# JAVA JDK+Maven 配置

# Maven - 环境配置

Maven 是一个基于 Java 的工具，所以要做的第一件事情就是安装 JDK。

## 系统要求

| **项目** | **要求** |
| --- | --- |
| JDK | Maven 3.3 要求 JDK 1.7 或以上 Maven 3.2 要求 JDK 1.6 或以上 Maven 3.0/3.1 要求 JDK 1.5 或以上 |
| 内存 | 没有最低要求 |
| 磁盘 | Maven 自身安装需要大约 10 MB 空间。除此之外，额外的磁盘空间将用于你的本地 Maven 仓库。你本地仓库的大小取决于使用情况，但预期至少 500 MB |
| 操作系统 | 没有最低要求 |

## 步骤 1：检查 Java 安装

现在打开控制台，执行下面的 java 命令。

| **操作系统** | **任务** | **命令** |
| --- | --- | --- |
| Windows | 打开命令控制台 | c:\> java -version |
| Linux | 打开命令终端 | $ java -version |
| Mac | 打开终端 | machine:~ joseph$ java -version |

我们来验证一下所有平台上的输出：

| **操作系统** | **输出** |
| --- | --- |
| Windows | java version "1.6.0\_21" Java(TM) SE Runtime Environment (build 1.6.0\_21-b07) Java HotSpot(TM) Client VM (build 17.0-b17, mixed mode, sharing) |
| Linux | java version "1.6.0\_21" Java(TM) SE Runtime Environment (build 1.6.0\_21-b07) Java HotSpot(TM) Client VM (build 17.0-b17, mixed mode, sharing) |
| Mac | java version "1.6.0\_21" Java(TM) SE Runtime Environment (build 1.6.0\_21-b07) Java HotSpot(TM)64-Bit Server VM (build 17.0-b17, mixed mode, sharing) |

如果你没有安装 Java，从以下网址安装 Java 软件开发套件（SDK）：[**http://www.oracle.com/technetwork/java/javase/downloads/index.html**](http://www.oracle.com/technetwork/java/javase/downloads/index.html)。我们假定你安装的 Java 版本为1.6.0\_21。

## 步骤 2：设置 Java 环境

设置 JAVA\_HOME 环境变量，并指向你机器上的 Java 安装目录。例如：

| **操作系统** | **输出** |
| --- | --- |
| Windows | Set the environment variable JAVA\_HOME to  C:\Program Files\Java\jdk1.6.0\_21 |
| Linux | export JAVA\_HOME=/usr/local/java-current |
| Mac | export JAVA\_HOME=/Library/Java/Home |

将 Java 编译器地址添加到系统路径中。

| **操作系统** | **输出** |
| --- | --- |
| Windows | 将字符串“;C:\Program Files\Java\jdk1.6.0\_21\bin”添加到系统变量“Path”的末尾 |
| Linux | export PATH=PATH:JAVA\_HOME/bin/ |
| Mac | not required |

使用上面提到的 **java -version** 命令验证 Java 安装。

## 步骤 3：下载 Maven 文件

从以下网址下载 Maven 3.2.5：[**http://maven.apache.org/download.html**](http://maven.apache.org/download.html)

## 步骤 4：解压 Maven 文件

解压文件到你想要的位置来安装 Maven 3.2.5，你会得到 apache-maven-3.2.5 子目录。

| **操作系统** | **位置 (根据你的安装位置而定)** |
| --- | --- |
| Windows | C:\Program Files\Apache Software Foundation\apache-maven-3.2.5 |
| Linux | /usr/local/apache-maven |
| Mac | /usr/local/apache-maven |

## 步骤 5：设置 Maven 环境变量

添加 M2\_HOME、M2、MAVEN\_OPTS 到环境变量中。

| **操作系统** | **输出** |
| --- | --- |
| Windows | 使用系统属性设置环境变量。 M2\_HOME=C:\Program Files\Apache Software Foundation\apache-maven-3.2.5 M2=%M2\_HOME%\bin MAVEN\_OPTS=-Xms256m -Xmx512m |
| Linux | 打开命令终端设置环境变量。 export M2\_HOME=/usr/local/apache-maven/apache-maven-3.2.5 export M2=$M2\_HOME/bin export MAVEN\_OPTS=-Xms256m -Xmx512m |
| Mac | 打开命令终端设置环境变量。 export M2\_HOME=/usr/local/apache-maven/apache-maven-3.2.5 export M2=$M2\_HOME/bin export MAVEN\_OPTS=-Xms256m -Xmx512m |

## 步骤 6：添加 Maven bin 目录到系统路径中

现在添加 M2 变量到系统“Path”变量中

| **操作系统** | **输出** |
| --- | --- |
| Windows | 添加字符串 “;%M2%” 到系统“Path”变量末尾 |
| Linux | export PATH=M2:PATH |
| Mac | export PATH=M2:PATH |

## 步骤 7：验证 Maven 安装

现在打开控制台，执行以下 **mvn** 命令。

| **操作系统** | **输出** | **命令** |
| --- | --- | --- |
| Windows | 打开命令控制台 | c:\> mvn --version |
| Linux | 打开命令终端 | $ mvn --version |
| Mac | 打开终端 | machine:~ joseph$ mvn --version |

最后，验证以上命令的输出，应该是像下面这样：

| **操作系统** | **输出** |
| --- | --- |
| Windows | Apache Maven 3.2.5 (r801777; 2009-08-07 00:46:01+0530) Java version: 1.6.0\_21 Java home: C:\Program Files\Java\jdk1.6.0\_21\jre |
| Linux | Apache Maven 3.2.5 (r801777; 2009-08-07 00:46:01+0530) Java version: 1.6.0\_21 Java home: C:\Program Files\Java\jdk1.6.0\_21\jre |
| Mac | Apache Maven 3.2.5 (r801777; 2009-08-07 00:46:01+0530) Java version: 1.6.0\_21 Java home: C:\Program Files\Java\jdk1.6.0\_21\jre |

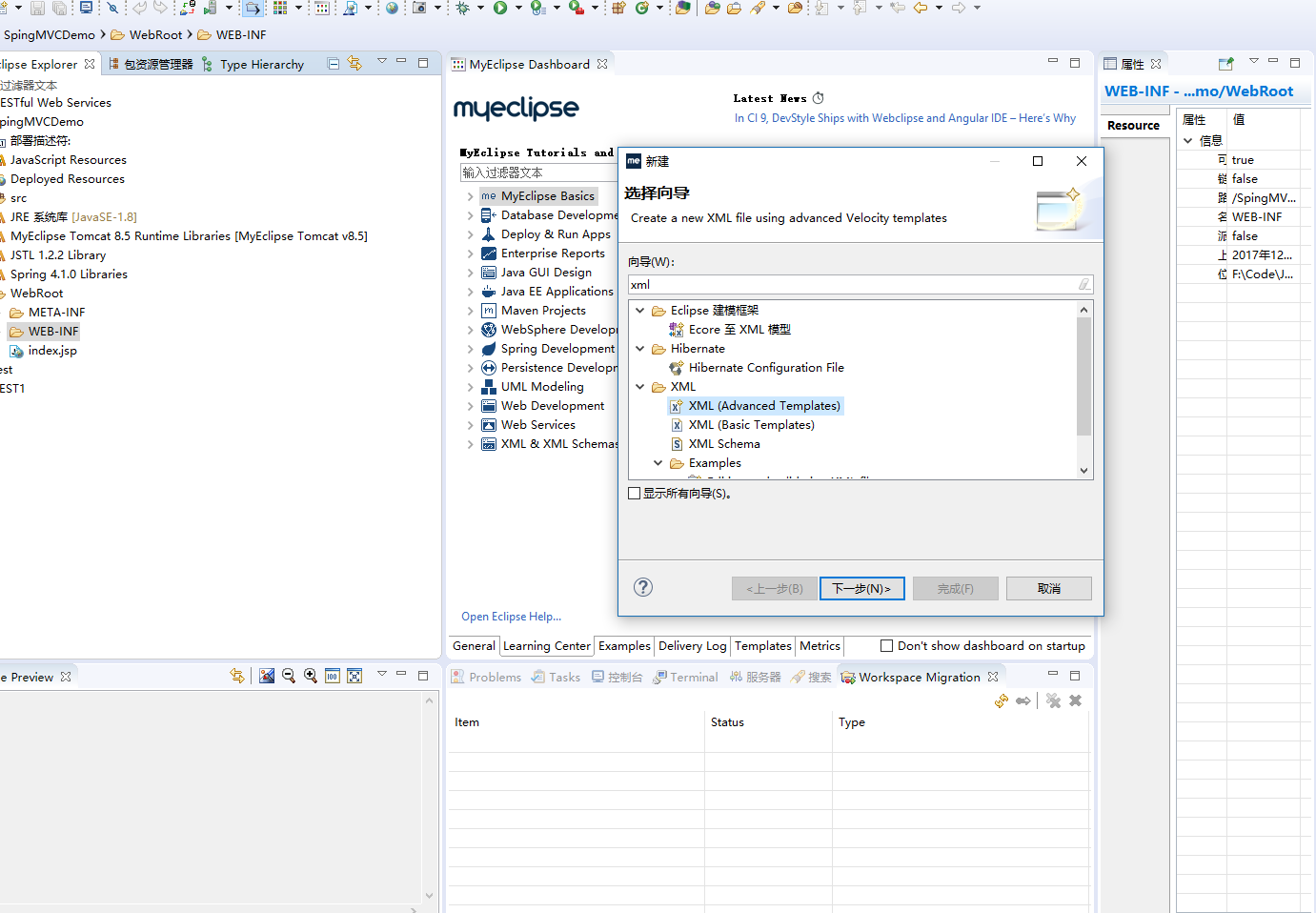
## Maven 配置文件

阿里云仓库配置

1. **<mirror>**
2. **<id>**alimaven**</id>**
3. **<name>**aliyun maven**</name>**
4. **<url>**http://maven.aliyun.com/nexus/content/groups/public/**</url>**
5. **<mirrorOf>**central**</mirrorOf>**
6. **</mirror>**
7. 将C:\Program Files\Maven\apache-maven-3.5.2\conf\settings拷贝至用户默认目录的.m2目录下（C:\Users\kongh\.m2）没有就新建一个

## MyEclipse M2E插件安装地址

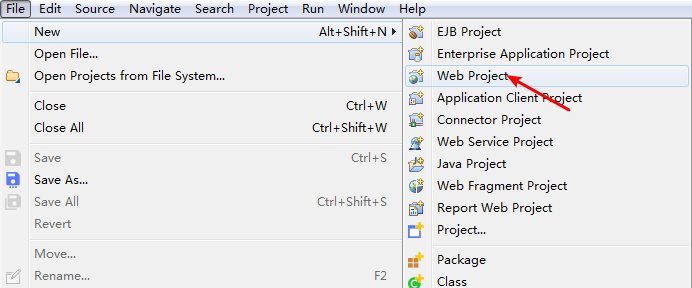
<http://m2eclipse.sonatype.org/sites/m2e>

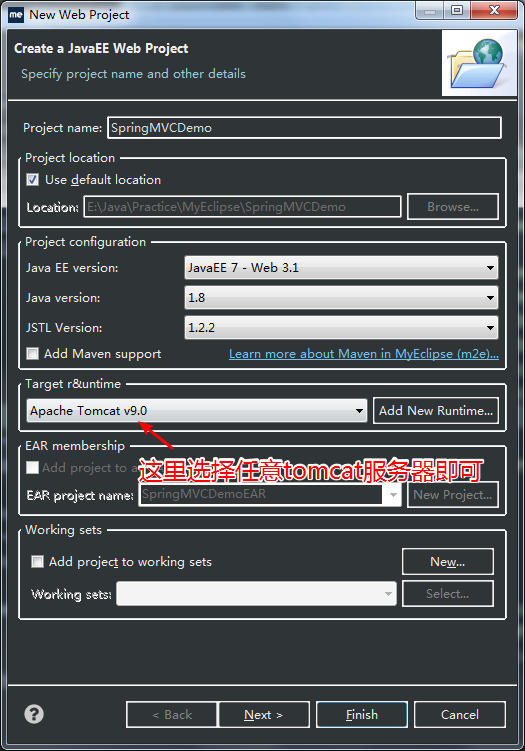


# 使用Myeclipse创建SpringMVC项目【超详细教程】

　　之前一直是使用Eclipse创建Web项目，用IDEA和MyEclipse的创建SpringMVC项目的时候时不时会遇到一些问题，这里把这个过程记录一下，希望能帮助到那些有需要的朋友。我是用的是MyEclipse2017 CI 3，相近版本应该都差不多。至于其他版本找到类似操作即可。

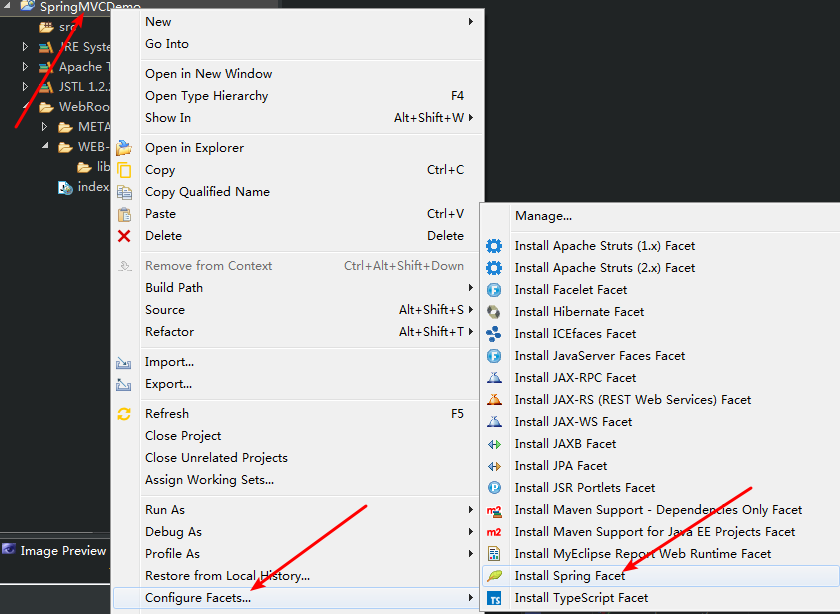
　　1.新建web项目

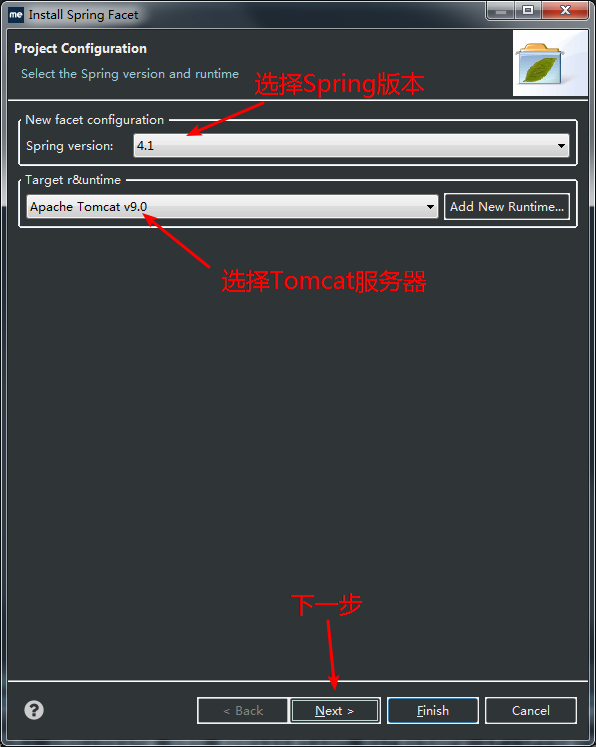


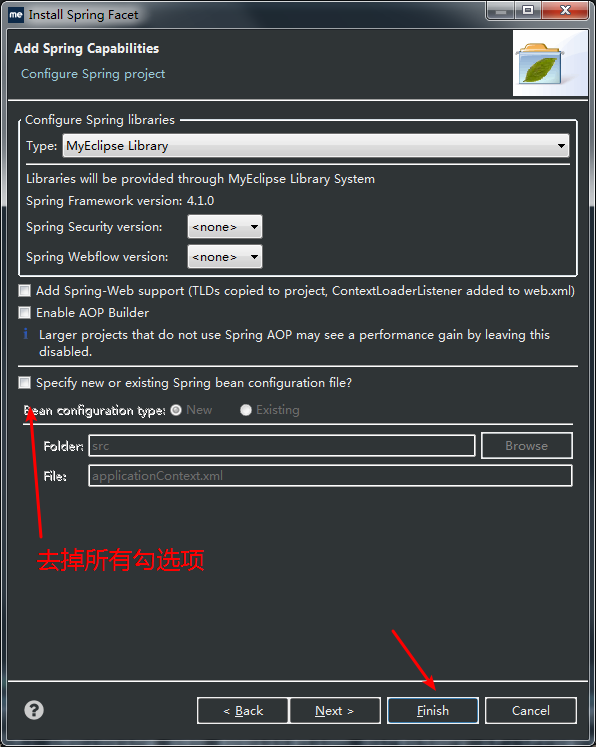


　　然后点击finish完成web项目创建。

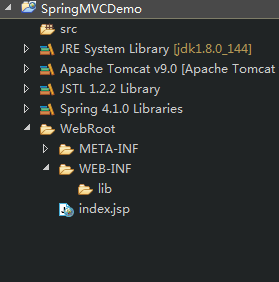
　　2.安装Spring框架



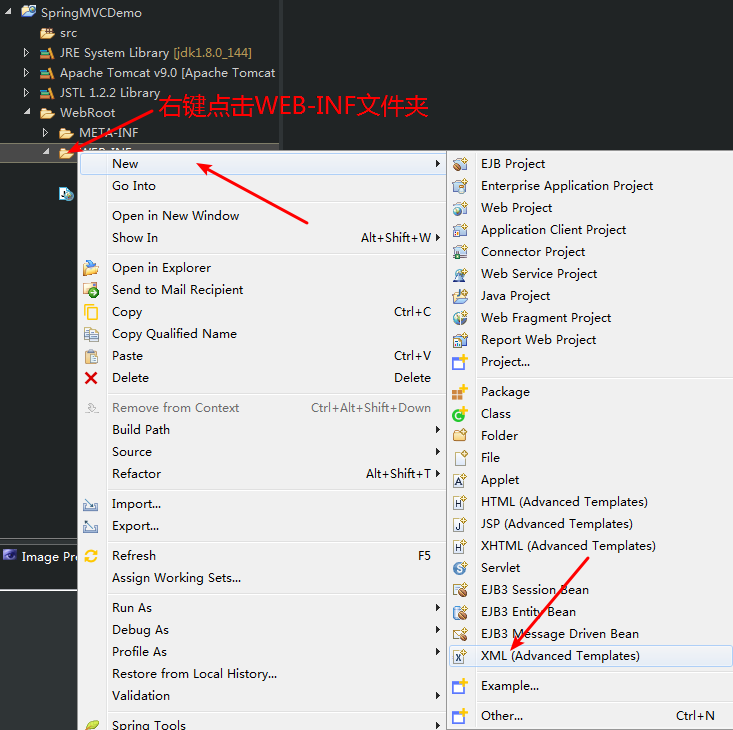


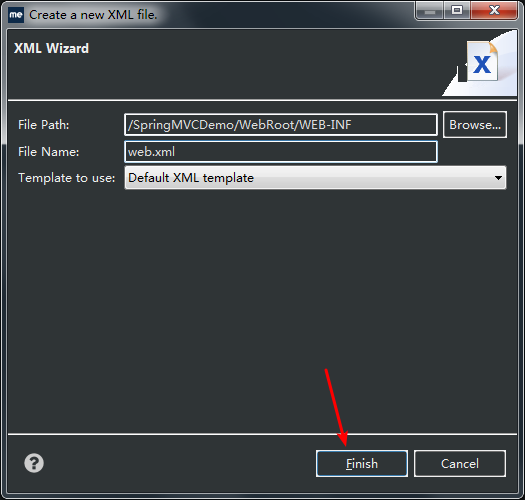


　　此时，项目结构如图：



　　3.创建xml文件





　　内容如下：

<?xml version="1.0" encoding="UTF-8"?>

<web-app version="3.0"

xmlns="http://java.sun.com/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/javaee

http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd">

<!-- 配置SpringMVC -->

<servlet>

<servlet-name>dispatcher</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:dispatcher-servlet.xml</param-value>

</init-param>

</servlet>

<servlet-mapping>

<servlet-name>dispatcher</servlet-name>

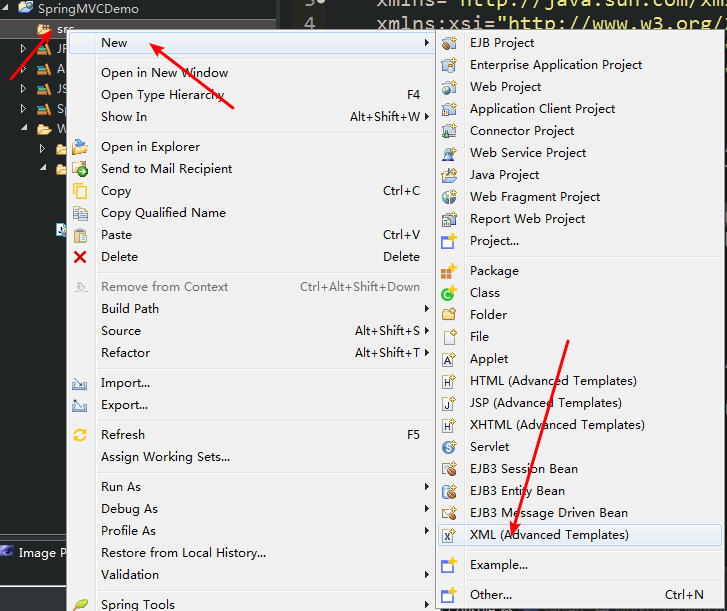
<!-- 监听所有请求 -->

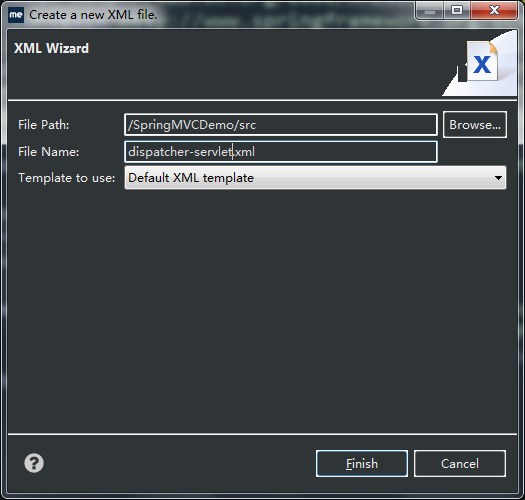
<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

　　然后在src目录下创建名为 dispatcher-servlet.xml （与上面指定的文件名要一致）的文件。





　　内容如下：

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:mvc="http://www.springframework.org/schema/mvc"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.1.xsd

http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.1.xsd">

<!-- 定义要扫描 controller的包 -->

<context:component-scan base-package="com.frank.springmvc.controller" />

<mvc:default-servlet-handler />

<!-- 启动注解驱动 SpringMVC 功能 -->

<mvc:annotation-driven />

<!-- 配置视图解析器解析路径 -->

<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver" id="internalResourceViewResolver">

<!-- 定义视图存放路径 -->

<property name="prefix" value="/WEB-INF/jsp/" />

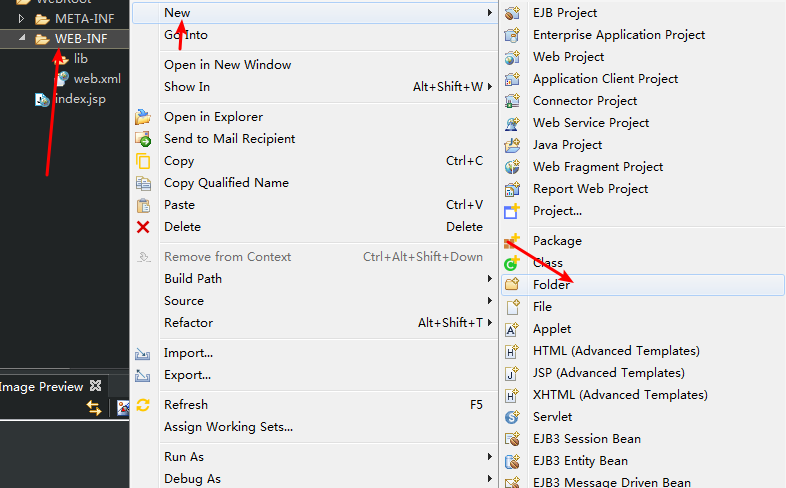
<!-- 定义视图后缀 -->

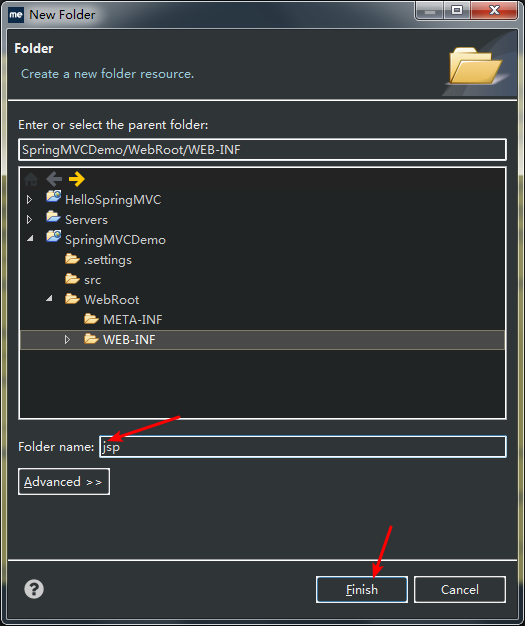
<property name="suffix" value=".jsp" />

</bean>

</beans>

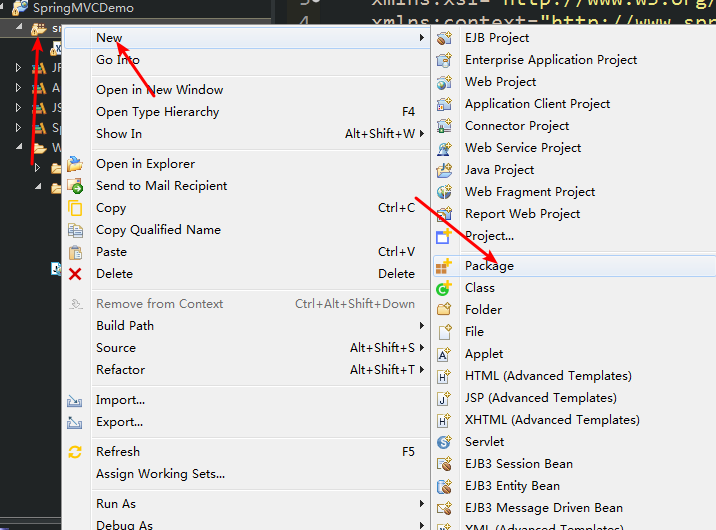
　　然后在WEB-INF目录下创建一个名为jsp的文件夹。

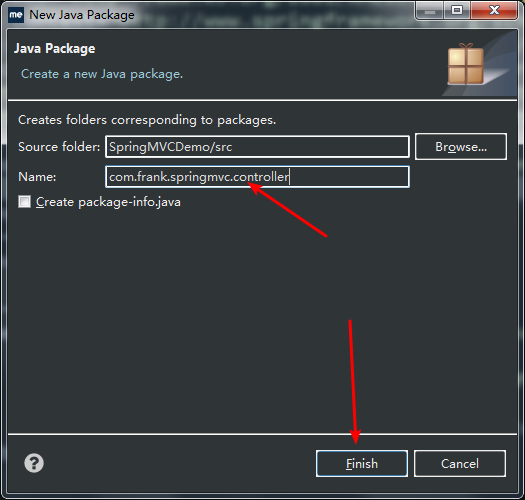




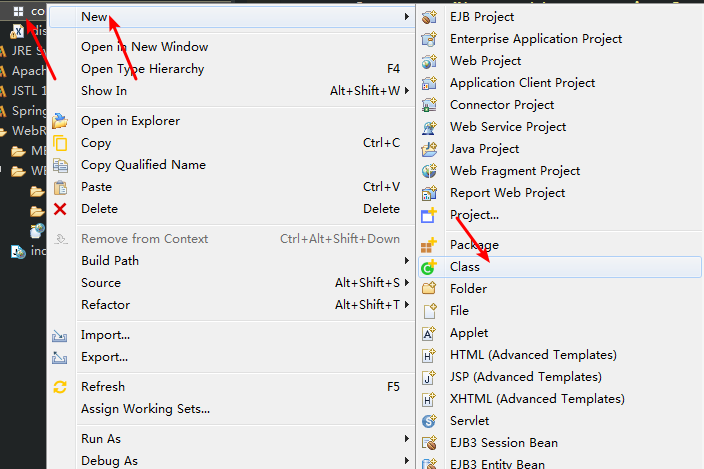
　　4.定义控制器

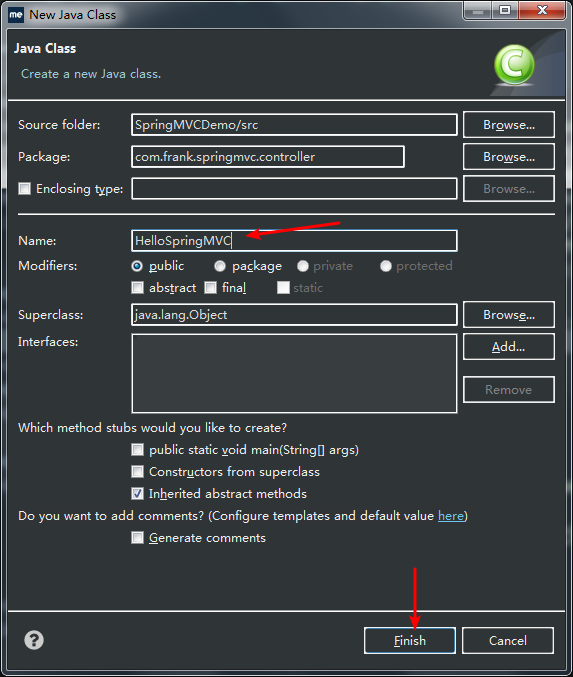
　　新建一个包com.frank.springmvc.controller(跟dispatcher-servlet.xml中指定的包名要一致)





　　然后在包下创建一个controller类，取名为HelloSpringMVC





　　代码如下：

package com.frank.springmvc.controller;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

@Controller

public class HelloSpringMVC {

@RequestMapping("/hello")

public String test() {

System.out.println("test");

return "hello";

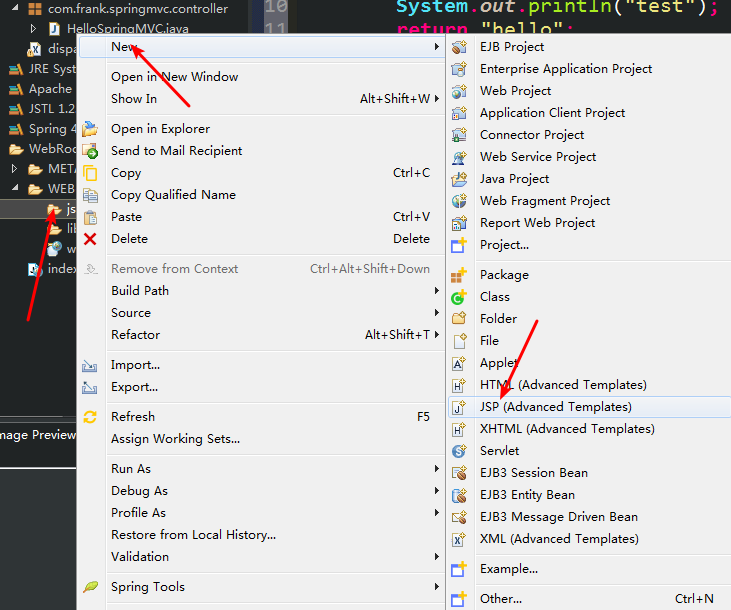
}

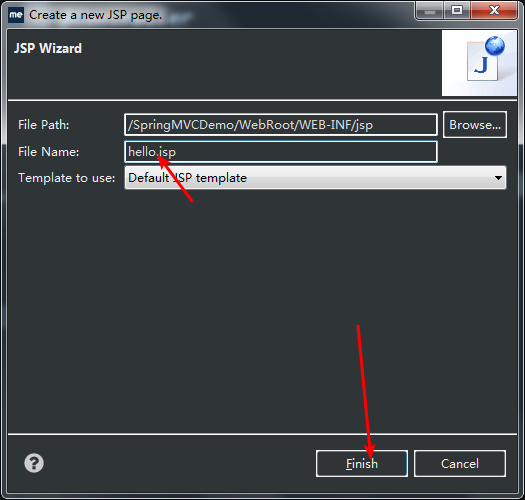
}

　　这样，就会将对/HelloSpringMVC/hello路径的请求转向/WEB-INF/jsp/hello.jsp文件

　　5.创建jsp文件

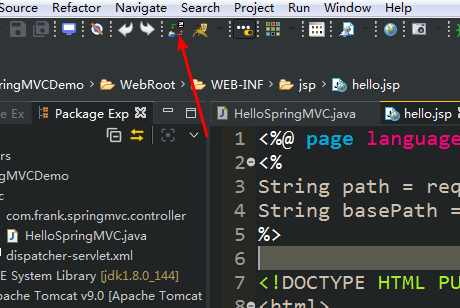
　　在jsp文件夹下创建hello.jsp文件

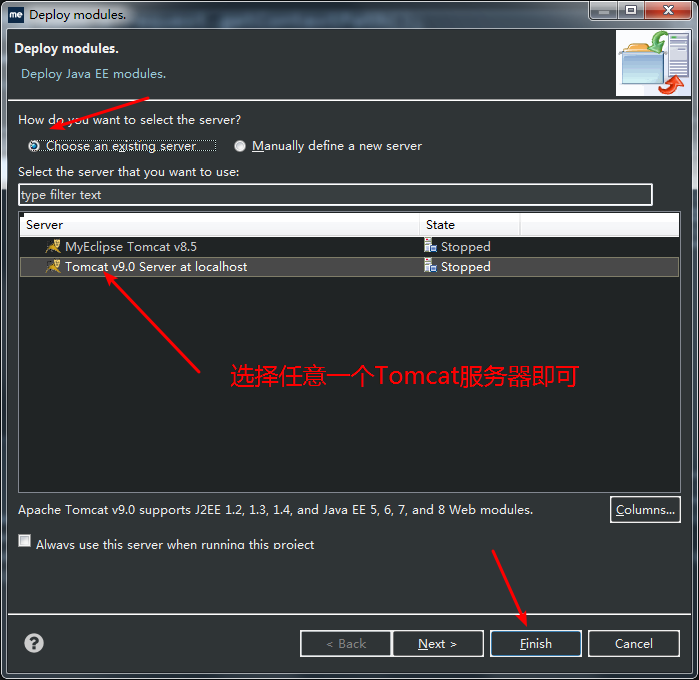


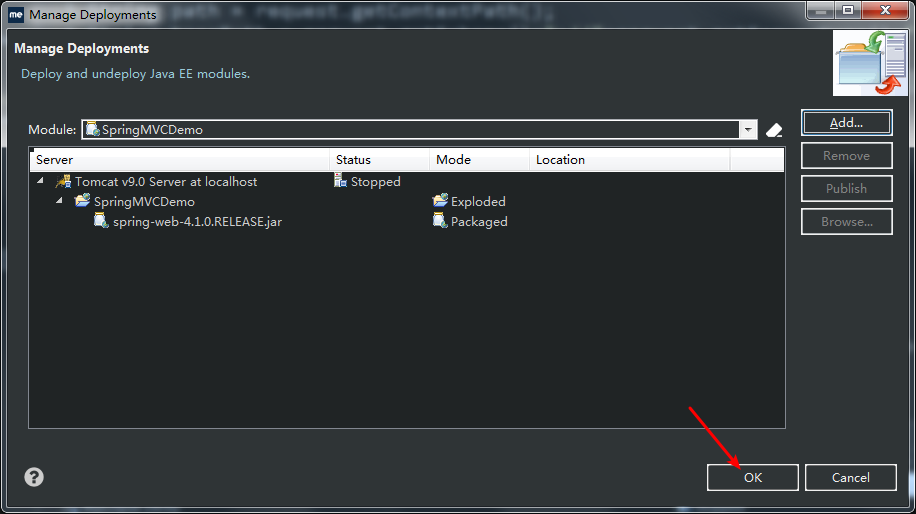


　　hello.jsp中会有一些代码，这里只是尽快建立SpringMVC运行项目，无需修改。

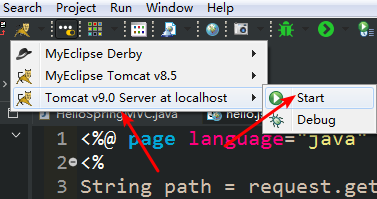
　　6.部署项目到Tomcat服务器





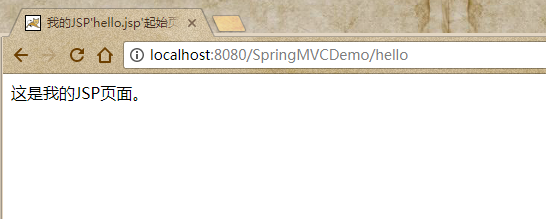


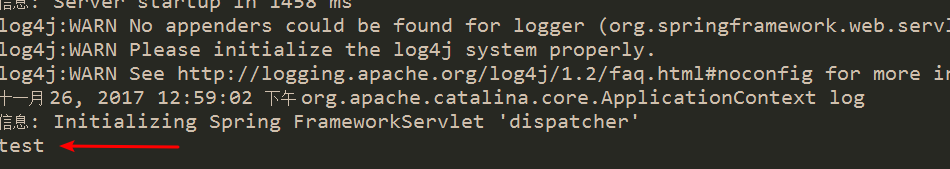
　　然后启动服务器。





　　在浏览器中输入：localhost:8080/SpringMVCDemo/hello





　　如果显示正常说明我们的项目部署成功。