

## 实验 6: SpringMVC 开发环境搭建

- 一、 实验目的: 搭建 SpringMVC 开发环境
- 二、 预习要求: 提前预习 Spring MVC 框架, 掌握 SpringMVC 基本特点
- 三、 实验内容:
  1. 搭建 SpringMVC 开发环境, 实现使用浏览器显示商品列表
  2. 根据上学期的登录控制例子, 将 login.jsp, welcome.jsp, 以及 error.jsp 页面放到 /WEB-INF/jsp/ 目录中, 写出控制器实现登陆控制访问。
- 四、 实验方法和步骤:

### ①创建工程

创建工程, 导入相关 jar 包

### ②创建 springmvc.xml

创建 SpringMVC 的核心配置文件

SpringMVC 本身就是 Spring 的一个子项目, 对 Spring 兼容性很好, 不需要做很多配置。这里只配置一个 Controller 扫描就可以了, 让 Spring 对页面控制层 Controller 进行管理。

创建 springmvc.xml

```
<?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:p="http://www.springframework.org/schema/p"
       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-4.0.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.0.xsd"
>
```

```

    <!-- 配置 controller 扫描包 -->
    <context:component-scan
base-package="cn.itcast.springmvc.controller" />

</beans>

```

### ③配置前端控制器

配置 SpringMVC 的前端控制器 DispatcherServlet

在 web.xml 中配置

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
    id="WebApp_ID" version="2.5">
    <display-name>springmvc-first</display-name>
    <welcome-file-list>
        <welcome-file>index.html</welcome-file>
        <welcome-file>index.htm</welcome-file>
        <welcome-file>index.jsp</welcome-file>
        <welcome-file>default.html</welcome-file>
        <welcome-file>default.htm</welcome-file>
        <welcome-file>default.jsp</welcome-file>
    </welcome-file-list>

    <!-- 配置 SpringMVC 前端控制器 -->
    <servlet>
        <servlet-name>springmvc-first</servlet-name>

        <servlet-class>org.springframework.web.servlet.DispatcherServlet<
/servlet-class>
        <!-- 指定 SpringMVC 配置文件 -->
        <init-param>
            <param-name>contextConfigLocation</param-name>
            <param-value>classpath:springmvc.xml</param-value>
        </init-param>
    </servlet>

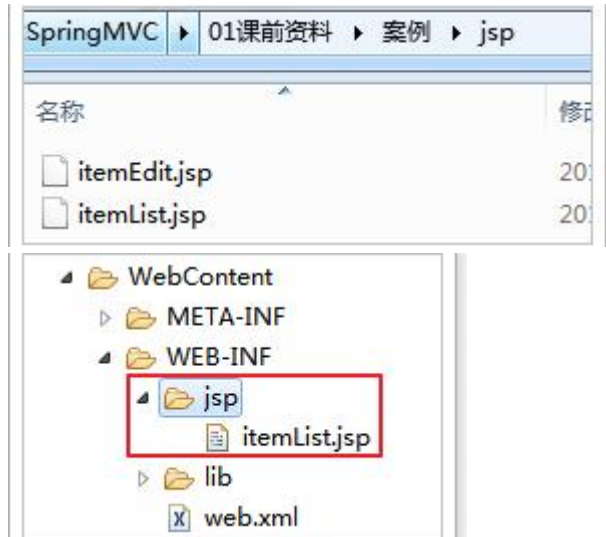
    <servlet-mapping>
        <servlet-name>springmvc-first</servlet-name>
        <!-- 设置所有以 action 结尾的请求进入 SpringMVC -->

```

```
<url-pattern>*.action</url-pattern>
</servlet-mapping>
</web-app>
```

## ④加入 jsp 页面

把参考资料中的 itemList.jsp 复制到工程的/WEB-INF/jsp 目录下。



## ⑤实现显示商品列表页

### (1)创建 pojo

分析页面，查看页面需要的数据



创建商品 pojo

```
public class Item {
    // 商品 id
    private int id;
    // 商品名称
    private String name;
    // 商品价格
```

```
private double price;
// 商品创建时间
private Date createtime;
// 商品描述
private String detail;
```

创建带参数的构造器

```
set/get。。。
}
```

## (2)创建 ItemController

ItemController 是一个普通的 java 类，不需要实现任何接口。

需要在类上添加@Controller 注解，把 Controller 交由 Spring 管理

在方法上面添加@RequestMapping 注解，里面指定请求的 url。其中“.action”可以加也可以不加。

```
@Controller
```

```
public class ItemController {
```

```
    // @RequestMapping: 里面放的是请求的 url，和用户请求的 url 进行匹配
    // action 可以写也可以不写
    @RequestMapping("/itemList.action")
    public ModelAndView queryItemList() {
        // 创建页面需要显示的商品数据
        List<Item> list = new ArrayList<>();
        list.add(new Item(1, "1 华为 荣耀 8", 2399, new Date(), "质量好! 1"));
        list.add(new Item(2, "2 华为 荣耀 8", 2399, new Date(), "质量好! 2"));
        list.add(new Item(3, "3 华为 荣耀 8", 2399, new Date(), "质量好! 3"));
        list.add(new Item(4, "4 华为 荣耀 8", 2399, new Date(), "质量好! 4"));
        list.add(new Item(5, "5 华为 荣耀 8", 2399, new Date(), "质量好! 5"));
        list.add(new Item(6, "6 华为 荣耀 8", 2399, new Date(), "质量好! 6"));

        // 创建 ModelAndView，用来存放数据和视图
        ModelAndView modelAndView = new ModelAndView();
        // 设置数据到模型中
        modelAndView.addObject("itemList", list);
        // 设置视图 jsp，需要设置视图的物理地址
        modelAndView.setViewName("/WEB-INF/jsp/itemList.jsp");

        return modelAndView;
    }
}
```

**2.** 根据上学期的登录控制例子, 将 login.jsp, welcome.jsp, 以及 error.jsp 页面放到/WEB-INF/jsp/目录中, 写出控制器实现登陆控制访问。

代码自行完成。

### 五、 思考题:

Spring MVC 框架的作用 ?

### 六、 实验作业要求:

- (一) 实验目的:
- (二) 实验内容:
- (三) 实验结果: 可以是运行结果截图或其他形式的结果展示
- (四) 问题及解决: 实验中遇到的问题及解决方法。

回答思考题提出的问题,