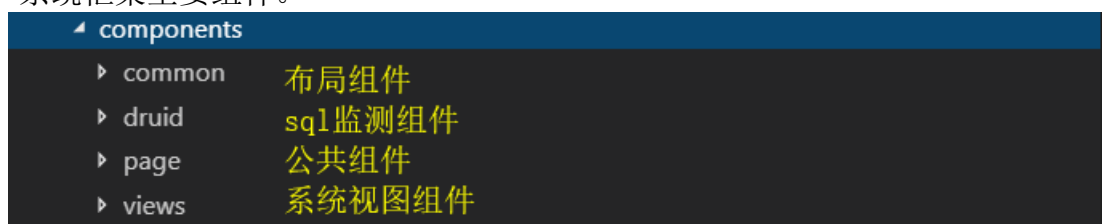
## 4.3 src 目录

### 4.3.1 assets 资源目录



### 4.3.2 components 组件目录

系统框架主要组件。



| views       |         |
|-------------|---------|
| ▸ zrcoll    | 表单引擎组件  |
| ▸ zrconfig  | 统计引擎组件  |
| ▸ zrdba     | 数据库管理组件 |
| ▸ zrflow    | 流程引擎组件  |
| ▸ zrquery   | 查询引擎组件  |
| ▸ zrsys     | 系统管理组件  |
| ▼ Login.vue | 登录组件    |

### 4.3.3 router 静态路由目录

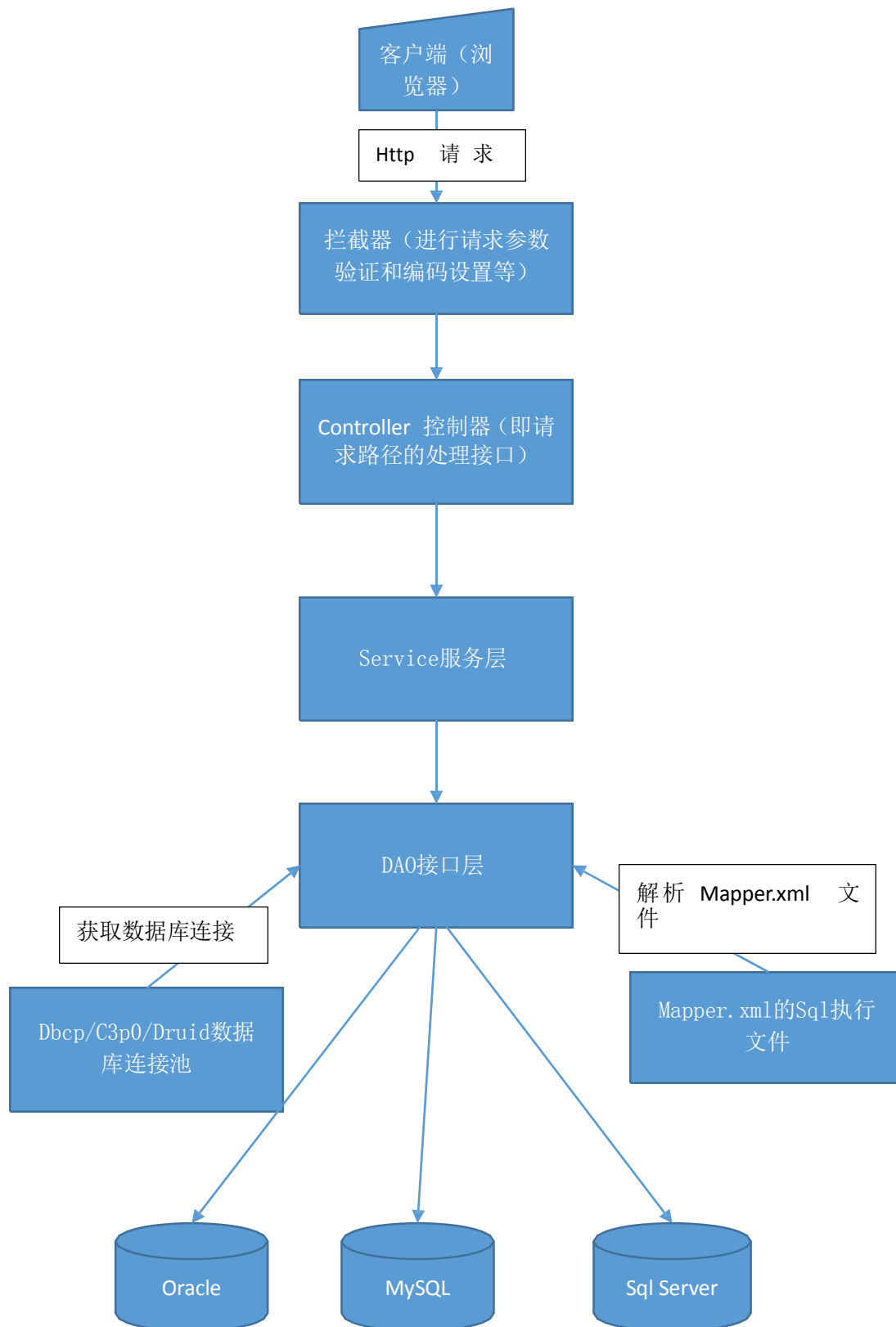
用于配置静态路由。

### 4.3.4 store 动态路由目录

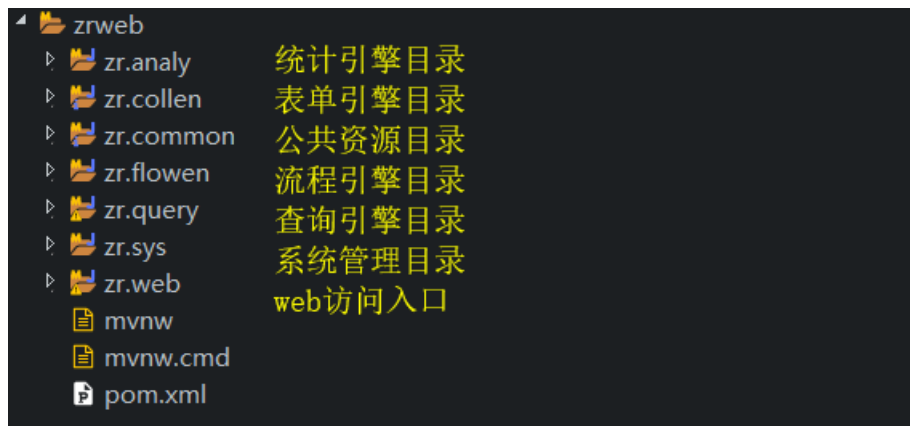
| store             |  |
|-------------------|--|
| JS addRoutes.js   |  |
| JS routersMenu.js |  |
| JS routersType.js |  |
| JS store.js       |  |
| JS viewPage.js    |  |

## 第五章 框架配置及说明

### 5.1 项目的软件架构图



## 5.2 项目结构说明



项目框架包含统计、流程、表单、查询等四大引擎及公共基础、数据库管理、系统管理等 maven 模块目录，以及 web 静态资源。

基于查询、流程、表单、统计四大引擎项目框架包含以下：



**src/main/java** : 包含了 controller 、 mapper 、 model\entiy 、 service\serviceImp 等控制、映射、模型\实体、接口等标准目录。

**src/main/resources** : 包含了 mapper、template、lib 等目录，分别存放 mybatis 数据库映射文件 xml、静态文件、jar 包等。

**src/test/java** : maven 模块下的测试代码。

**pom.xml** : 项目下的子模块（module）的 maven 版本控制，相对独立。

## 5.3 项目的核心技术说明

1、项目的 Maven 配置，pom.xml 文件中，本项目采用 maven 模块开发，用多

个 pom.xml 文件，主要分为 maven project、maven module，主要解释如下：

### (1) 主项目 maven 的 pom.xml:

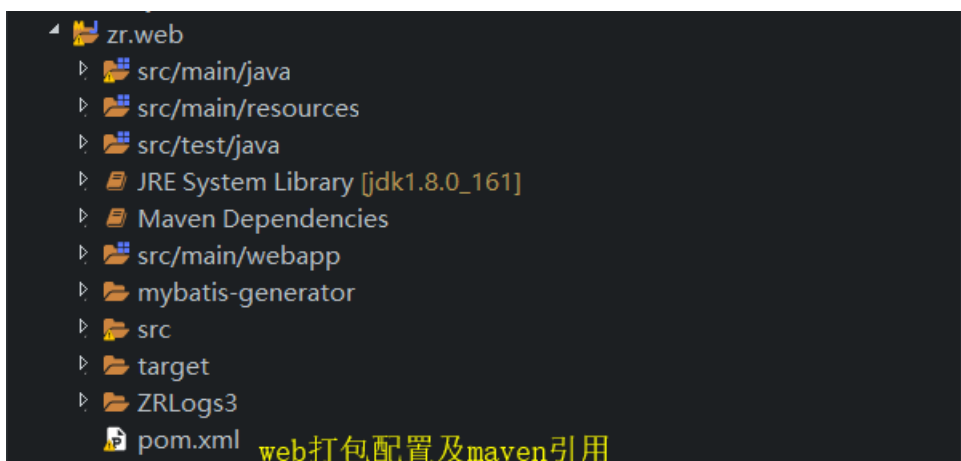


```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="
3     <modelVersion>4.0.0</modelVersion>
4     <groupId>zr.zrpower</groupId>
5     <artifactId>ZRSpringBootThymfEsysui</artifactId>
6     <version>0.0.1-SNAPSHOT</version>
7     <packaging>pom</packaging>
8     <name>ZRSpringBootThymfEsysui</name>
9
10    <!--maven module WEB打包-->
11    <modules>
12        <module>zr.web</module>
13        <module>zr.common</module>
14        <module>zr.query</module>
15        <module>zr.sys</module>
16        <module>zr.collen</module>
17        <module>zr.analy</module>
18        <module>zr.flowen</module>
19        <module>zr.dbEngine</module>
20    </modules>
21
  
```

项目公共maven版本控制、公共jar引用、Spring Boot配置, module模块引用

### (2) 入口 maven 的 pom.xml:



web打包配置及maven引用

|    |                                        |  |
|----|----------------------------------------|--|
| 59 | <dependencies>                         |  |
| 60 | <!-- 加载模块 -->                          |  |
| 61 | <dependency>                           |  |
| 62 | <groupId>zr.zrpower</groupId>          |  |
| 63 | <artifactId>zr.dbEngine</artifactId>   |  |
| 64 | <version>\${project.version}</version> |  |
| 65 | </dependency>                          |  |
| 66 | <dependency>                           |  |
| 67 | <groupId>zr.zrpower</groupId>          |  |
| 68 | <artifactId>zr.sys</artifactId>        |  |
| 69 | <version>\${project.version}</version> |  |
| 70 | </dependency>                          |  |
| 71 | <dependency>                           |  |
| 72 | <groupId>zr.zrpower</groupId>          |  |
| 73 | <artifactId>zr.query</artifactId>      |  |
| 74 | <version>\${project.version}</version> |  |
| 75 | </dependency>                          |  |
| 76 | <dependency>                           |  |
| 77 | <groupId>zr.zrpower</groupId>          |  |
| 78 | <artifactId>zr.common</artifactId>     |  |
| 79 | <version>\${project.version}</version> |  |
| 80 | </dependency>                          |  |
| 81 | <dependency>                           |  |
| 82 | <groupId>zr.zrpower</groupId>          |  |
| 83 | <artifactId>zr.colln</artifactId>      |  |
| 84 | <version>\${project.version}</version> |  |
| 85 | </dependency>                          |  |
| 86 | <dependency>                           |  |
| 87 | <groupId>zr.zrpower</groupId>          |  |
| 88 | <artifactId>zr.analy</artifactId>      |  |
| 89 | <version>\${project.version}</version> |  |
| 90 | </dependency>                          |  |
| 91 | <dependency>                           |  |
| 92 | <groupId>zr.zrpower</groupId>          |  |
| 93 | <artifactId>zr.flown</artifactId>      |  |
| 94 | <version>\${project.version}</version> |  |

加载web项目需要引用的子模块

### (3) 其它 maven 的 pom.xml:

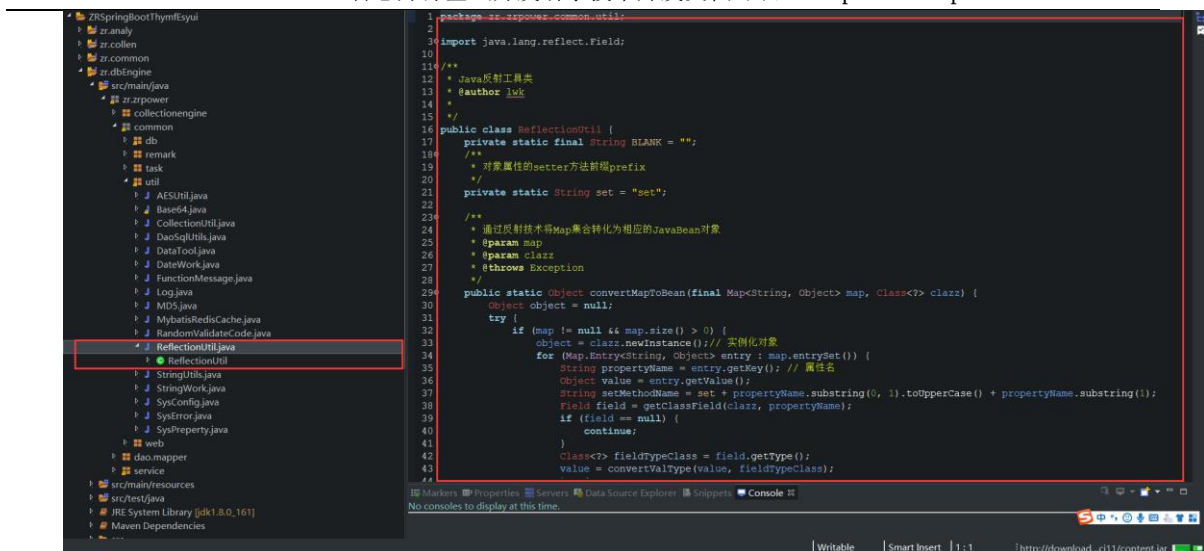
|    |                                                                    |  |
|----|--------------------------------------------------------------------|--|
| 7  | <parent>                                                           |  |
| 8  | <groupId>zr.zrpower</groupId>                                      |  |
| 9  | <artifactId>ZRSpringBootThymfEsyui</artifactId>                    |  |
| 10 | <version>0.0.1-SNAPSHOT</version>                                  |  |
| 11 | </parent>                                                          |  |
| 12 |                                                                    |  |
| 13 | <artifactId>zr.query</artifactId>                                  |  |
| 14 | <packaging>jar</packaging>                                         |  |
| 15 |                                                                    |  |
| 16 | <properties>                                                       |  |
| 17 | <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding> |  |
| 18 | </properties>                                                      |  |
| 19 |                                                                    |  |
| 20 | <dependencies>                                                     |  |
| 21 | <!-- 引用工具的模块 -->                                                   |  |
| 22 | <dependency>                                                       |  |
| 23 | <groupId>zr.zrpower</groupId>                                      |  |
| 24 | <artifactId>zr.common</artifactId>                                 |  |
| 25 | <version>\${project.version}</version>                             |  |
| 26 | </dependency>                                                      |  |
| 27 | <dependency>                                                       |  |
| 28 | <groupId>zr.zrpower</groupId>                                      |  |
| 29 | <artifactId>zr.sys</artifactId>                                    |  |
| 30 | <version>\${project.version}</version>                             |  |
| 31 | </dependency>                                                      |  |
| 32 |                                                                    |  |
| 33 | </dependencies>                                                    |  |

定义父亲maven项目的名称、版本

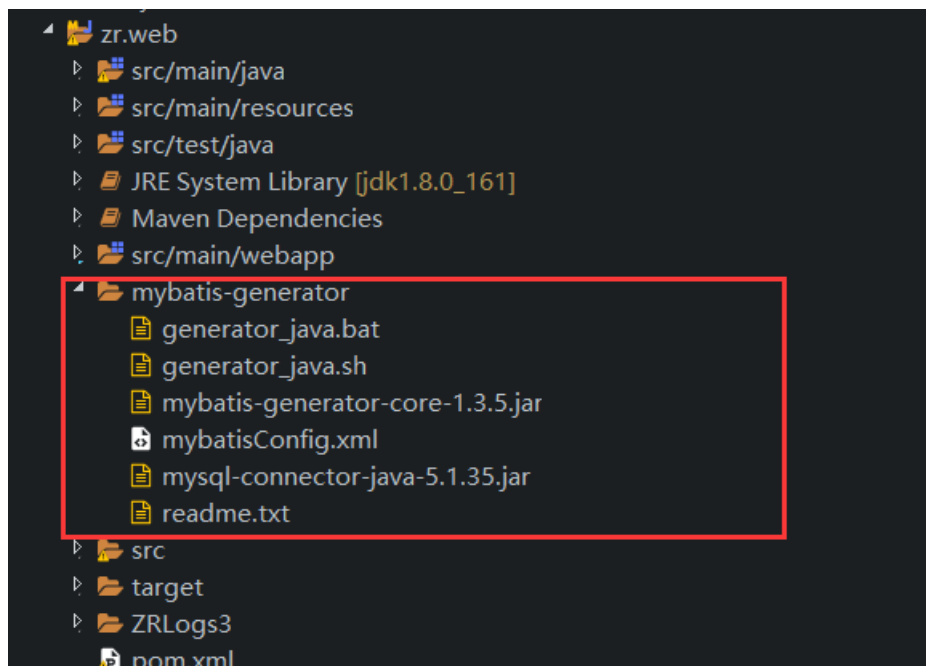
定义本maven模块的标识符、打包类型为jar

本maven模块是否引用框架内其它的maven项目及版本号。

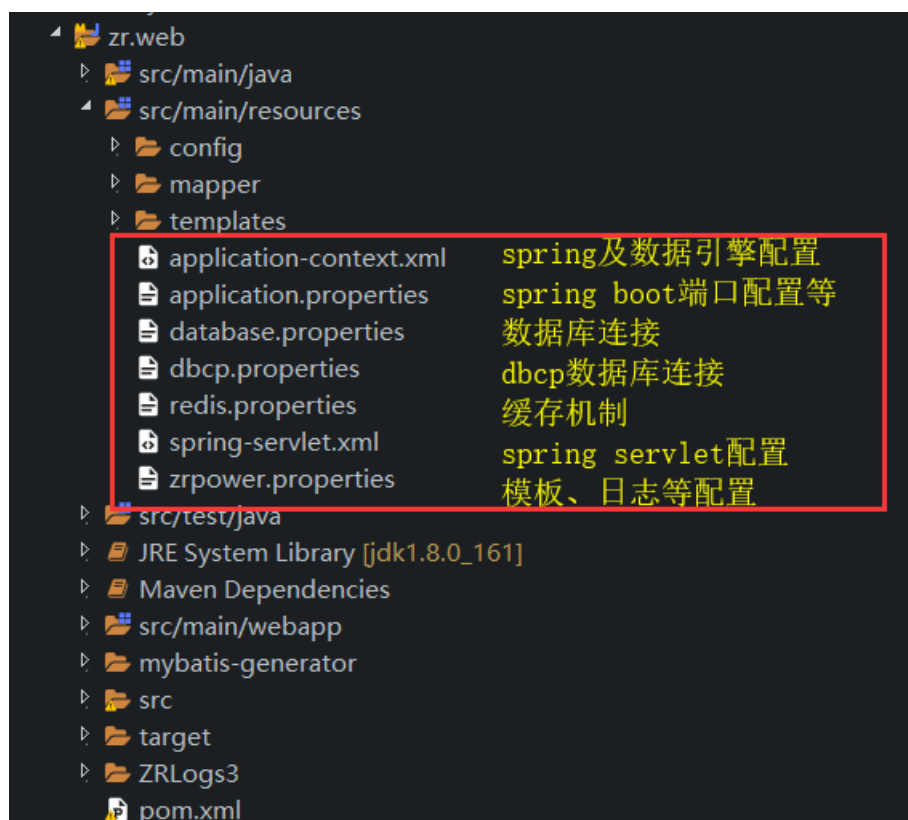
2、项目中的 Java 反射工具类，用于解析 Map 和 JavaBean 之间的转换，如图：



### 3、Mybatis 动态生成代码的配置，如图



### 4、项目中的数据库连接池配置，如图：



druid 数据库连接池:

```

69 <!-- 数据源: druid数据库连接池 -->
70 <!-- 参数说明:
71 filters: 配置监控统计拦截的filters
72 initialSize: 初始化时建立物理连接的个数。初始化发生在显示调用init方法, 或者第一次getConnection时
73 maxActive: 最大连接池数量。Default: 8
74 minIdle: 最小连接池数量
75 maxWait: 获取连接时最大等待时间, 单位毫秒
76 timeBetweenEvictionRunsMillis: 配置间隔多久才进行一次检测, 检测需要关闭的空闲连接, 单位是毫秒
77 minEvictableIdleTimeMillis: 配置一个连接在池中最小生存的时间, 单位是毫秒
78 poolPreparedStatements: 是否缓存preparedStatement, 也就是PSCache。
79                                PSCache对支持游标的数据库性能提升巨大, 比如说oracle。在MySQL下建议关闭
80 maxOpenPreparedStatements: 要启用PSCache, 必须配置大于0, 当大于0时, poolPreparedStatements自动触发修改为true
81 -->
82 <bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init" destroy-method="close">
83   <property name="driverClassName" value="${db.driver}" />
84   <property name="url" value="${db.url}" />
85   <property name="username" value="${db.username}" />
86   <property name="password" value="${db.password}" />
87   <property name="filters" value="stat" />
88   <property name="initialSize" value="${db.initialSize}" />
89   <property name="maxActive" value="${db.maxActive}" />
90   <property name="minIdle" value="${db.minIdle}" />
91   <property name="maxWait" value="${db.maxWait}" />
92   <!-- 这里配置提交方式, 默认就是TRUE, 可以不用配置 -->
93   <property name="defaultAutoCommit" value="true" />
94   <property name="timeBetweenEvictionRunsMillis" value="${db.timeBetweenEvictionRunsMillis}" />
95   <property name="minEvictableIdleTimeMillis" value="${db.minEvictableIdleTimeMillis}" />
96   <property name="testWhileIdle" value="true" />
97   <property name="testOnBorrow" value="false" />
98   <property name="testOnReturn" value="false" />
99   <property name="poolPreparedStatements" value="true" />
100  <property name="maxOpenPreparedStatements" value="50" />
101 </bean>

```

c3p0 数据库连接池:

```

42 <!-- 数据源 : c3p0数据库连接池 -->
43 <!-- 参数说明:
44 initialPoolSize: 初始化时获取的连接数, 取值应在minPoolSize与maxPoolSize之间. Default: 3
45 minPoolSize: 连接池中保留的最小连接数. Default: 3
46 maxPoolSize: 连接池中保留的最大连接数. Default: 15
47 maxIdleTime: 最大空闲时间, 60秒内未使用则连接被丢弃. 若为0则永不丢弃. Default: 0
48 acquireIncrement: 当连接池中的连接耗尽的时候c3p0一次同时获取的连接数. Default: 3
49 maxStatements: JDBC的标准参数, 用以控制数据源内加载的PreparedStatements数量. 但由于预缓存的statements
50 属于单个connection而不是整个连接池. 所以设置这个参数需要考虑到多方面的因素.
51 如果maxStatements与maxStatementsPerConnection均为0, 则缓存被关闭. Default: 0
52 idleConnectionTestPeriod: 每60秒检查所有连接池中的空闲连接. Default: 0
53 acquireRetryAttempts: 定义在从数据库获取新连接失败后重复尝试的次数. Default: 30
54 -->
55 <!-- <bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource" destroy-method="close">
56 <property name="driverClass" value="${db.driver}" />
57 <property name="jdbcUrl" value="${db.url}" />
58 <property name="user" value="${db.username}" />
59 <property name="password" value="${db.password}" />
60 <property name="initialPoolSize" value="${db.initialSize}" />
61 <property name="minPoolSize" value="${db.minIdle}" />
62 <property name="maxPoolSize" value="${db.maxIdle}" />
63 <property name="acquireIncrement" value="${db.acquireIncrement}" />
64 <property name="maxStatements" value="${db.maxStatements}" />
65 <property name="idleConnectionTestPeriod" value="${db.idleConnectionTestPeriod}" />
66 <property name="acquireRetryAttempts" value="${db.acquireRetryAttempts}" />
67 </bean> -->

```

dbcp 数据库连接池:

```

22 <!-- 数据源 : dbcp数据库连接池 -->
23 <!-- 参数说明:
24 initialSize: 连接初始值, 连接池启动时创建的连接数量的初始值, 默认值是0
25 minIdle: 最小空闲值, 当空闲的连接数少于阈值时, 连接池就会预申请去一些连接, 以免洪峰来时来不及申请, 默认值是0
26 maxIdle: 最大空闲值, 当经过一个高峰时间后, 连接池将已经用不到的连接释放一部分, 一直减少到maxIdle为止, 0时无限制, 默认值是8
27 defaultAutoCommit: 默认的SQL语句自动提交状态(开启或关闭) 设置由连接池本身设置 (false由连接池定),
28 不设置该值setAutoCommit方法不被调用
29 -->
30 <!-- <bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
31 <property name="driverClassName" value="${db.driver}" />
32 <property name="url" value="${db.url}" />
33 <property name="username" value="${db.username}" />
34 <property name="password" value="${db.password}" />
35 <property name="initialSize" value="${db.initialSize}" />
36 <property name="minIdle" value="${db.minIdle}" />
37 <property name="maxIdle" value="${db.maxIdle}" />
38 <property name="defaultAutoCommit" value="${db.defaultAutoCommit}" />
39 </bean> -->

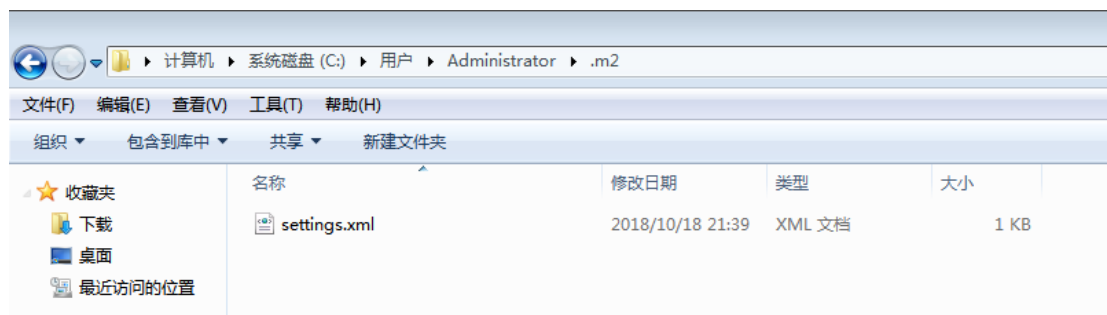
```

5、项目中 Java 使用 dom4j 动态生成 mybatisConfig.xml 文件。

6、项目中 Java 动态生成 generator\_java.bat 脚本文件及调用。

## 5.4 Eclipse 开发工具的 Maven 配置

在 C:\Users\Administrator\目录下创建.m2 文件夹, 在.m2 文件夹下创建 settings.xml 的文件, 如图所示:



在 settings.xml 的文件中配置本地 Maven 仓库地址及阿里云代理, 文件内容如下:

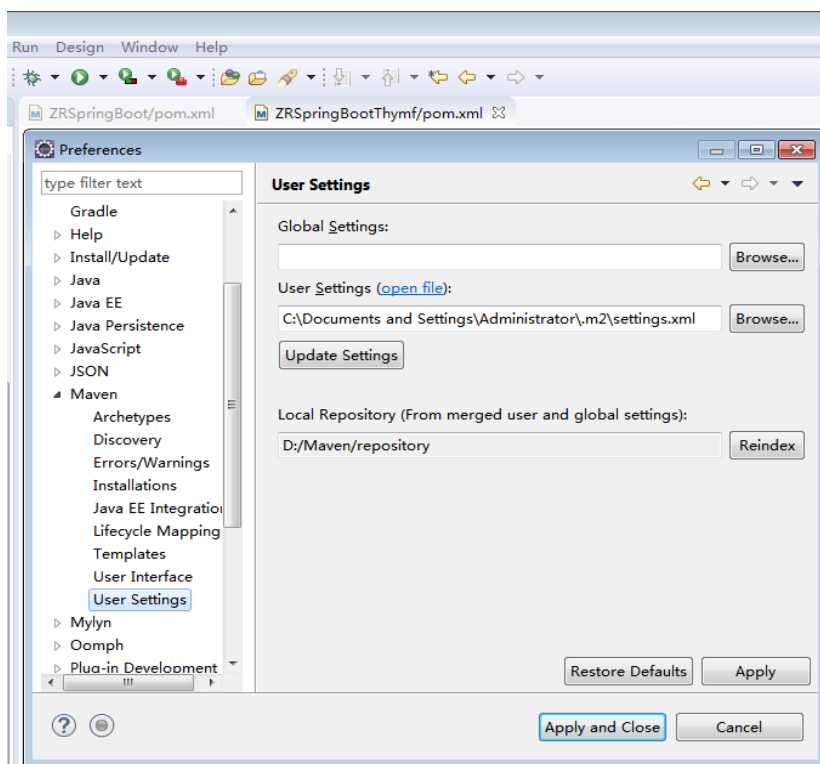
```
1<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2      xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://maven.apache.org/xsd/settings-1.0.0.xsd">
3      <!-- 指定Maven本地仓库的位置 -->
4      <localRepository>D:/repos</localRepository>
5      <!--
6      <localRepository>${user.home}/.m2/repository</localRepository>
7      -->
8      <interactiveMode>true</interactiveMode>
9      <usePluginRegistry>false</usePluginRegistry>
10     <offline>false</offline>
11
12     <mirrors>
13     <mirror>
14         <id>nexus-aliyun</id>
15         <mirrorOf>*</mirrorOf>
16         <name>Nexus aliyun</name>
17         <url>http://maven.aliyun.com/nexus/content/groups/public</url>
18     </mirror>
19     </mirrors>
20 </settings>
21
```

具体代码如下:

```
<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.org/xsd/settings-1.0.0.xsd">
  <!-- 指定Maven本地仓库的位置 -->
  <localRepository>D:/repos</localRepository>
  <!--
  <localRepository>${user.home}/.m2/repository</localRepository>
  -->
  <interactiveMode>true</interactiveMode>
  <usePluginRegistry>false</usePluginRegistry>
  <offline>false</offline>

  <mirrors>
  <mirror>
    <id>nexus-aliyun</id>
    <mirrorOf>*</mirrorOf>
    <name>Nexus aliyun</name>
    <url>http://maven.aliyun.com/nexus/content/groups/public</url>
  </mirror>
  </mirrors>
</settings>
```

打开 Eclipse 开发工具, 点击菜单 Window→Preferences→Maven→User Settings, 配置 User Settings(浏览找到 settings.xml 文件), 点击 Update Settings 按钮, 如图所示:

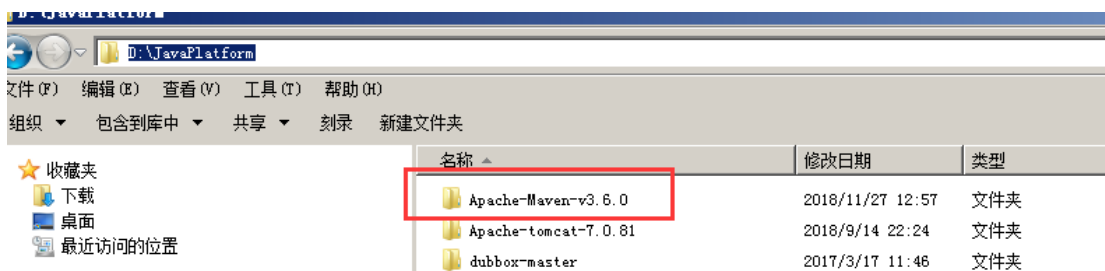


点击 Reindex 按钮, 然后点击 Apply and Close 按钮即可 Eclipse 开发工具的 Maven 配置。

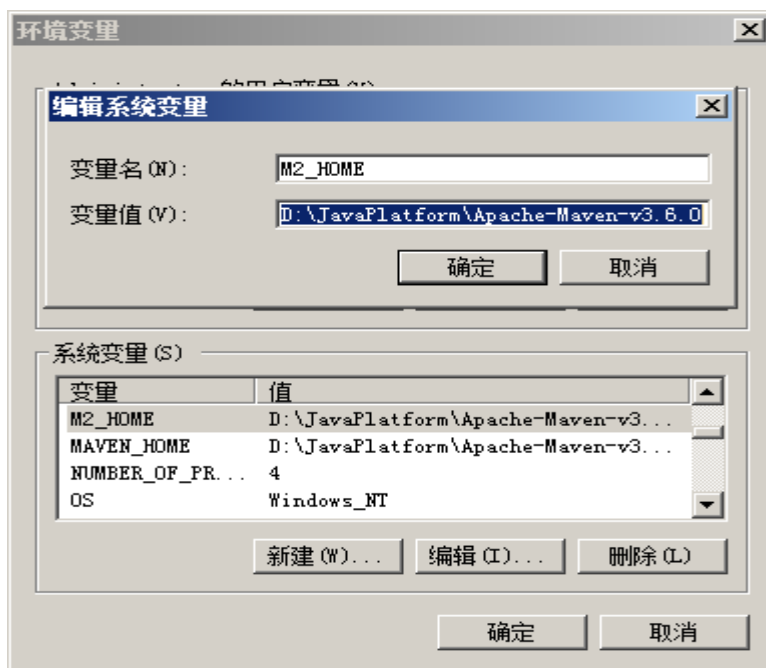
## 5.5 打包部署操作说明

一、在本地上安装 Maven 环境

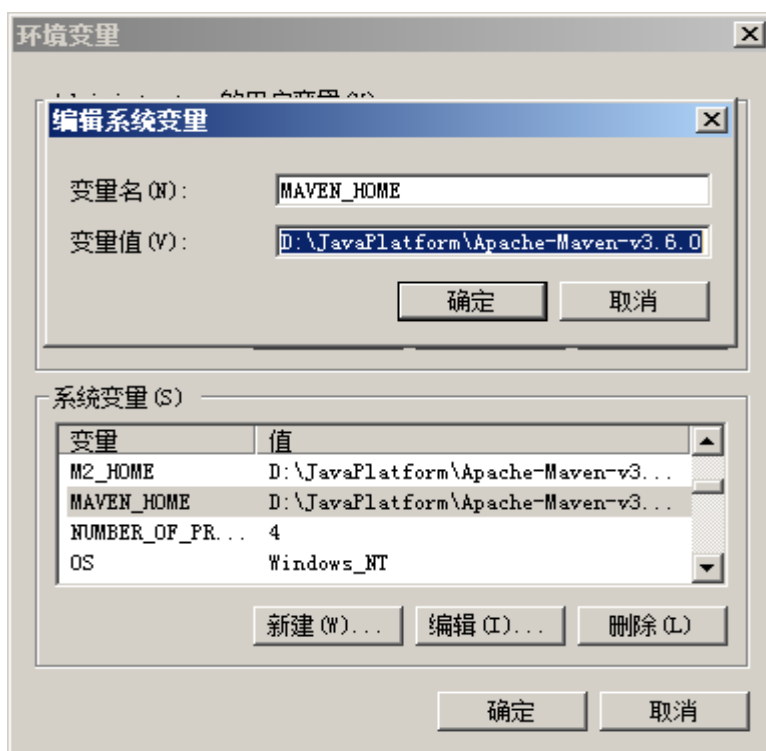
1. Maven 下载地址 <http://maven.apache.org/download.cgi>
2. 将下载的文件 (apache-maven-3.6.0-bin.zip) 放在 D:\JavaPlatform 目录下并解压, 如图



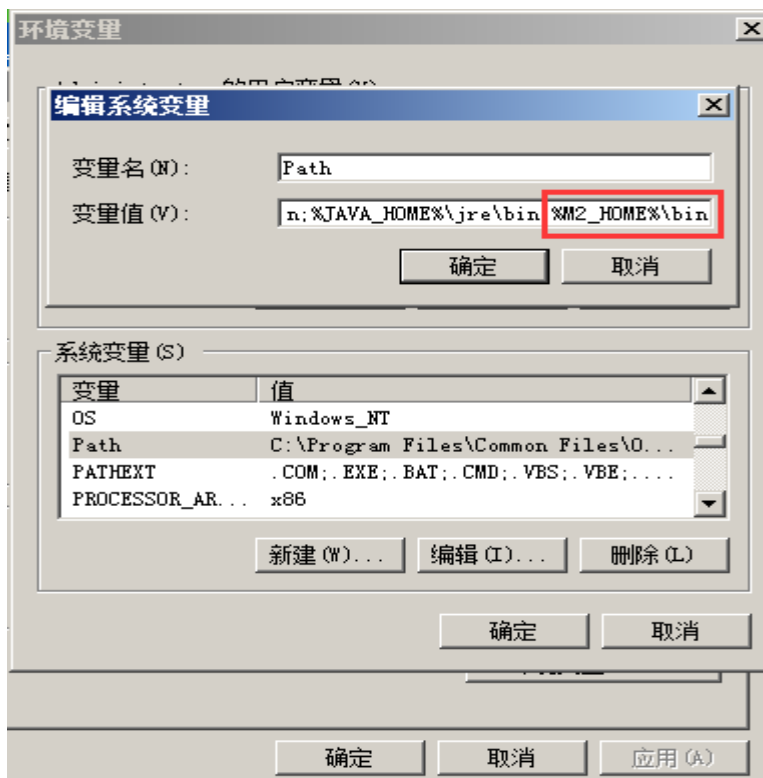
3. 配置 Maven 环境:



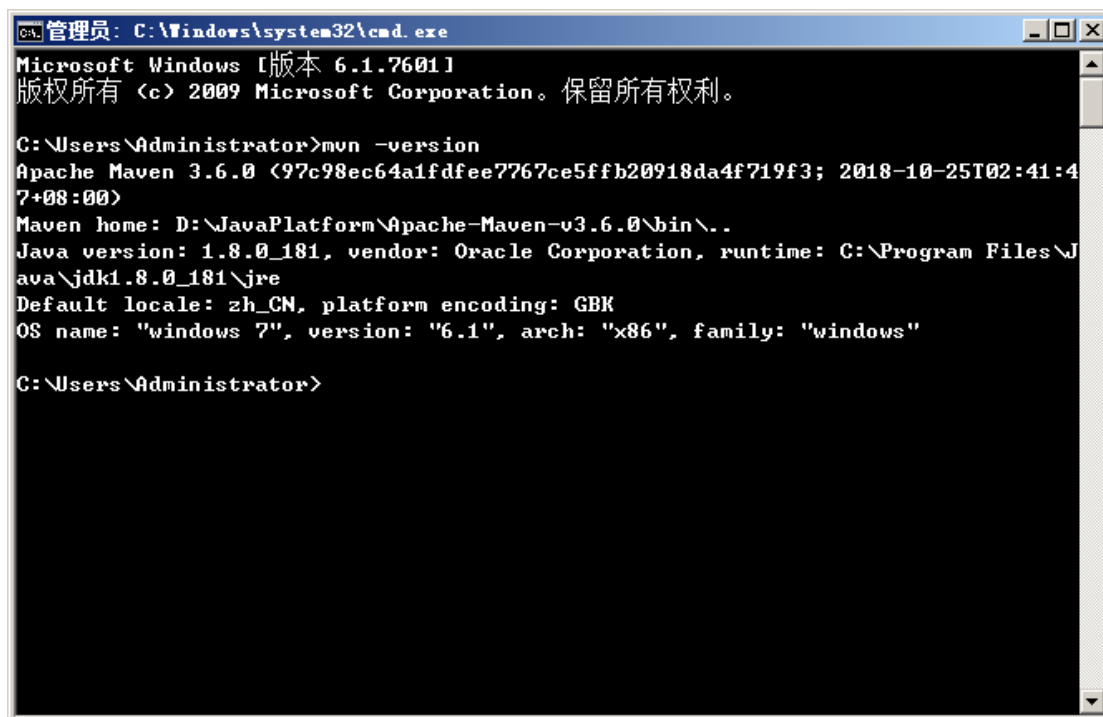
新增 M2\_HOME 环境变量



新增 MAVEN\_HOME 环境变量



在 Path 环境变量中新增 Maven 配置，点击确定即可。%M2\_HOME%\bin

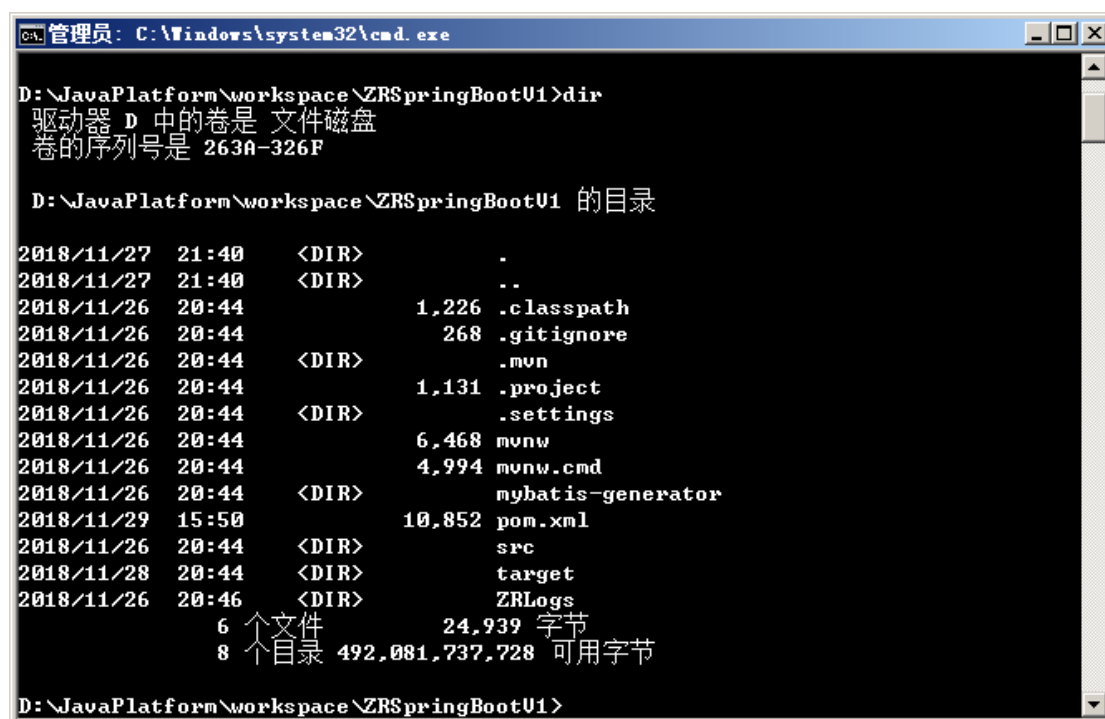


查看 maven 是否安装成功，输入 mvn -version，出现上图说明安装配置完成。

## 二、将 ZRSpringBoot 项目中本地引用的 jar 包发布到本地 Maven 仓库

说明: 理论上可以在 ZRSpringBoot 的 pom.xml 文件中设置将本地引用 jar 包打包到项目 jar 包 zrweb-0.0.1-SNAPSHOT.jar 中, 但是实际上会有很多问题出现, 不建议使用。

进入 ZRSpringBoot 项目所在的目录, 如图



```
管理员: C:\Windows\system32\cmd.exe

D:\JavaPlatform\workspace\ZRSpringBoot01>dir
驱动器 D 中的卷是 文件磁盘
卷的序列号是 263A-326F

D:\JavaPlatform\workspace\ZRSpringBoot01 的目录

2018/11/27  21:40    <DIR>          .
2018/11/27  21:40    <DIR>          ..
2018/11/26  20:44             1,226 .classpath
2018/11/26  20:44             268 .gitignore
2018/11/26  20:44    <DIR>          .mvn
2018/11/26  20:44             1,131 .project
2018/11/26  20:44    <DIR>          .settings
2018/11/26  20:44             6,468 mvnw
2018/11/26  20:44             4,994 mvnw.cmd
2018/11/26  20:44    <DIR>          mybatis-generator
2018/11/29  15:50             10,852 pom.xml
2018/11/26  20:44    <DIR>          src
2018/11/28  20:44    <DIR>          target
2018/11/26  20:46    <DIR>          ZRLogs
               6 个文件             24,939 字节
               8 个目录 492,081,737,728 可用字节

D:\JavaPlatform\workspace\ZRSpringBoot01>
```

将所有的本地引用 jar 包发布到本地 Maven 仓库中去, 命令如下:

如将 xssProtect-0.1.jar 发布到本地 Maven 仓库中去, 输入命令执行

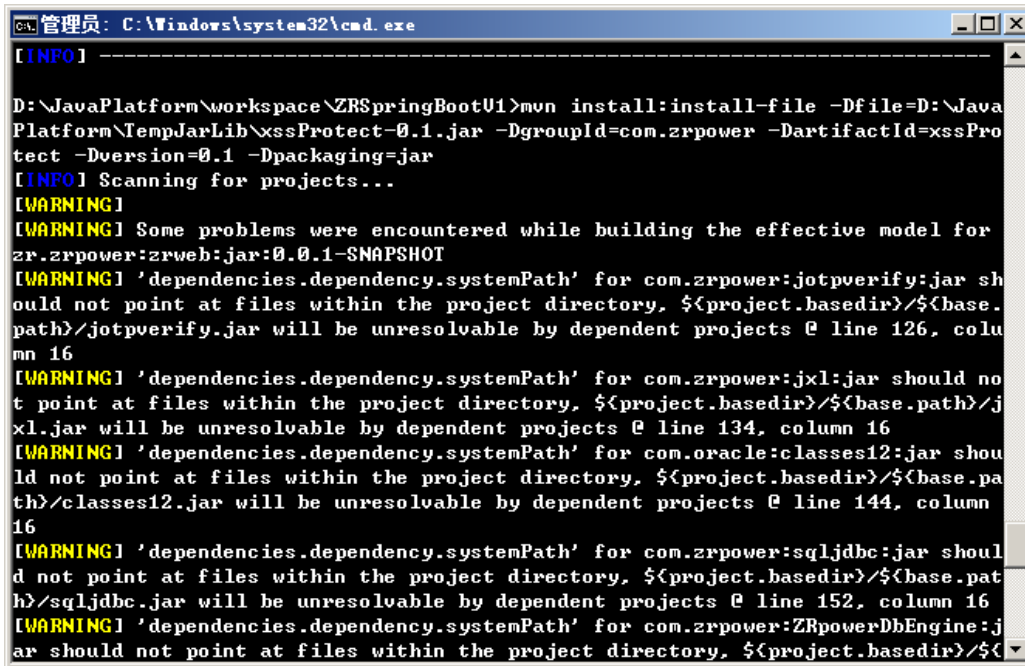
```
mvn install:install-file -Dfile=D:\JavaPlatform\TempJarLib\xssProtect-0.1.jar
-DgroupId=com.zrpower -DartifactId=xssProtect -Dversion=0.1 -Dpackaging=jar
```

```
<dependency>
  <groupId>com.zrpower</groupId>
  <artifactId>xssProtect</artifactId>
  <version>0.1</version>
<!--
```

命令参数说明:

- Dfile 为 jar 文件所在目录
- DgroupId=com.zrpower 为 jar 包的组
- DartifactId=xssProtect 为 jar 包的 ID

- Dversion=0.1 为 jar 包的版本号
- Dpackaging=jar 为发布类型为 jar 包

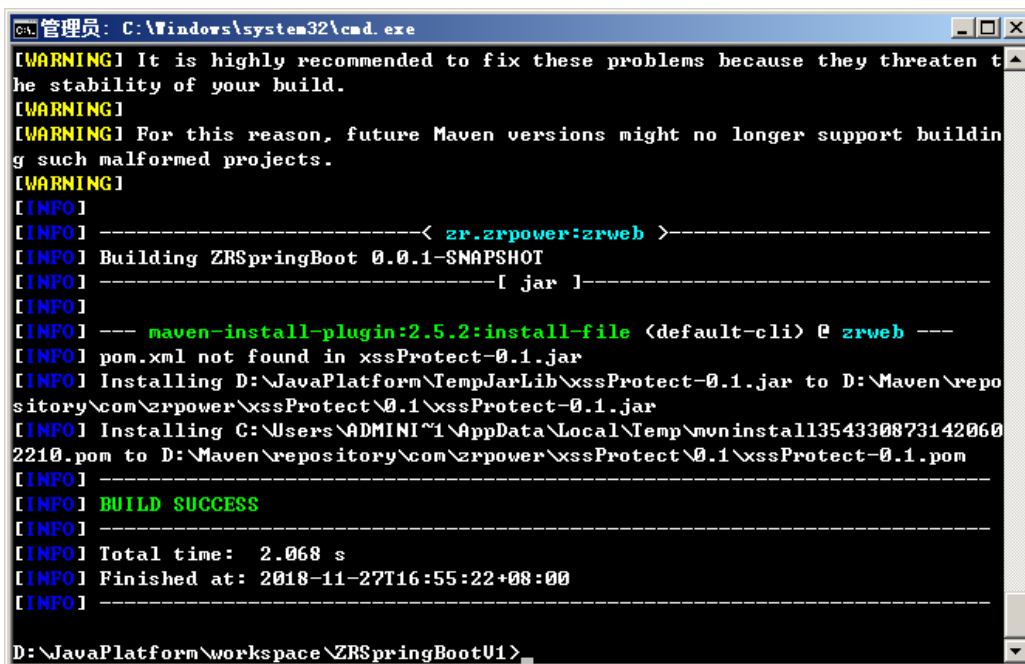


```

管理员: C:\Windows\system32\cmd.exe

[INFO] -----
D:\JavaPlatform\workspace\ZRSpringBootV1>mvn install:install-file -Dfile=D:\JavaPlatform\TempJarLib\xssProtect-0.1.jar -DgroupId=com.zrpower -DartifactId=xssProtect -Dversion=0.1 -Dpackaging=jar
[INFO] Scanning for projects...
[WARNING]
[WARNING] Some problems were encountered while building the effective model for
zr.zrpower:zrweb:jar:0.0.1-SNAPSHOT
[WARNING] 'dependencies.dependency.systemPath' for com.zrpower:jotpverify:jar sh
ould not point at files within the project directory, ${project.basedir}/${base.
path}/jotpverify.jar will be unresolvable by dependent projects @ line 126, colu
mn 16
[WARNING] 'dependencies.dependency.systemPath' for com.zrpower:jxl:jar should no
t point at files within the project directory, ${project.basedir}/${base.path}/j
xl.jar will be unresolvable by dependent projects @ line 134, column 16
[WARNING] 'dependencies.dependency.systemPath' for com.oracle:classes12:jar shou
ld not point at files within the project directory, ${project.basedir}/${base.pa
th}/classes12.jar will be unresolvable by dependent projects @ line 144, column
16
[WARNING] 'dependencies.dependency.systemPath' for com.zrpower:sqljdbc:jar shoul
d not point at files within the project directory, ${project.basedir}/${base.pat
h}/sqljdbc.jar will be unresolvable by dependent projects @ line 152, column 16
[WARNING] 'dependencies.dependency.systemPath' for com.zrpower:ZRpowersDbEngine:j
ar should not point at files within the project directory, ${project.basedir}/${

```



```

管理员: C:\Windows\system32\cmd.exe

[WARNING] It is highly recommended to fix these problems because they threaten t
he stability of your build.
[WARNING]
[WARNING] For this reason, future Maven versions might no longer support buildin
g such malformed projects.
[WARNING]
[INFO]
[INFO] -----< zr.zrpower:zrweb >-----
[INFO] Building ZRSpringBoot 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install-file (default-cli) @ zrweb ---
[INFO] pom.xml not found in xssProtect-0.1.jar
[INFO] Installing D:\JavaPlatform\TempJarLib\xssProtect-0.1.jar to D:\Maven\repo
sitory\com\zrpower\xssProtect\0.1\xssProtect-0.1.jar
[INFO] Installing C:\Users\ADMINI~1\AppData\Local\Temp\mvninstall354330873142060
2210.pom to D:\Maven\repository\com\zrpower\xssProtect\0.1\xssProtect-0.1.pom
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 2.068 s
[INFO] Finished at: 2018-11-27T16:55:22+08:00
[INFO]
D:\JavaPlatform\workspace\ZRSpringBootV1>

```

至此，发布 jar 包到本地 Maven 仓库成功，可以在本地 Maven 仓库中查看到发布的 jar 包，如下图



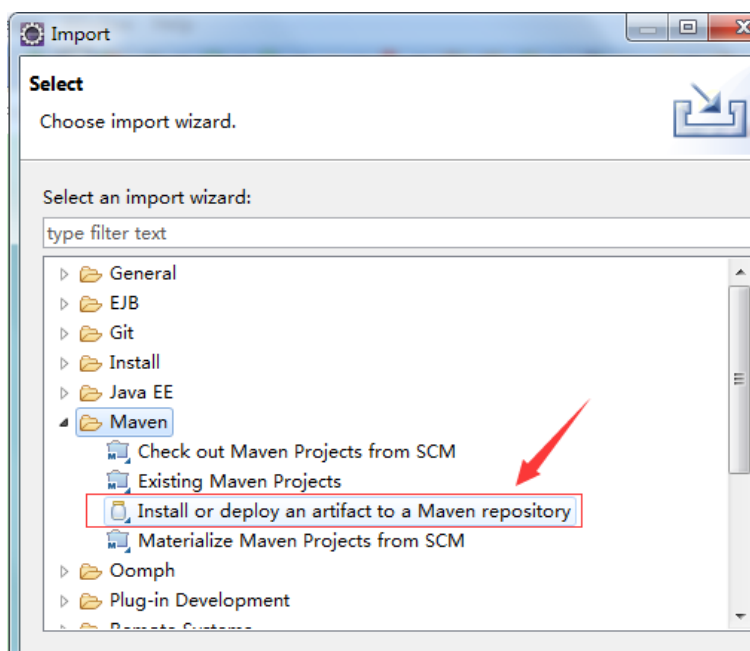
其他 jar 包的发布操作类似。

另一种方式：使用 eclipse 安装 jar 包：

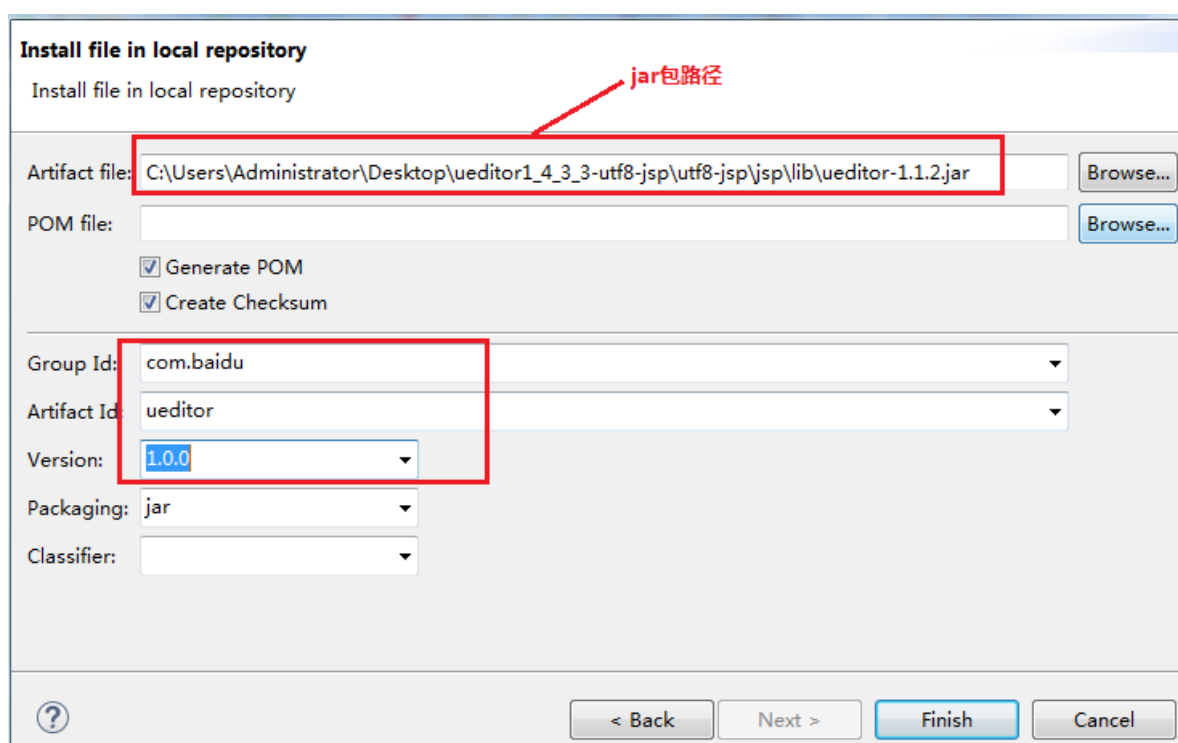
使用 eclipse 安装也有个前提，就是 eclipse 的 Maven 要先配置好。不过我相信使用 Maven 的小伙伴的 eclipse 的 Maven 设置肯定是没有问题的，不然还怎么用 Maven。

具体操作：

(1) File --> import --> Maven --> instal or deploy an artifact to a Maven repository



(2) 填写相关信息，如 Maven 坐标，具体参考下图。完成后点击 Finish。



Group Id, Artifact Id, Version 的设置参考 pom.xml 文件。

(3) 完成上述步骤后, 就已经大功告成了, 可以去本地仓库看下上面安装的东西在不在。

三、将 ZRSpringBoot 项目 pom.xml 中所有本地引用的 jar 包改为正常引用

如图所示:

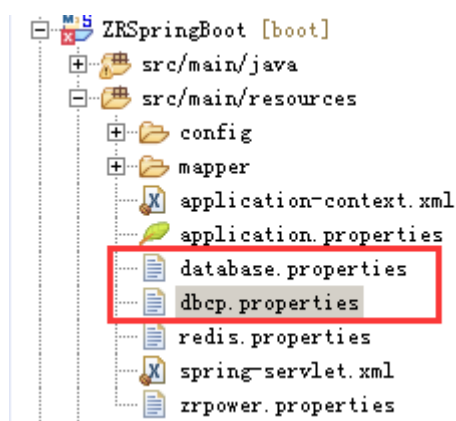
```
<dependency>
  <groupId>com.zrpower</groupId>
  <artifactId>jotpverify</artifactId>
  <version>1.0.0</version>
  <!--
  <scope>system</scope>
  本地jar的路径, 相对或者绝对都可以
  <systemPath>${project.basedir}/${base.path}/jotpverify.jar</systemPath>
  -->
</dependency>
<dependency>
  <groupId>com.zrpower</groupId>
  <artifactId>jxl</artifactId>
  <version>1.0.0</version>
  <!--
  <scope>system</scope>
  本地jar的路径, 相对或者绝对都可以
  <systemPath>${project.basedir}/${base.path}/jxl.jar</systemPath>
  -->
</dependency>
```

```

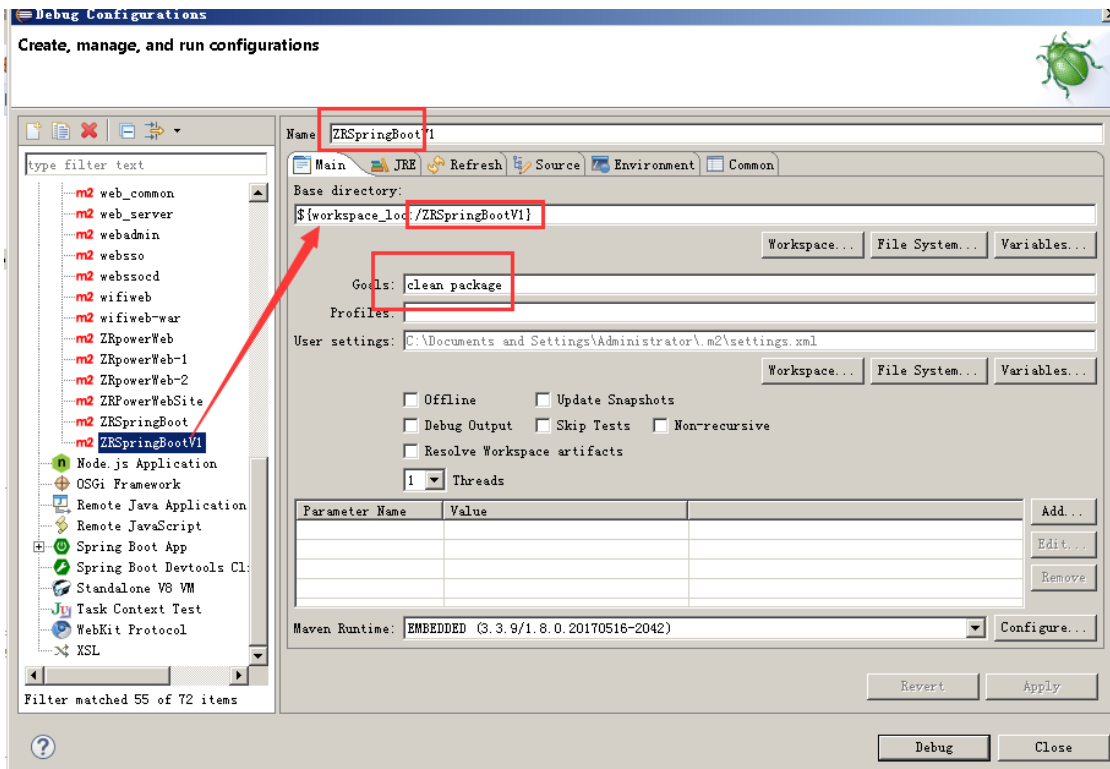
<!-- 运行时需要加入的第三方jar -->
<dependency>
  <groupId>com.oracle</groupId>
  <artifactId>classes12</artifactId>
  <version>1.0.0</version>
  <!--
  <scope>system</scope>
  本地jar的路径, 相对或者绝对都可以
  <systemPath>${project.basedir}/${base.path}/classes12.jar</systemPath>
  -->
</dependency>
<dependency>
  <groupId>com.zrpower</groupId>
  <artifactId>sqljdbc</artifactId>
  <version>1.0.0</version>
  <!--
  <scope>system</scope>
  本地jar的路径, 相对或者绝对都可以
  <systemPath>${project.basedir}/${base.path}/sqljdbc.jar</systemPath>
  -->
</dependency>
<dependency>
  <groupId>com.zrpower</groupId>
  <artifactId>ZrPowerDbEngine</artifactId>
  <version>1.0.0</version>
  <!--
  <scope>system</scope>
  本地jar的路径, 相对或者绝对都可以
  <systemPath>${project.basedir}/${base.path}/ZrPowerDbEngine.jar</systemPath>
  -->
</dependency>
<dependency>
  <groupId>com.zrpower</groupId>
  <artifactId>antlr4-runtime</artifactId>
  <version>4.1</version>
  <!--
  <scope>system</scope>
  本地jar的路径, 相对或者绝对都可以
  <systemPath>${project.basedir}/${base.path}/antlr4-runtime-4.1.jar</systemPath>
  -->
</dependency>
<dependency>
  <groupId>com.zrpower</groupId>
  <artifactId>xssProtect</artifactId>
  <version>0.1</version>
  <!--
  <scope>system</scope>
  本地jar的路径, 相对或者绝对都可以
  <systemPath>${project.basedir}/${base.path}/xssProtect-0.1.jar</systemPath>
  -->
</dependency>

```

#### 四、将 ZRSpringBoot 项目的数据库配置改为测试环境的配置参数

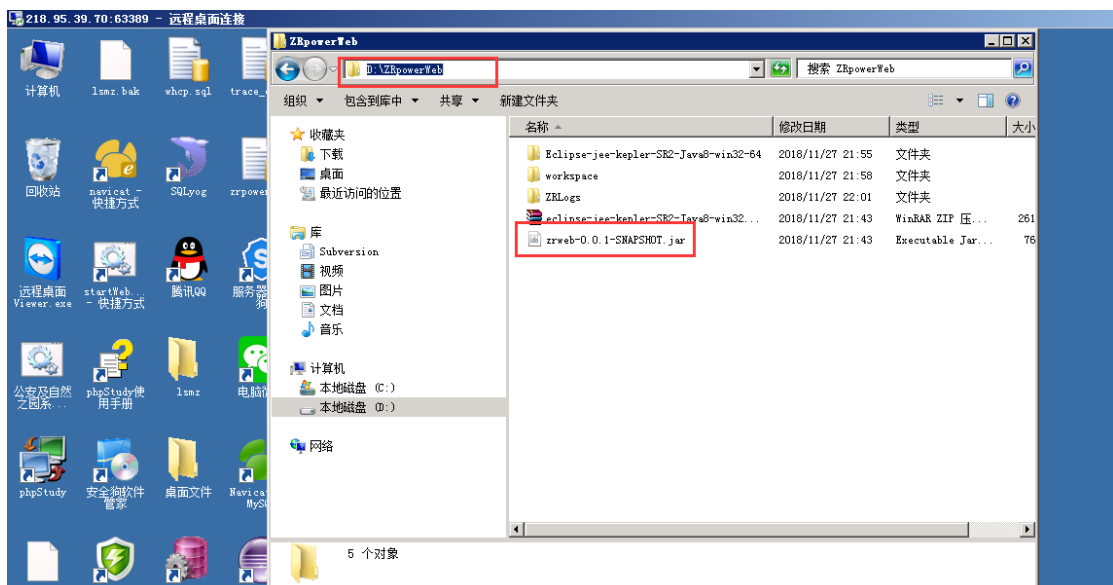


五、使用 clean package 命令将项目打成 jar 包，如图：



点击 Debug 即可运行打包操作。

六、将项目的 jar 包拷贝到服务器 D:\ZRpowerWeb 的目录下，如图：



七、cmd 进入到 D:\ZRpowerWeb 的目录，执行命令：java -jar zrweb-0.0.1-SNAPSHOT.jar，如图：

```
D:\ZrpowerWeb>java -jar zrweb-0.0.1-SNAPSHOT.jar

管理员: C:\Windows\system32\cmd.exe - java -jar zrweb-0.0.1-SNAPSHOT.jar

[2018-11-29 20:46:55.335][main][INFO][org.springframework.context.support.Default
LifecycleProcessor$LifecycleGroup.start(DefaultLifecycleProcessor.java:345)] St
arting beans in phase 0
[2018-11-29 20:46:55.337][main][INFO][org.springframework.integration.endpoint.E
ventDrivenConsumer.logComponentSubscriptionEvent(EventDrivenConsumer.java:108)]
Adding <logging-channel-adapter:_org.springframework.integration.errorLogger> as
a subscriber to the 'errorChannel' channel
[2018-11-29 20:46:55.339][main][INFO][org.springframework.integration.channel.Ab
stractSubscribableChannel.adjustCounterIfNecessary(AbstractSubscribableChannel.j
ava:81)] Channel 'application:8080.errorChannel' has 1 subscriber(s).
[2018-11-29 20:46:55.341][main][INFO][org.springframework.integration.endpoint.A
bstractEndpoint.start(AbstractEndpoint.java:120)] started _org.springframework.i
ntegration.errorLogger
[2018-11-29 20:46:55.365][main][INFO][org.apache.juli.logging.DirectJDKLog.log(D
irectJDKLog.java:180)] Starting ProtocolHandler ["http-nio-8080"]
[2018-11-29 20:46:55.391][main][INFO][org.apache.juli.logging.DirectJDKLog.log(D
irectJDKLog.java:180)] Using a shared selector for servlet write/read
[2018-11-29 20:46:55.488][main][INFO][org.springframework.boot.context.embedded.
tomcat.TomcatEmbeddedServletContainer.start(TomcatEmbeddedServletContainer.java:
216)] Tomcat started on port(s): 8080 (http)
[2018-11-29 20:46:55.506][main][INFO][org.springframework.boot.StartupInfoLogger
.logStarted(StartupInfoLogger.java:57)] Started ZrSpringBootApplication in 48.35
7 seconds (JVM running for 53.324)
=====>>>http://localhost:8080/
```

至此，项目在服务器上的部署完成。

## 5.6 只集成引擎时说明

不使用公司的系统管理部分，只使用其它引擎时集成说明(目的是对集成到项目中的各种引擎赋 session 值，不然不能正常使用各种引擎)：

引入 Session Bean：

```
import zr.zrpower.entity.sys.SessionUser;
```

Session 赋值的方法如下：

```
SessionUser su = new SessionUser();
```

```
su.setUserID("0000000000000001");//用户 ID
```

```
su.setLCODE("000000");//用户编号
```

```
su.setName("张三");//用户名称
```

```
su.setUnitID("001");//单位(部门)编号
```

```
su.setUnitName("办公室");//单位(部门)名称
```

```
su.setUserPageSize(28);//表格每页显示的记录数
```

```
su.setCustom1(""); //自定义 1
su.setCustom2(""); //自定义 2
su.setCustom3(""); //自定义 3
su.setCustom4(""); //自定义 4
su.setCustom5(""); //自定义 5
session.setAttribute("userinfo", su);
```

基础数据同步。

使用各种引擎需要将原开发框架中的一些数据同步到指定的表中。数据表结构请参照开发框架部分的表结构。

需要同步的表包括:

单位[部门]表(BPIP\_UNIT)

用户表(BPIP\_USER) 角色表(BPIP\_ROLE)

用户角色表(BPIP\_USER\_ROLE)

注: 单位[部门]ID、用户 ID、角色 ID 规范请参照示例数据生成, 单位 ID 是 12 位、用户 ID 是 16 位(前 12 位为单位编号, 后 4 位为流水号), 角色 ID 为数字类型编号, 不按编码规范生成引擎的有些功能不能正常使用。

由于各种 ID 是按新的规范生成, 所以需在集成项目的相关表中存放生成的编号, 以便更新数据。

上面赋值 session 值时用的单位[部门]编号、用户 ID 要用新生成的编号赋值。

## 5.7 创建数据表及演示数据

通过[开发文档及数据库建表 sql / 数据库建表语句及示例数据]下的 sql 语句建表或插入演示数据, 演示效果与公司网站 [www.zrpower.cn](http://www.zrpower.cn) 上的在线示例一样。

提供三种数据库 (oracle、mssql、mysql) 的 sql 语句, 用户根据所使用的数据库选用相关的 sql 语句执行。

## 5.8 Mysql 数据库乱码处理

如果软件执行保存到数据库中的中文出现乱码, 按以下方法设置 mysql。

检查本地 mysql 安装文件目录下的 my.ini 配置文件，服务器和客户端的默认编码方式设置成 utf8，设置的项目如下。

```
character-set-server=utf8
```

```
default-character-set=utf8
```

## 第六章 开发技术文档捐赠

免费和开源项目，项目的可持续发展离不开您的支持，请作者喝杯咖啡吧。

扫描下面的二维码捐赠 298 元，获取项目二次开发《智慧开源基础开发引擎（项目开发技术手册）（SpringBoot+Mybatis+Thymeleaf+Vue+Element UI）V7.0 版本》（共 94 页）。



说明：捐赠后请将使用人姓名、工作单位、手机号、qq 号、邮箱、捐赠凭证截图发到 672561350@qq.com, 我们会在 24 小时内通过邮件发送最新完整的技术开发文档。

