

东软睿道内部公开

文件编号: D000-

SpringMVC框架技术

版本: 3.6.0

第1章 SpringMVC入门

东软睿道教育信息技术有限公司
(版权所有, 翻版必究)

Copyright © Neusoft Educational Information Technology Co., Ltd
All Rights Reserved



本章教学目标

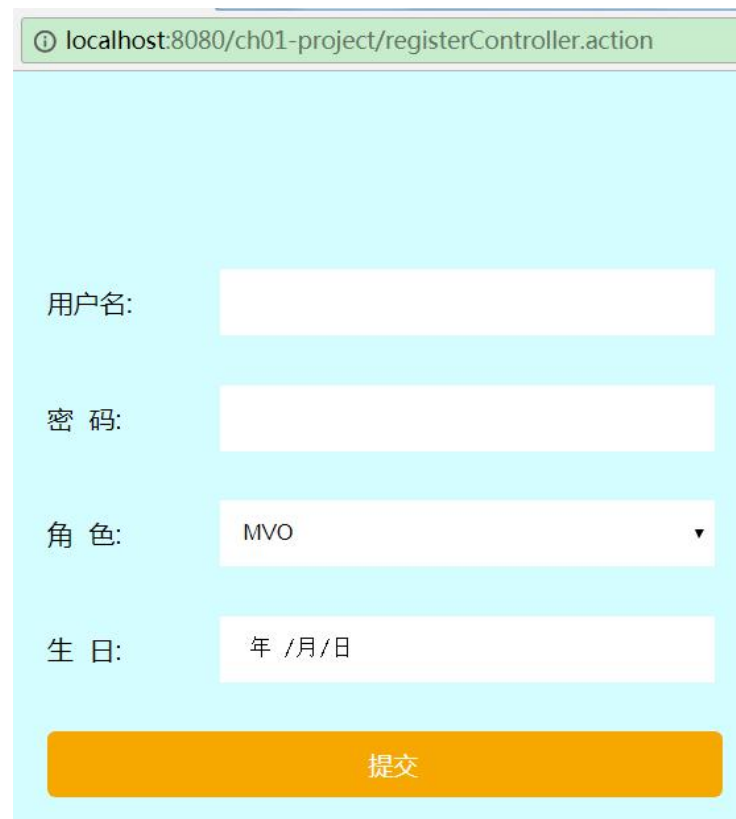
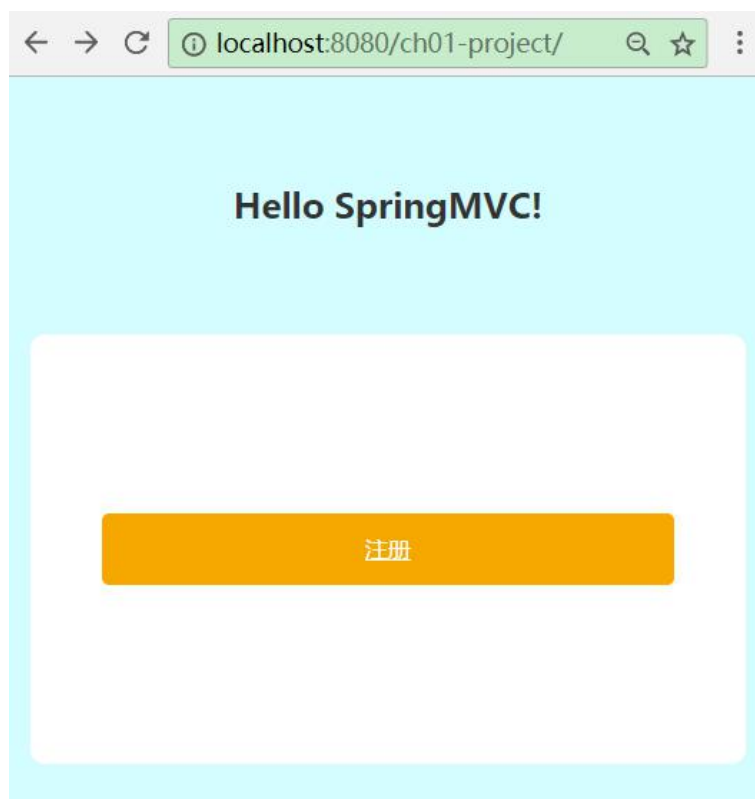
- ✓ 了解什么是SpringMVC;
- ✓ 了解MVC在B/S系统下的应用;
- ✓ 理解SpringMVC框架;
- ✓ 掌握SpringMVC组件;
- ✓ 掌握SpringMVC框架开发的环境搭建、开发步骤;

本章教学内容

节	知识点	掌握程度	难易程度	教学形式	对应在线微课
SpringMVC概述	什么是SpringMVC	了解	普通	线下	什么是SpringMVC
	MVC在B/S系统下的应用	了解	普通	线下	MVC在B/S系统下的应用
	SpringMVC框架	理解	难	线下	SpringMVC框架
	SpringMVC组件	掌握	普通	线下	SpringMVC组件
入门程序	环境搭建	掌握	普通	线下	环境搭建
	开发步骤	掌握	普通	线下	开发步骤
本章实战项目任务实现	实战项目任务实现	掌握	普通	线下	实战项目任务实现

本章实战项目任务

- ❖ 通过本章学习，完成《跨境电商系统》用户注册页面的显示。
- ❖ 项目运行效果如下



CONTENTS

目录

01

SpringMVC概述

02

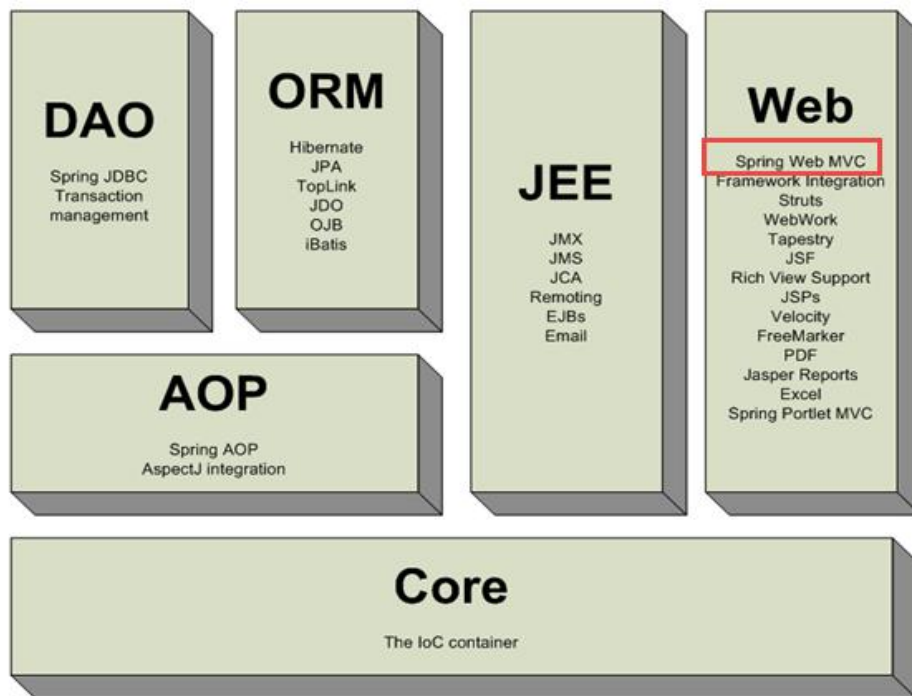
入门程序

03

本章实战项目任务实现

什么是SpringMVC

- ❖ Spring MVC属于SpringFrameWork的后续产品，已经融合在Spring Web Flow里面。
- ❖ Spring MVC是一个基于MVC的web框架，Spring MVC 分离了控制器、模型对象、过滤器以及处理程序对象的角色，这种分离让它们更容易进行定制。



什么是SpringMVC

❖ SpringMVC框架

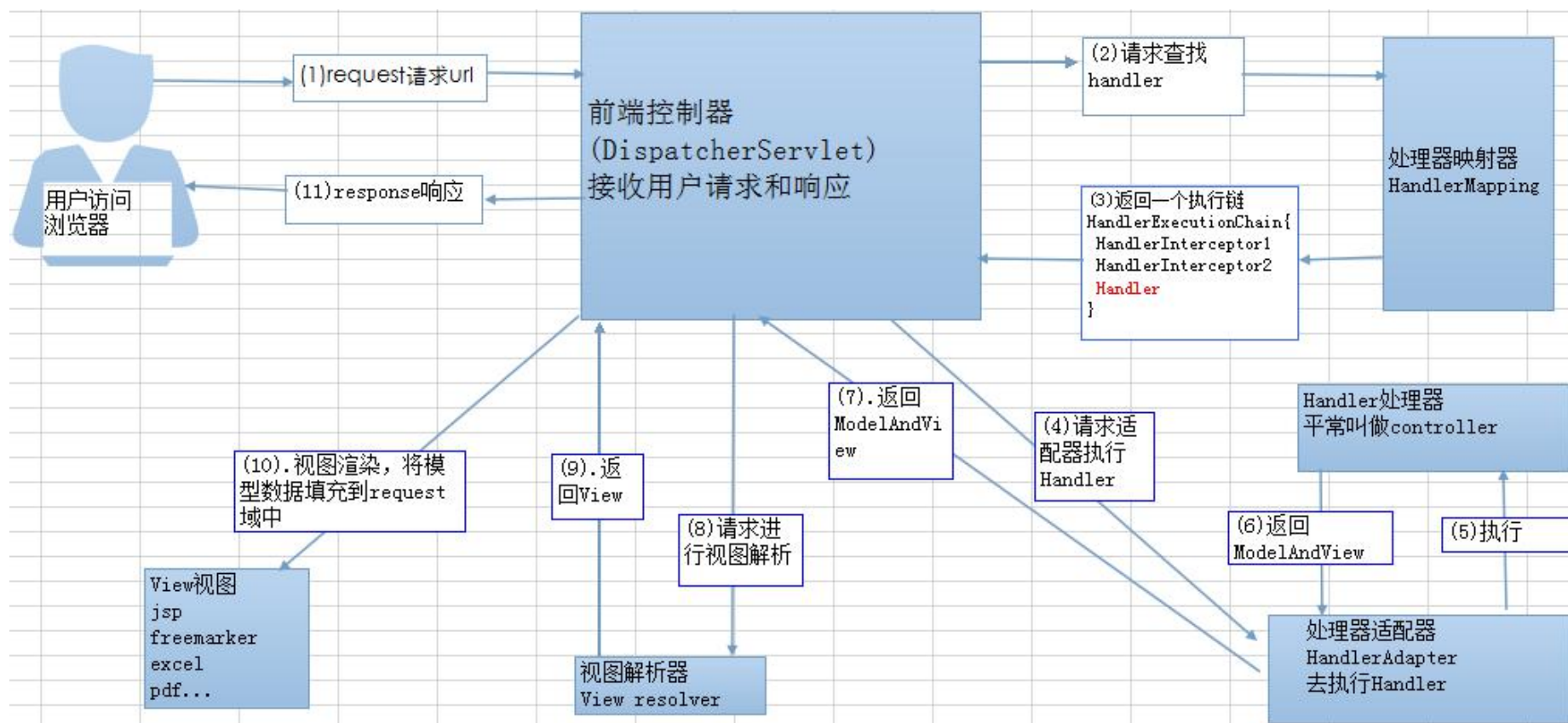
- ▶ Spring3.0 后全面超越 Struts2, 成为最优秀的 MVC 框架
- ▶ Spring MVC 通过一套 MVC 注解, 让 POJO 成为处理请求的控制器, 而无须实现任何接口
- ▶ 支持 REST 风格的 URL 请求
- ▶ 采用了松散耦合可插拔组件结构, 比其他 MVC 框架更具扩展性和灵活性
- ▶ 进行更简洁的Web层的开发;
- ▶ 天生与Spring框架集成(如IoC容器、AOP等);
- ▶ 能简单的进行Web层的单元测试;
- ▶ 支持灵活的URL到页面控制器的映射;
- ▶ 非常容易与其他视图技术集成, 如Velocity、FreeMarker等等,

MVC在B/S系统下的应用

- ❖ 用户发起request请求至控制器(Controller)
 - ▶ 控制接收用户请求的数据，委托给模型进行处理，如果只是页面的请求，那么可以直接在控制器处理器中分发到相应的view
- ❖ 控制器通过模型(Model)处理数据并得到处理结果
 - ▶ 模型通常是指业务逻辑
- ❖ 模型处理结果返回给控制器
- ❖ 控制器将模型数据在视图(View)中展示
 - ▶ web中模型无法将数据直接在视图上显示，需要通过控制器完成。如果在C/S应用中模型是可将数据在视图中展示的。
- ❖ 控制器将视图response响应给用户
 - ▶ 通过视图展示给用户要的数据或处理结果。

SpringMVC框架

❖ 框架处理流程图



SpringMVC框架

- ❖ 1、首先用户发送请求——>DispatcherServlet，前端控制器收到请求后自己不进行处理，而是委托给其他的解析器进行处理，作为统一访问点，进行全局的流程控制；
- ❖ 2、DispatcherServlet——>HandlerMapping，HandlerMapping将会把请求映射为HandlerExecutionChain对象（包含一个Handler处理器（页面控制器）对象、多个HandlerInterceptor拦截器）对象，通过这种策略模式，很容易添加新的映射策略；
- ❖ 3、DispatcherServlet——>HandlerAdapter，HandlerAdapter将会把处理器包装为适配器，从而支持多种类型的处理器，即适配器设计模式的应用，从而很容易支持很多类型的处理器；

SpringMVC框架

- ❖ 4、HandlerAdapter——>处理器，功能处理方法的调用，HandlerAdapter将会根据适配的结果调用真正的处理器的功能处理方法，完成功能处理；并返回一个ModelAndView对象（包含模型数据、逻辑视图名）；
- ❖ 5、DispatcherServlet——> ViewResolver， ViewResolver将把逻辑视图名解析为具体的View，通过这种策略模式，很容易更换其他视图技术；
- ❖ 6、DispatcherServlet——> View，View会根据传进来的Model模型数据进行渲染，此处的Model实际是一个Map数据结构，因此很容易支持其他视图技术；
- ❖ 7、由DispatcherServlet返回响应给用户，到此一个流程结束。

SpringMVC组件

- ❖ 前端控制器DispatcherServlet（不需要程序员开发）
 - ▶ 作用接收请求，响应结果，相当于转发器，中央处理器
 - ▶ 有了DispatcherServlet减少了其它组件之间的耦合度
- ❖ 处理器映射器HandlerMapping（不需要程序员开发）
 - ▶ 作用：根据请求的url查找Handler
- ❖ 处理器适配器HandlerAdapter
 - ▶ 作用：按照特定规则（HandlerAdapter要求的规则）去执行Handler
- ❖ 处理器Handler（需要程序员开发）
 - ▶ 注意：编写Handler时按照HandlerAdapter的要求去做，这样适配器才可以去正确执行Handler
- ❖ 视图解析器View resolver（不需要程序员开发）
 - ▶ 作用：进行视图解析，根据逻辑视图名解析成真正的视图（view）
- ❖ 视图View（需要程序员开发jsp）
 - ▶ View是一个接口，实现类支持不同的View类型（jsp、freemarker、pdf...）

课堂练习（5分钟）

- ❖ 1、什么是SpringMVC?
- ❖ 2、SpringMVC框架的工作流程?
- ❖ 3、SpringMVC组件有哪些?



CONTENTS

目录

01

SpringMVC概述

02

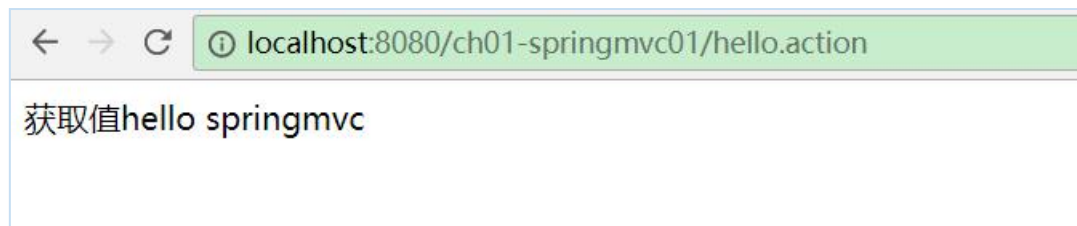
入门程序

03

本章实战项目任务实现

编写第一个springmvc程序

- ❖ Hello SpringMVC
- ❖ 示例：



- ▶ 全部代码参见：ch01-springmvc01工程

环境搭建

- ❖ java环境：
 - ▶ jdk1.8
 - ▶ Spring Tool Suite™ (STS)
- ❖ Maven环境
 - ▶ 3.0版本以上
- ❖ SpringMVC版本：
 - ▶ Spring4.0
 - ▶ 需要Spring4.0所有jar
- ❖ Web服务器：
 - ▶ Tomcat9



开发步骤

❖ 开发步骤

- ▶ 1. 创建maven web工程
- ▶ 2. pom.xml导入jar 包
- ▶ 3. resources下新增Spring基本配置文件：SpringMVC.xml
- ▶ 4. 在 web.xml 中配置 DispatcherServlet
- ▶ 5. 新建HelloController继承Controller，指定返回视图页面（本案例使用默认index.jsp）
- ▶ 6. SpringMVC.xml配置访问映射
- ▶ 7. 视图编写
- ▶ 8. 启动tomcat测试

开发步骤

❖ 1. 创建maven web工程:

- ▶ 在Eclipse中创建web工程
- ▶ File->New Maven Project
- ▶ 注意选择maven-archetype-webapp
- ▶ 创建完成后, 指定项目运行的容器: 右键工程→Build Path→Configure Build Path→Libraries→Add Library→选中Server Runtime, Next→选中Apache Tomcat 9, Finish (如果看不到tomcat, 先在Server里面配置Runtime Environments)

开发步骤

- ❖ 2. pom.xml 导入 jar 包：
 - ▶ properties: 统一管理 Spring 包版本;

```
ch01-springmvc01/pom.xml
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.neuedu</groupId>
5   <artifactId>ch01-springmvc01</artifactId>
6   <packaging>war</packaging>
7   <version>0.0.1-SNAPSHOT</version>
8   <name>ch01-springmvc01 Maven Webapp</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <org.springframework.version>4.0.3.RELEASE</org.springframework.version>
12  </properties>
```

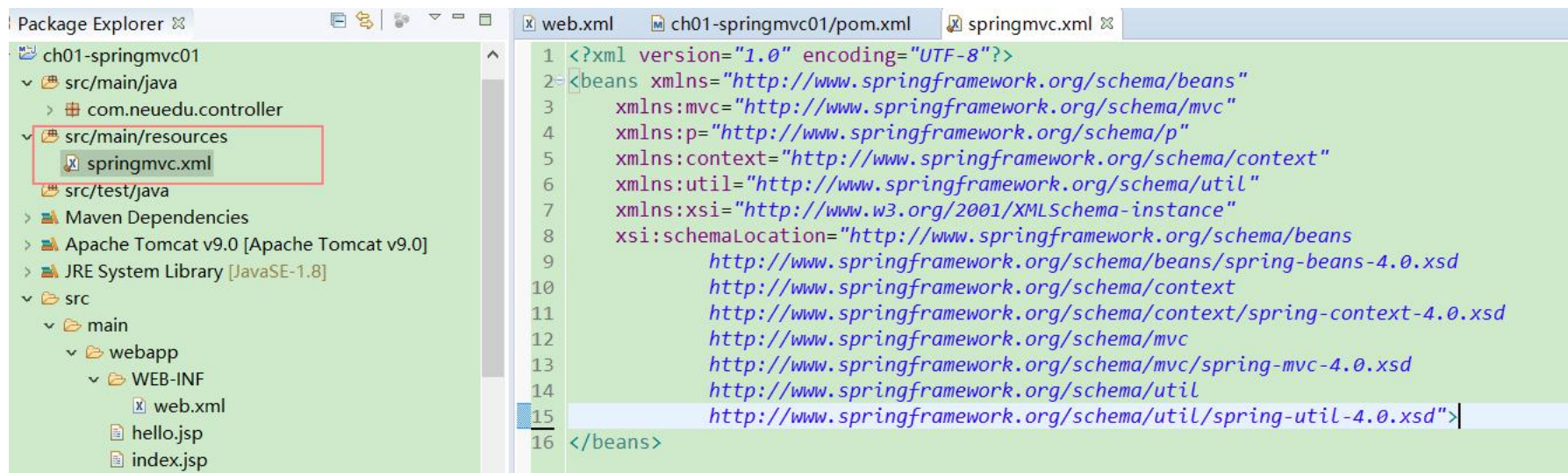
开发步骤

- ❖ 2. pom.xml 导入 jar 包：
 - ▶ dependencies 添加需要使用的 jar 包；

```
ch01-springmvc01/pom.xml
21- <dependency>
22-     <groupId>org.springframework</groupId>
23-     <artifactId>spring-aop</artifactId>
24-     <version>${org.springframework.version}</version>
25- </dependency>
26- <dependency>
27-     <groupId>org.springframework</groupId>
28-     <artifactId>spring-beans</artifactId>
29-     <version>${org.springframework.version}</version>
30- </dependency>
31- <dependency>
32-     <groupId>org.springframework</groupId>
33-     <artifactId>spring-context</artifactId>
34-     <version>${org.springframework.version}</version>
35- </dependency>
36- <dependency>
37-     <groupId>org.springframework</groupId>
38-     <artifactId>spring-core</artifactId>
39-     <version>${org.springframework.version}</version>
40- </dependency>
41- <dependency>
42-     <groupId>org.springframework</groupId>
43-     <artifactId>spring-web</artifactId>
44-     <version>${org.springframework.version}</version>
45- </dependency>
46- <dependency>
47-     <groupId>org.springframework</groupId>
48-     <artifactId>spring-webmvc</artifactId>
49-     <version>${org.springframework.version}</version>
50- </dependency>
```

开发步骤

❖ 3. resources下新增Spring基本配置文件：springmvc.xml



开发步骤

❖ 4. 在 web.xml 中配置 DispatcherServlet

```
web.xml  ch01-springmvc01/pom.xml
1  <!DOCTYPE web-app PUBLIC
2  "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
3  "http://java.sun.com/dtd/web-app_2_3.dtd" >
4
5  <web-app>
6    <display-name>Archetype Created Web Application</display-name>
7    <servlet>
8      <servlet-name>springmvc</servlet-name>
9      <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
10     <init-param>
11       <param-name>contextConfigLocation</param-name>
12       <param-value>classpath:springmvc.xml</param-value>
13     </init-param>
14
15   </servlet>
16   <servlet-mapping>
17     <servlet-name>springmvc</servlet-name>
18     <url-pattern>*.action</url-pattern>
19   </servlet-mapping>
20 </web-app>
21
```

开发步骤

❖ 5. 新建HelloController继承Controller:



开发步骤

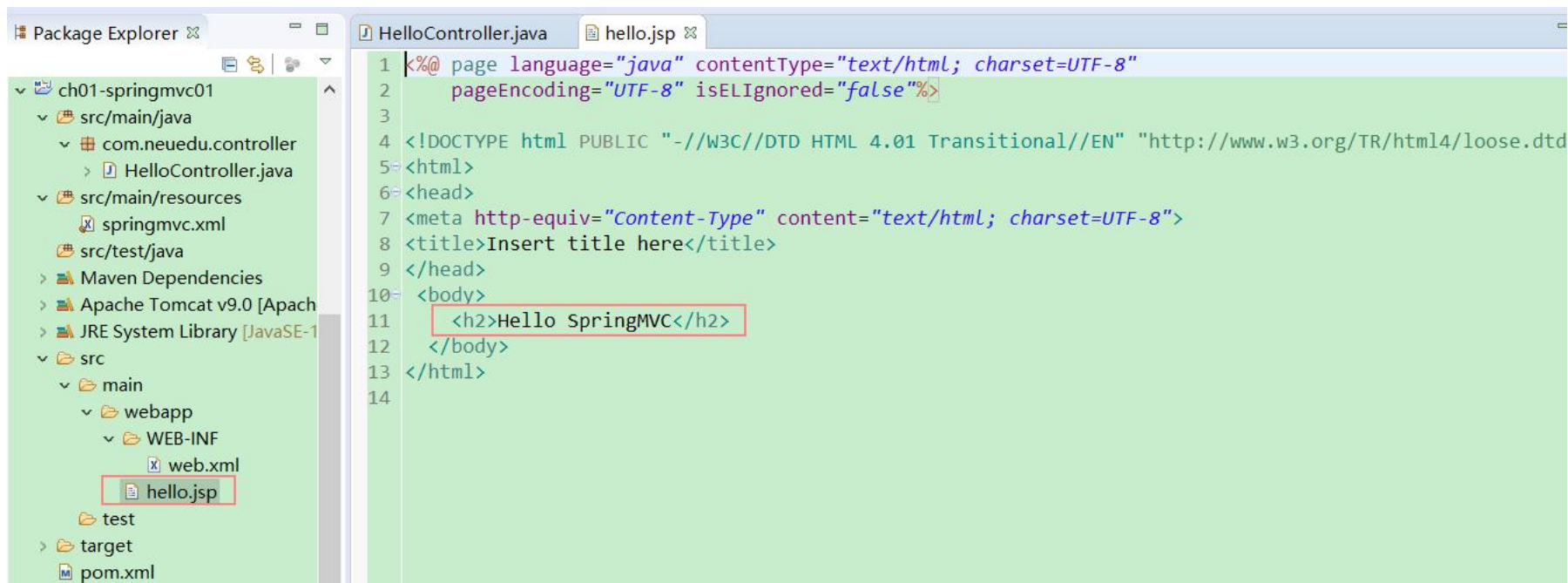
❖ 6. springmvc.xml 配置访问映射

```
web.xml  ch01-springmvc01/pom.xml  springmvc.xml  HelloController.java
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:mvc="http://www.springframework.org/schema/mvc"
4     xmlns:p="http://www.springframework.org/schema/p"
5     xmlns:context="http://www.springframework.org/schema/context"
6     xmlns:util="http://www.springframework.org/schema/util"
7     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8     xsi:schemaLocation="http://www.springframework.org/schema/beans
9         http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
10        http://www.springframework.org/schema/context
11        http://www.springframework.org/schema/context/spring-context-4.0.xsd
12        http://www.springframework.org/schema/mvc
13        http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd
14        http://www.springframework.org/schema/util
15        http://www.springframework.org/schema/util/spring-util-4.0.xsd">
16     <bean name="/hello.action" class="com.neuedu.controller.HelloController"/>
17
18
19 </beans>
```


开发步骤

❖ 7. 视图编写

▶ 编写hello.jsp

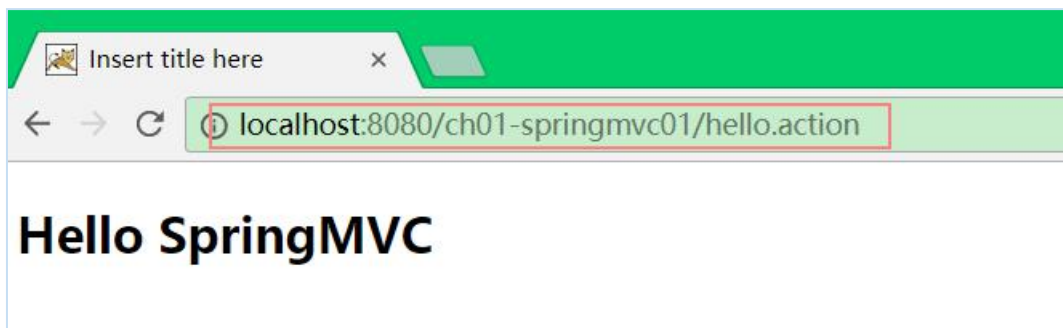


开发步骤

❖ 8. 启动tomcat测试:

<http://localhost:8080/ch01-springmvc01/hello.action>

- ▶ IP地址: localhost (本机)
- ▶ 端口: 8080
- ▶ 工程名称: ch01-springmvc01
- ▶ 请求映射路径: hello.action



课堂练习（30分钟）

❖ 练习

- ▶ 独立完成springmvc的环境搭建和hello springmvc程序编写
- ▶ 答案同课堂案例
- ▶ 全部代码参见：ch01-ktlx工程



CONTENTS

目录

01

SpringMVC概述

02

入门程序

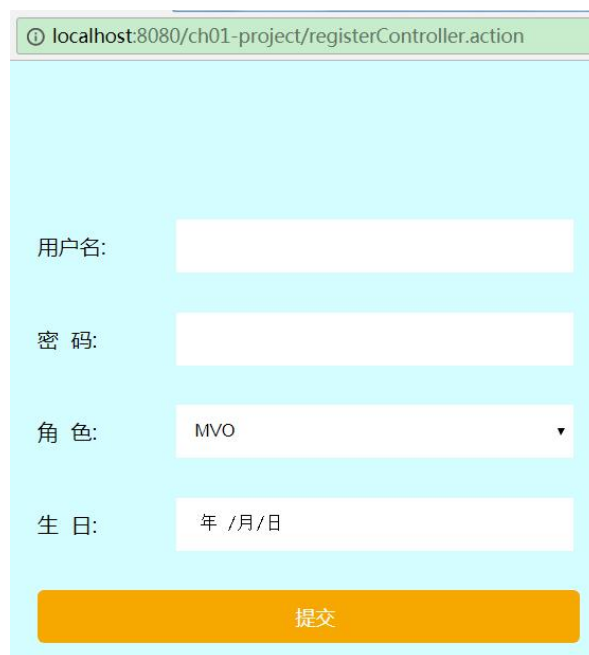
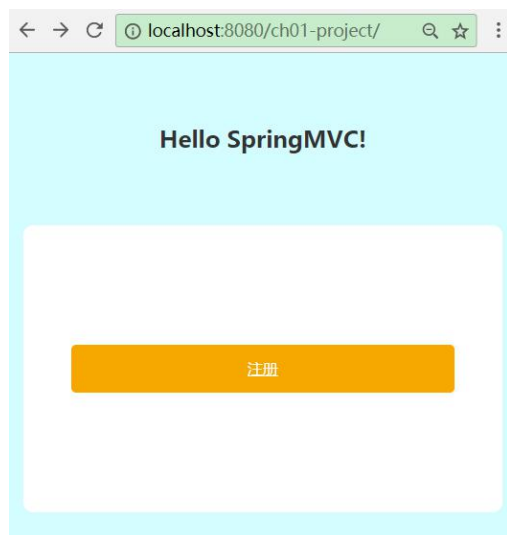
03

本章实战项目任务实现

本章实战项目任务实现

- ❖ 通过本章内容的学习，完成《跨境电商系统》用户注册页面的显示。
 - ▶ 搭建springmvc开发的环境
 - ▶ 编写工程代码ch01-project
 - ▶ 工程部署和运行<http://localhost:8080/ch01-project/>
 - ★ 首先显示index.jsp
 - ★ 单击index.jsp页面中的链接，访问registerController.action
 - ★ 在控制器registerController中设置注册页面的显示

❖ 运行效果如图：

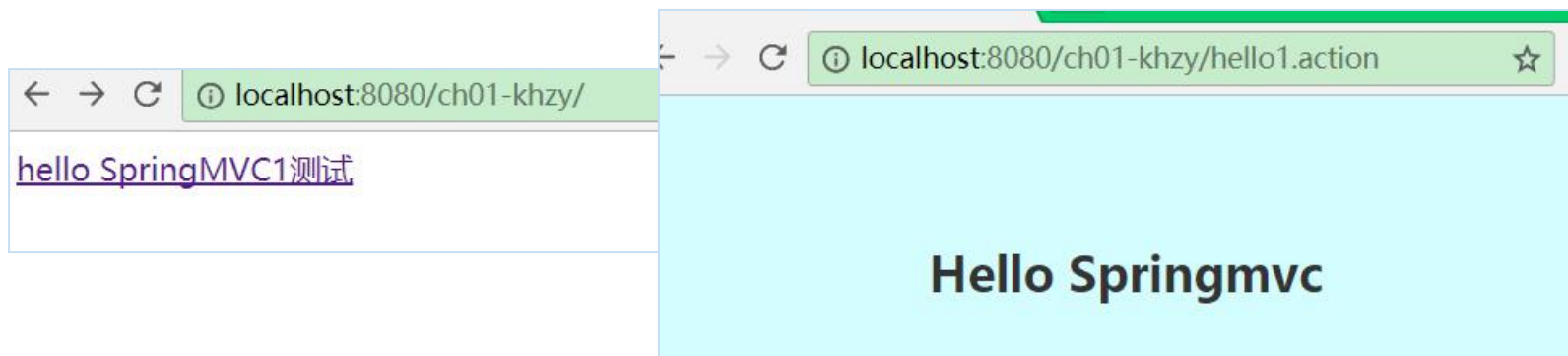


本章重点总结

- ❖ 了解什么是SpringMVC;
- ❖ 了解MVC在B/S系统下的应用;
- ❖ 理解SpringMVC框架;
- ❖ 掌握SpringMVC组件;
- ❖ 掌握SpringMVC框架开发的环境搭建、开发步骤;

课后作业【必做任务】

- ❖ 完成springmvc开发的环境搭建和Hello1.jsp页面内容的显示。
 - ▶ 搭建springmvc开发的环境
 - ▶ 编写工程代码ch01-khzy
 - ▶ 工程部署
 - ▶ 工程运行, <http://localhost:8080/ch01-khzy/>
 - ★ 首先显示index.jsp
 - ★ 单击index.jsp页面中的链接, 访问hello1.action
 - ★ 在控制器中设置hello1.jsp页面的显示



课后作业【选做任务】

- ❖ 完成springmvc开发的环境搭建和Hello.jsp页面内容的显示。
 - ▶ 搭建springmvc开发的环境
 - ▶ 编写工程代码ch01-khzy
 - ▶ 工程部署
 - ▶ 工程运行, <http://localhost:8080/ch01-khzy/>
 - ★ 首先显示index.jsp
 - ★ 单击index.jsp页面中的链接, 访问hello.action
 - ★ hello.jsp页面中要显示的内容在控制器中进行设置



课后作业【线上任务】

❖ 线上任务

- ▶ 安排学员线上学习任务（安排学员到睿道实训平台进行复习和预习的任务，主要是进行微课的学习）

