# 在pom.xml中定义项目的打包方式和打包后的jar包名称

<!-- 打包方式必须是jar,不能是war -->

<packaging>jar</packaging>

<!-- 打包后的jar包名称 -->

<name>bank\_demo</name>

# 在pom.xml文件中的build标签中添加resources标签，配置所要拷贝的文件

<!-- 对资源文件的定义 -->

<resources>

<!-- 将webapp下的文件拷贝到 META-INF/resources目录下-->

<resource>

<directory>${basedir}/src/main/resources</directory>

<filtering>true</filtering>

</resource>

<resource>

<directory>${basedir}/src/main/webapp</directory>

<!--注意此次必须要放在此目录下才能被访问到 -->

<targetPath>META-INF/resources</targetPath>

<includes>

<include>\*\*/\*\*</include>

</includes>

</resource>

</resources>

# 在pom.xml中的 <plugins> 添加忽略没有web.xml配置文件的错误

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>2.6</version>

<!-- 是否忽略没有web.xml文件的错误 -->

<configuration>

<failOnMissingWebXml>false</failOnMissingWebXml>

</configuration>

</plugin>

# 在pom.xml中的 <plugins>配置springboot-maven的打包插件版本

<!-- springboot 使用maven打包的版本，必须是1.4.2-->

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

<version>1.4.2.RELEASE</version>

</plugin>

# 在pom.xml中的 <plugins>配置定义maven-resources-plugin 插件的配置信息

<!-- 定义maven-resources-plugin 插件的配置信息 -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-resources-plugin</artifactId>

<configuration>

<encoding>UTF-8</encoding>

<useDefaultDelimiters>true</useDefaultDelimiters>

</configuration>

</plugin>

# 选择项目-->右键-->run as-->maven clean-->maven install

**然后在项目的target目录下将生成指定名称的jar包**