SpringMVC 学习

1. 什么是SpringMVC？

它是基于MVC开发模式的框架,用来优化控制器.它是Spring家族的一员.它也具备IOC和AOP.

1. 什么是MVC?

它是一种开发模式,它是模型视图控制器的简称.所有的web应用都是基于MVC开发.

M:模型层,包含实体类,业务逻辑层,数据访问层

V:视图层,html,javaScript,vue等都是视图层,用来显现数据

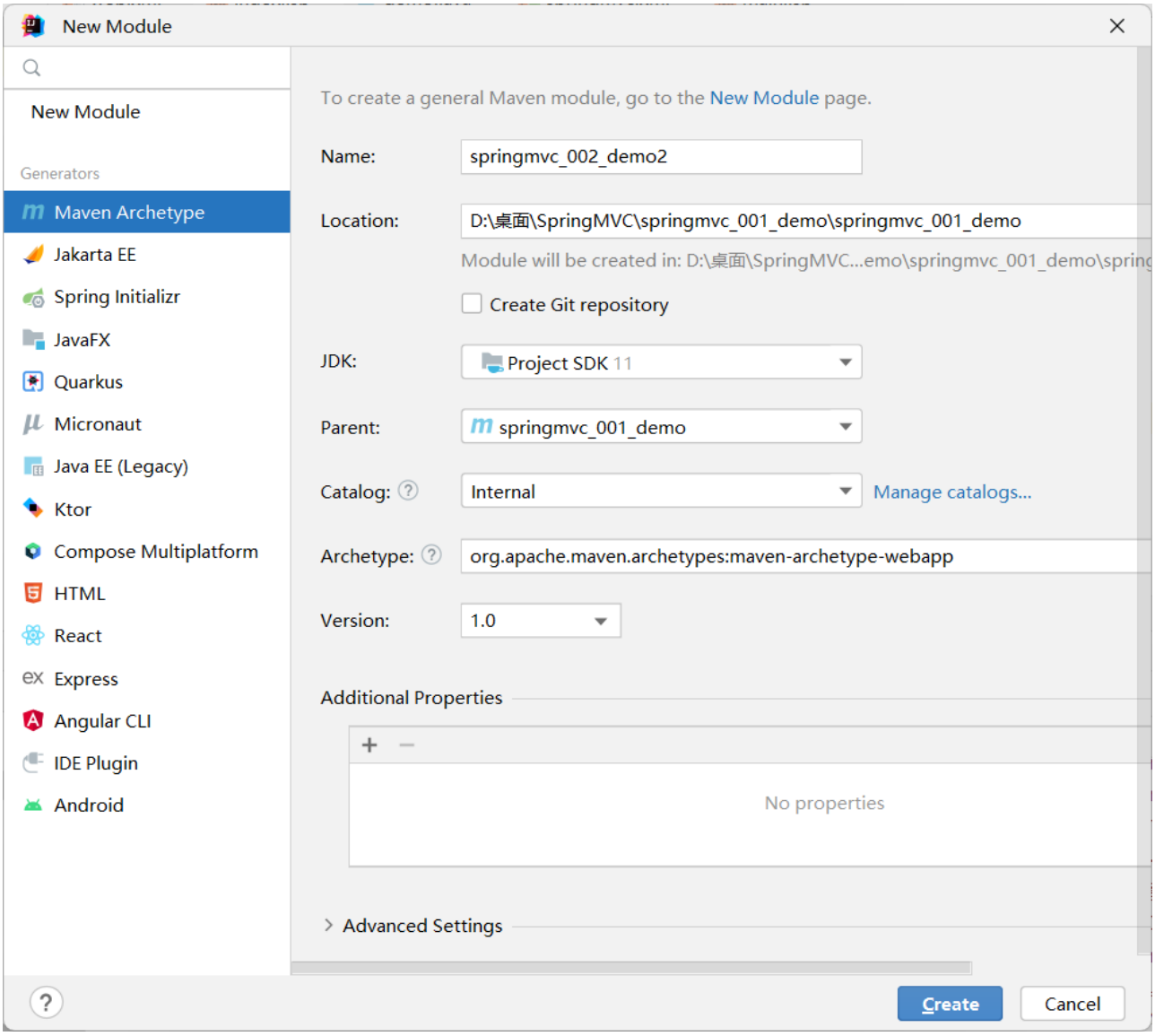
C:控制器,它是用来接收客户端的请求,并返回响应到客户端的组件,Servlet就是组件

1. SpringMVC框架特点有什么？

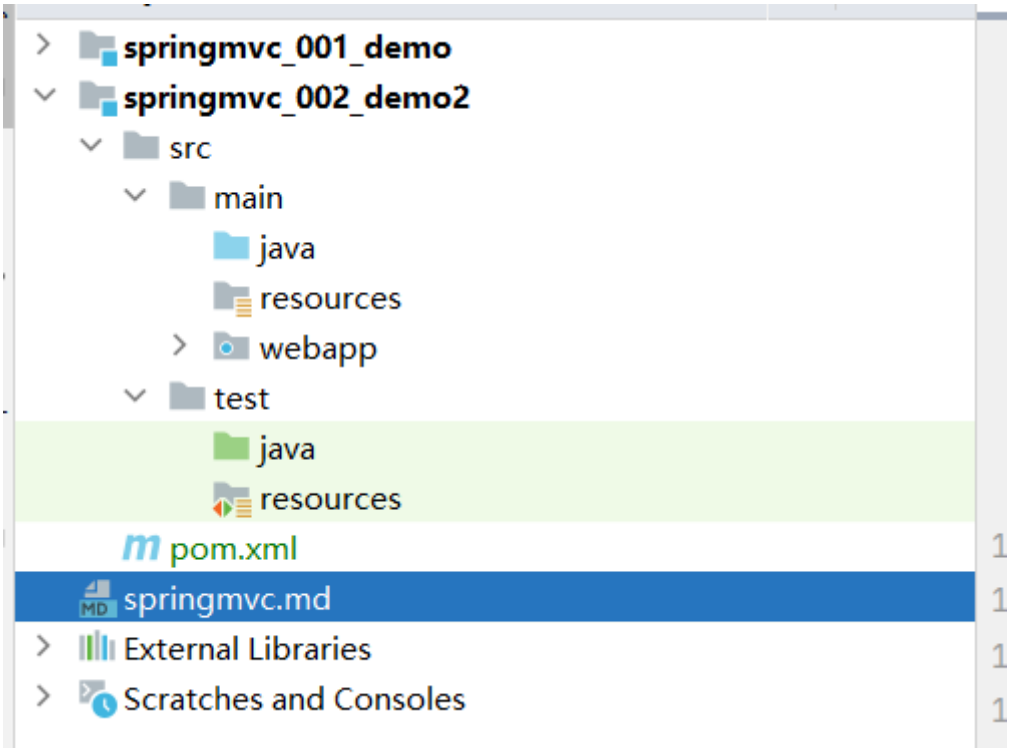
1)轻量级,基于MVC的框架
2)易于上手,容易理解,功能强大
3)它具备IOC和AOP
4)完全基于注解开发

2. 基于注解的SpringMVC框架开发的步骤

第一步new一个子模块,选择webapp模板.



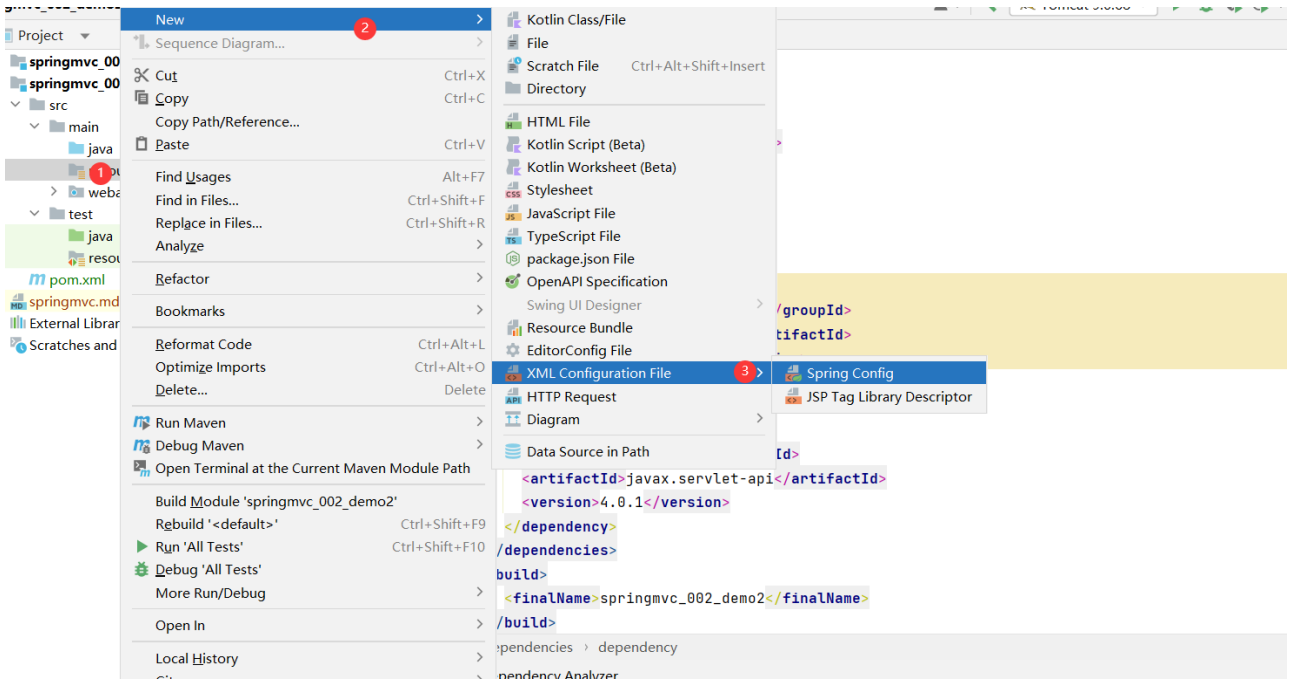
第二步：修改目录,添加缺失的test,java,resources(两套),并修改目录属性



第三步：修改pom.xml文件,添加SpringMVC的依赖,添加Servlet的依赖

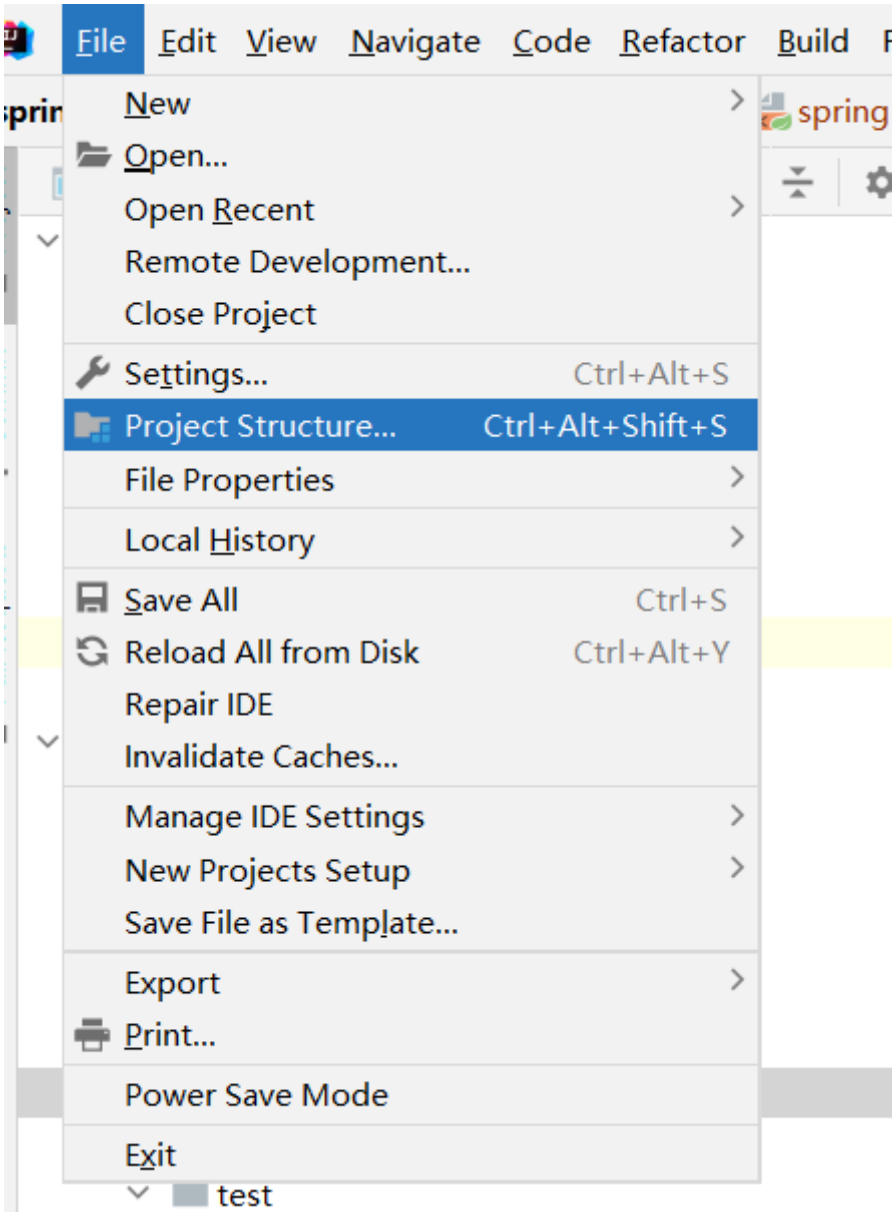
```
    <!--测试依赖-->
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>

    <!--springmvc-->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>5.2.12.RELEASE</version>
    </dependency>
    <!--servlet-->
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>4.0.1</version>
    </dependency>
  </dependencies>
```

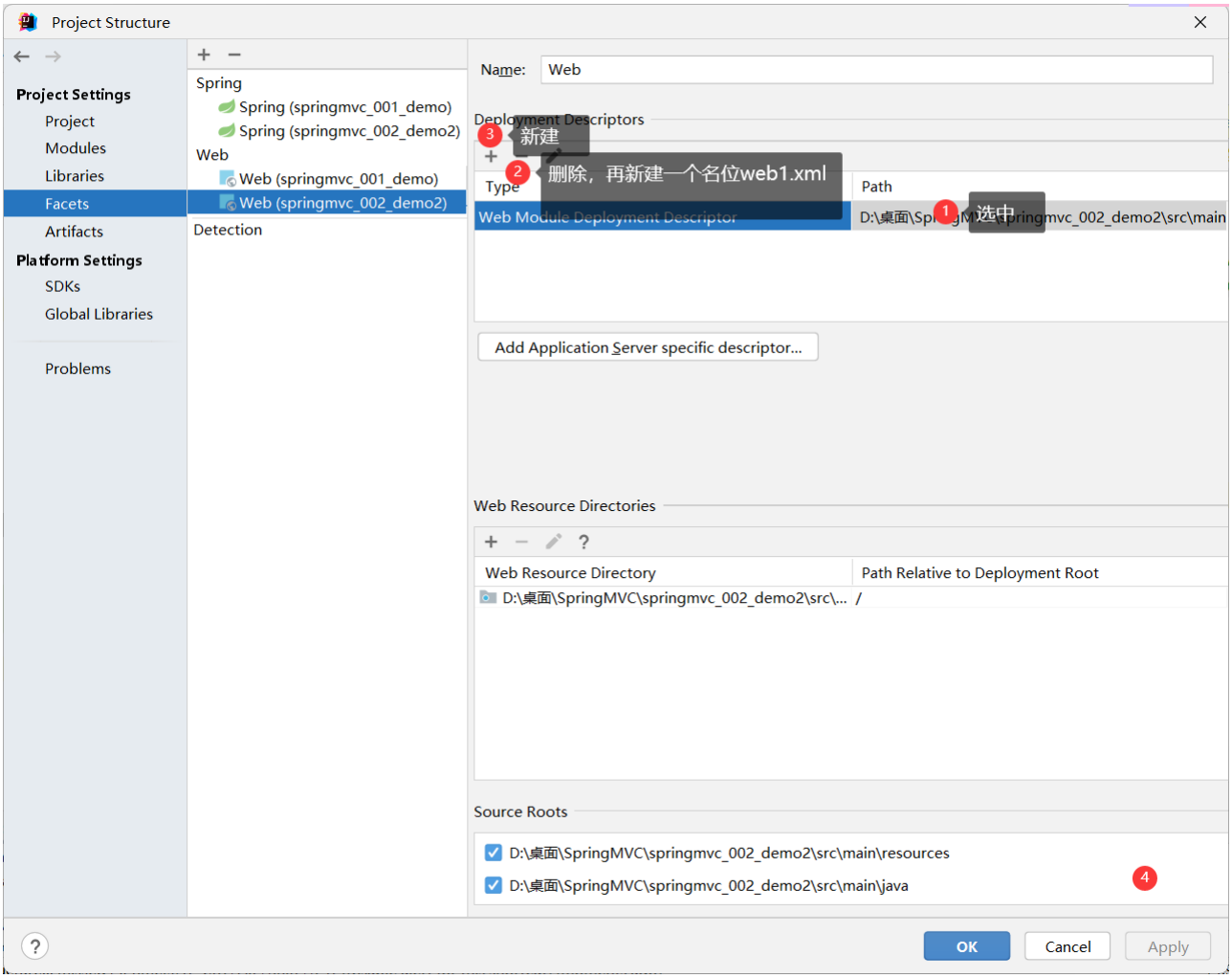
第四步：在main下的Resource添加springmvc.xml(此处名字任意)配置文件,指定包扫描,添加视图解析器

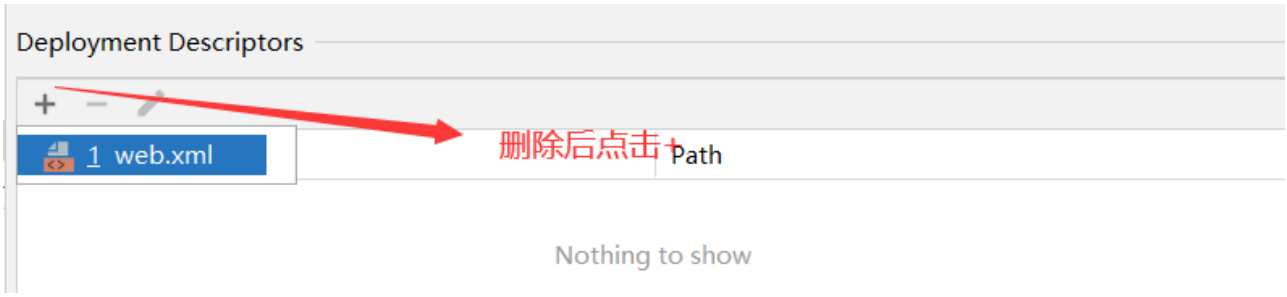


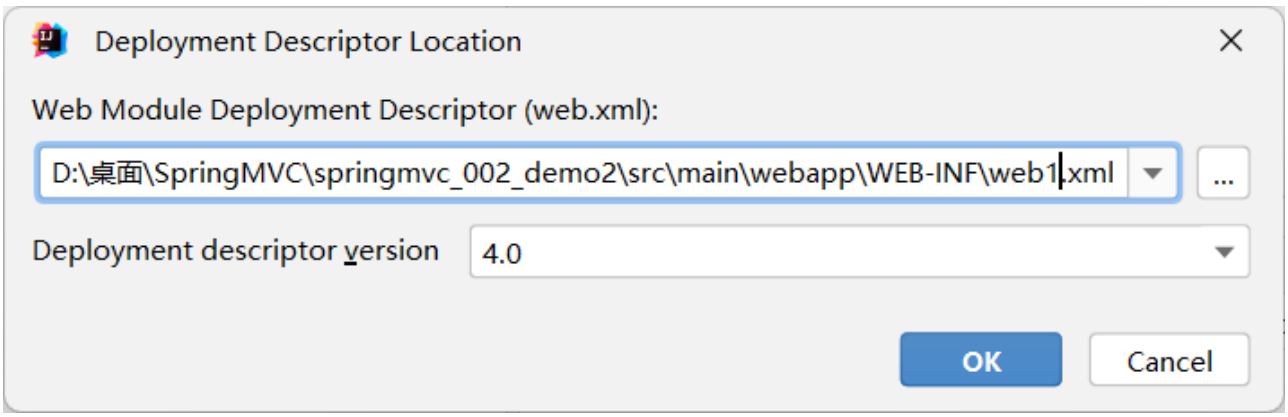
```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

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

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.xsd">
        <!--配置包扫描-->
        <context:component-scan base-package="edu.beihua">
</context:component-scan>
        <!--配置视图解析器-->
        <bean
class="org.springframework.web.servlet.view.InternalResourceViewRes
olver">
            <property name="prefix" value="/admin/"/>
            <property name="suffix" value=".action"></property>
        </bean>
</beans>
```

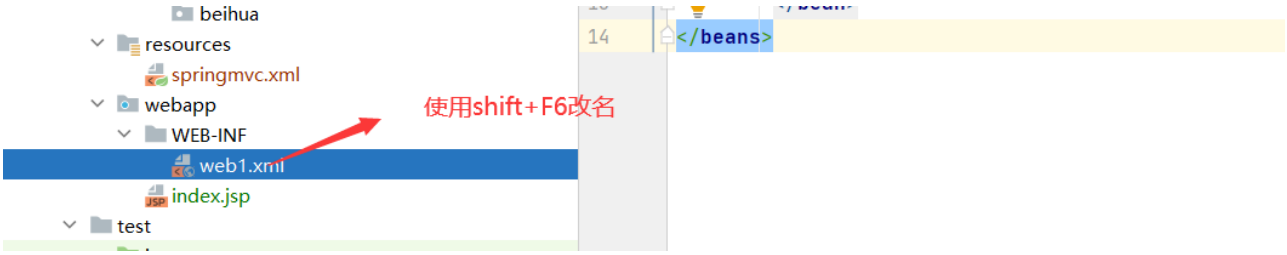
第五步：删除web.xml文件,新建web.xml，原因约束文件太老


File
Edit
View
Navigate
Code
Refactor
Build
New
Open...
Open Recent
Remote Development...
Close Project
Settings... Ctrl+Alt+S
Project Structure... Ctrl+Alt+Shift+S
File Properties
Local History
Save All Ctrl+S
Reload All from Disk Ctrl+Alt+Y
Repair IDE
Invalidate Caches...
Manage IDE Settings
New Projects Setup
Save File as Template...
Export
Print...
Power Save Mode
Exit
spring
test


Project Structure
Project Settings
Project
Modules
Libraries
Facets
Artifacts
Platform Settings
SDKs
Global Libraries
Problems
Spring
Spring (springmvc_001_demo)
Spring (springmvc_002_demo2)
Web
Web (springmvc_001_demo)
Web (springmvc_002_demo2)
Detection
Name: Web
Deployment Descriptors
3 新建
2 删除，再新建一个名位web1.xml
Type
Path
Web Module Deployment Descriptor
1 选中
Add Application Server specific descriptor...
Web Resource Directories
Web Resource Directory
Path Relative to Deployment Root
D:\桌面\SpringMVC\springmvc_002_demo2\src\... /
Source Roots
D:\桌面\SpringMVC\springmvc_002_demo2\src\main\resources
D:\桌面\SpringMVC\springmvc_002_demo2\src\main\java
4
OK
Cancel
Apply


Deployment Descriptors
1 web.xml
删除后点击+
Path
Nothing to show


Deployment Descriptor Location
Web Module Deployment Descriptor (web.xml):
D:\桌面\SpringMVC\springmvc_002_demo2\src\main\webapp\WEB-INF\web1.xml
...
Deployment descriptor version
4.0
OK
Cancel



改位web.xml

第六步：在web.xml文件中注册springMVC框架(所有的web请求都是基于servlet的)

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
         version="4.0">

    <!--注册框架-->
    <servlet>
        <servlet-name>springmvc</servlet-name>
        <servlet-
class>org.springframework.web.servlet.DispatcherServlet</servlet-
class>

        <!--初始化参数 读取配置文件-->
        <init-param>
            <param-name>contextConfigLocation</param-name>
            <param-value>classpath:springmvc.xml</param-value>
        </init-param>
    </servlet>

    <!--配置拦截器规则-->
    <servlet-mapping>
        <servlet-name>springmvc</servlet-name>
        <!--只要是*.active的请求才能通过核心控制器的检查-->
        <url-pattern>*.active</url-pattern>
    </servlet-mapping>
</web-app>
```

第八步：在webapp目录下新建admin目录,在admin目录下新建main.jsp页面,删除index.jsp页面,并新建,发送请求给服务器

index内容：

```
<%--
  Created by IntelliJ IDEA.
  User: 23705
  Date: 2023/3/17
  Time: 23:22
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>Title</title>
</head>
<body>
<a href="${pageContext.request.contextPath}/demo.active">点击跳转主页
</a>
</body>
</html>
```

编写控制器类

```
package edu.beihua;

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


//@Controller  //交给Spring去创建对象
@Controller
public class demo2 {
    @RequestMapping("/demo2")
    public String tomain(){
        return "main";
    }
}
```

最后添加tomcat测试

- 分析web请求

web请求执行的流程

核心处理器

index.jsp<---------->DispatcherServlet<---------->SpringMVC的处理器是一个普通的方法

one.jsp <---------->DispatcherServlet<---------->SpringMVC的处理器是一个普通的方法

DispatcherServlet要在web.xml文件中注册才可用.

@RequestMapping注解详解

此注解就是来映射服务器访问的路径.

1)此注解可加在方法上,是为此方法注册一个可以访问的名称(路径)

@RequestMapping("/demo")

```
<a href="${pageContext.request.contextPath}/demo.action">访问服务器</a>
```

此注解可以加在类上,相当于是包名(虚拟路径),区分不同类中相同的action的名称