# Day17\_SpringMVC

## SpringMVC简介

### 概念

SpringMVC也是一个基于MVC模式,用来解决Web层功能的轻量级web框架,在功能上类似于Struts2.

SpringMVC本身就是Spring框架的一个子模块.属于Spring中负责web层的子框架.

### SpringMVC优点

1. .实现web层功能更加简单;
2. .SpringMVC本身就是Spring的一部分,所以在和Spring进行整合的时候,可以做到无缝的集成.
3. .SpringMVC中从url到页面资源的映射实现起来更加灵活;
4. .SpringMVC提供了一套比较灵活的数据验证,格式化和数据绑定的功能;
5. .支持restful风格的url地址;

## SpringMVC与Struts2框架的区别

### 1.框架机制

Struts2框架是基于过滤器的,核心过滤器是StrutsPrepareAndExecuteFilter.

SpringMVC框架是基于Servlet的,核心Servlet是DispatcherServlet.

### 2.拦截机制

Struts2框架是一种类级别的拦截,每次访问请求,都会创建一个Action对象.因为是基于类级别的拦截,所以可能存在线程安全问题.

SpringMVC是一种基于方法的拦截.每个url地址请求的是一个不同的方法,也就不存在线程安全的问题.

### 性能问题

Struts2的性能不如SpringMVC,因为过滤器的生命周期比Servlet长的多,也就是说过滤器在后台运行时间长.

而且在开发性能上,SpringMVC也比Struts2优良,SpringMVC可以做到零配置实现与Spring的整合.

## SpringMVC核心API

### 一个中心,DispatcherServlet

前端控制器.DispatcherServlet

### 三个基本组件

处理器映射器:XXXHanlerMapping

处理器适配器:XXXHandlerAdapter

视图解析器:InternalResourceViewResolver

### **3.处理器(控制器**),Handler/Controller

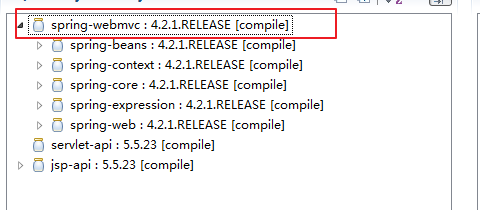
需要程序员自己实现,类似于Struts中的Action.

### 4.视图,View

jsp/html.

## SpringMVC基本使用

### 1.添加依赖包



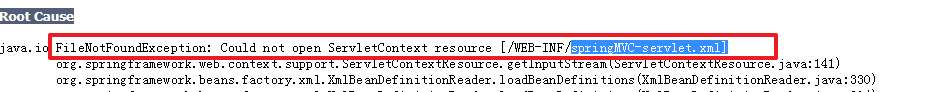
### 2.在web.xml文件中配置一个中心,前端控制器



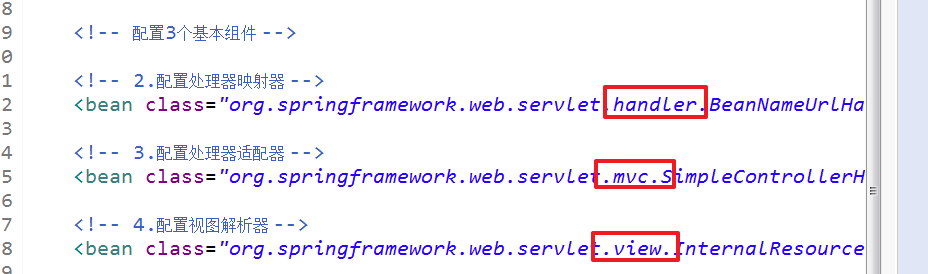
注意:

默认情况下,spring配置文件的位置和名称!

默认情况下,如果spring配置文件不正确,可能会产生如下异常!



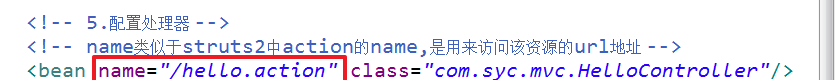
### 3.创建spring的配置文件,配置3个基本组件



### 4.创建控制器类

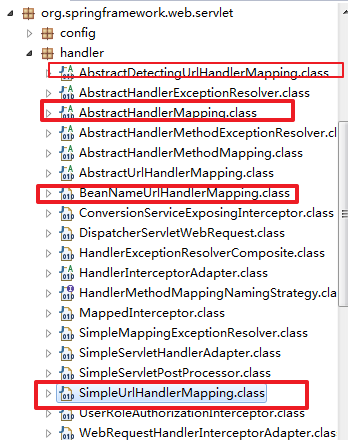


### 5.在spring配置文件中,配置该控制器

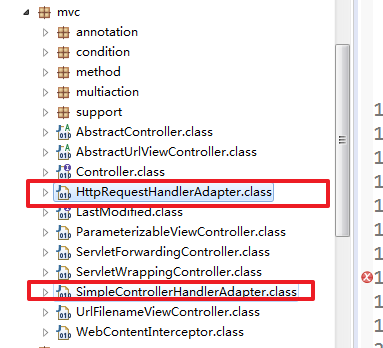


## 五.3个基本组件所在包

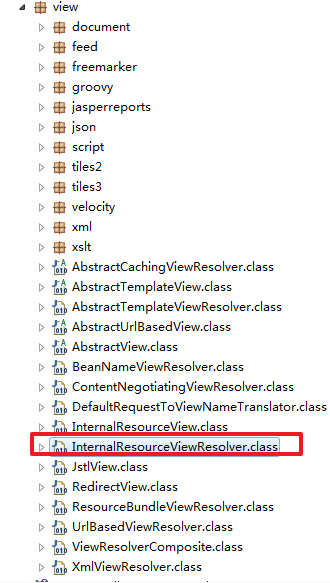
### 1.处理器映射器所在的包



### 2.处理器适配器所在的包



### 3.视图解析器所在的包



## 六.创建控制器

### 1.创建Handler类



注意:

必须在spring的配置文件中,配置对应的处理器适配器.

否则会出现以下异常!



### 2.配置HttpRequestHandlerAdapter

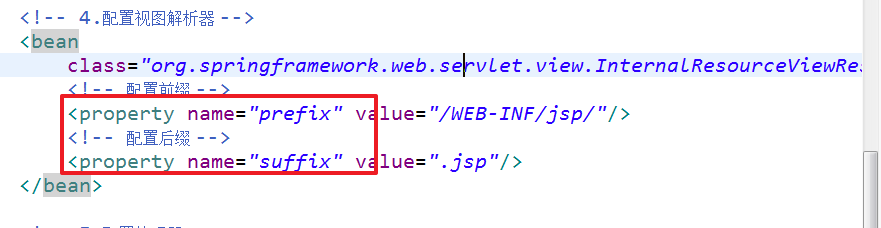


## 七.同时配置多个处理器映射器



## 八.逻辑视图名

### 1.配置视图解析器

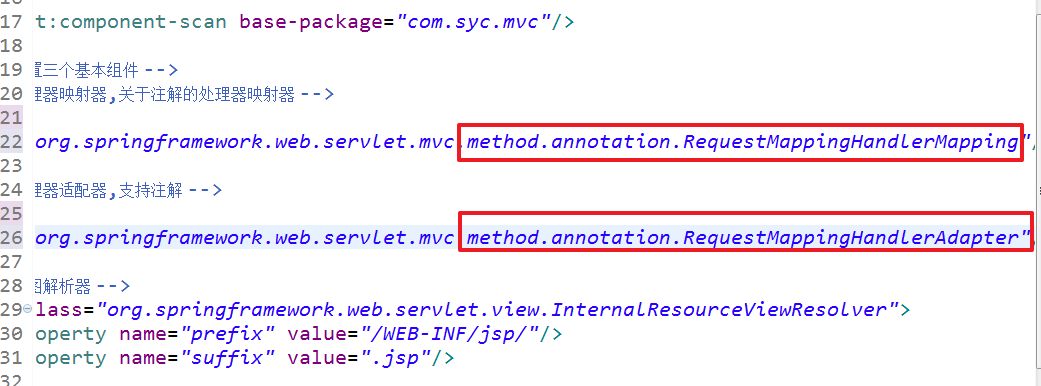


### 2.代码中设置逻辑视图名

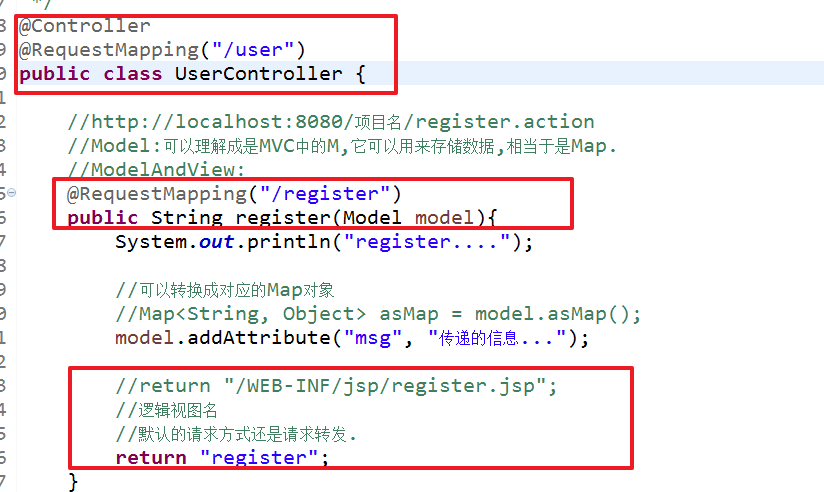


## 九.注解方式来实现控制器

### 1.配置三个基本组件

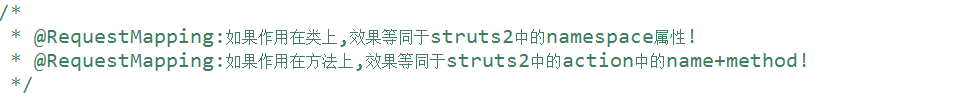


### 2.创建控制器类

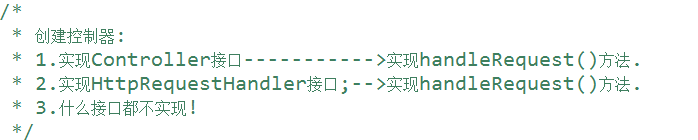


#### 注意:

##### @RequestMapping作用在类和方法上的区别:



##### 创建控制器/处理器类的3种方式:



## 十.SpingMVC的运行原理(重点)

### 1.第一步:

Tomcat服务器启动之后,DispatcherServlet就可以工作了.

假如此时用户在浏览器上输入了<http://localhost:8080/Day17_SpringMVC03/user/login.action>,

相当于发起了一个url请求.那么此时该请求就会先传送给DispatcherServlet这个调度中心,用户就在等待着一个response响应.

### 2.第二步:

DispatcherServlet调度中心拿到该request请求,在request请求中有一个url地址,调度中心就会将该url地址发送给处理器映射器,让它根据该url地址去寻找对应的处理器/控制器.

### 3.第三步:

处理器映射器接收到该url地址,然后内部经过层层处理,它利用HandlerChain进行处理,去找对应的处理器.如果找不到对应的处理器,会把找不到的结果告诉DispatcherServlet调度中心,此时就会报404异常了.

如果找到了对应的处理器,处理器映射器就将该处理器返回给调度中心.

### 4.第四步:

调度中心拿到了该处理器,此时调度中心会把该处理器交给处理器适配器进行处理.处理器适配器会将该处理器经过内部一些列的处理之后,把处理的结果封装成一个ModelAndView对象,并且把ModelAndView对象返回给调度中心.

### 5.第五步:

调度中心此时接收到了ModelAndView对象,调度中心就让视图解析器来处理这个ModelAndView.

视图解析器把这个ModelAndView拆分成两部分,分别Model和View.并且最后会把Model和View再次返回给调度中心.

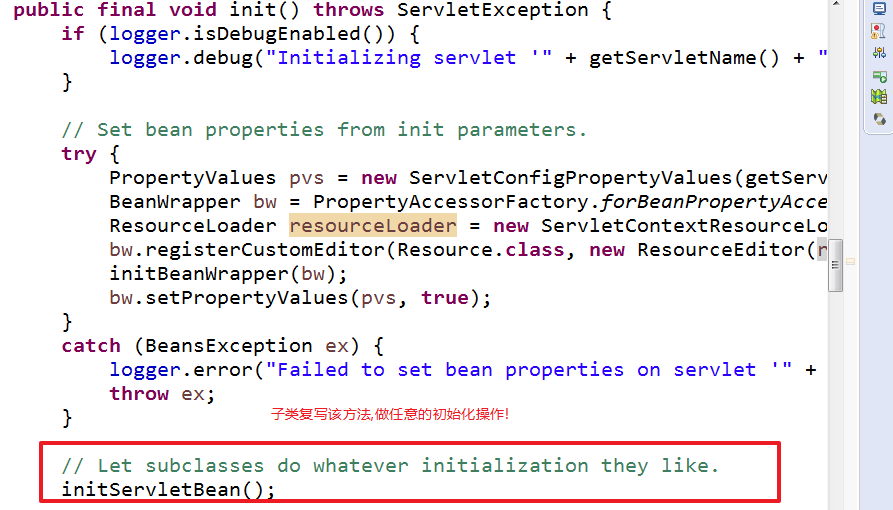
### 6.第六步:

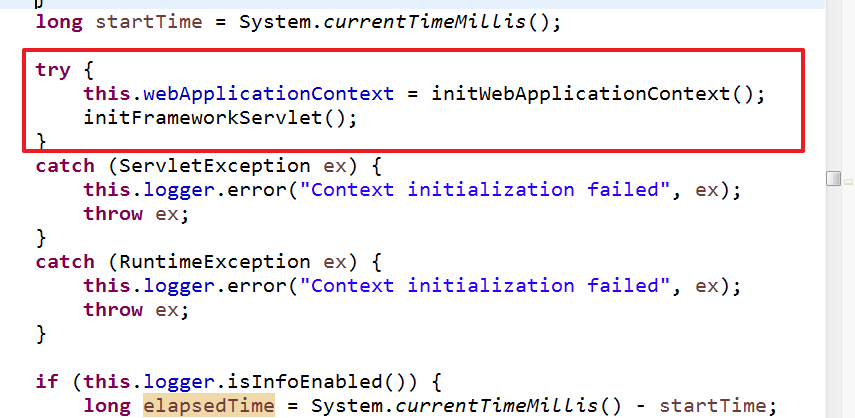
调度中心拿到了Model和View,它内部就会对View进行渲染,也就是将Model里的数据填充到View里面.

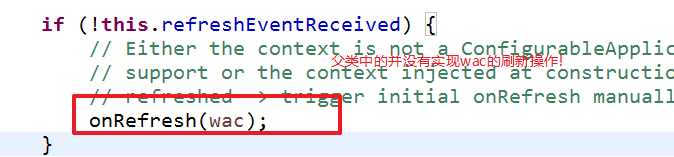
然后把具有数据的View,封装到response对象.

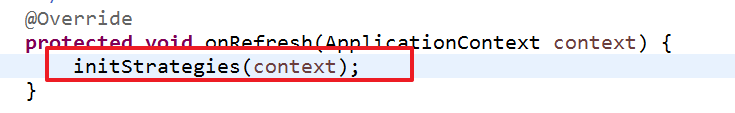
用户接收到response对象,得到期望结果.

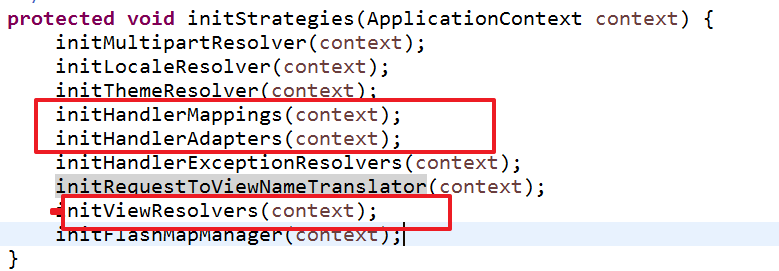
## 十一.SpringMVC源码分析(重点)

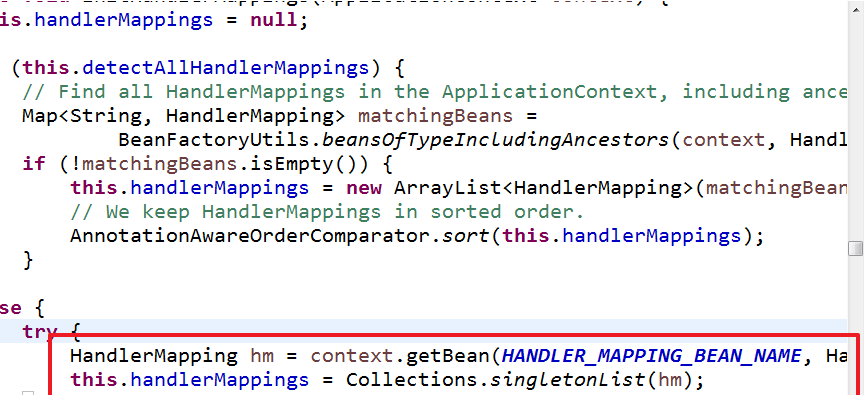


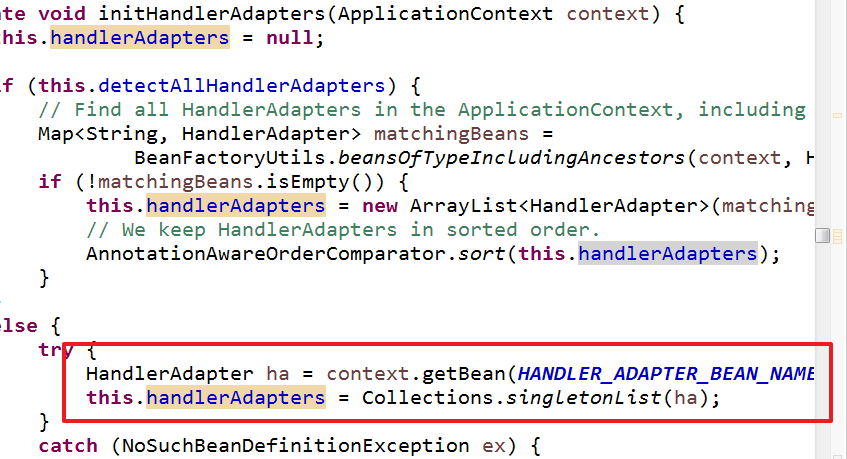


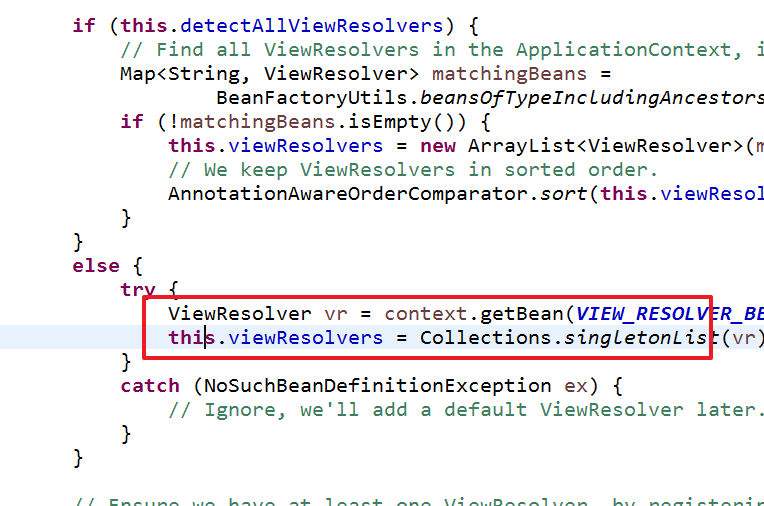


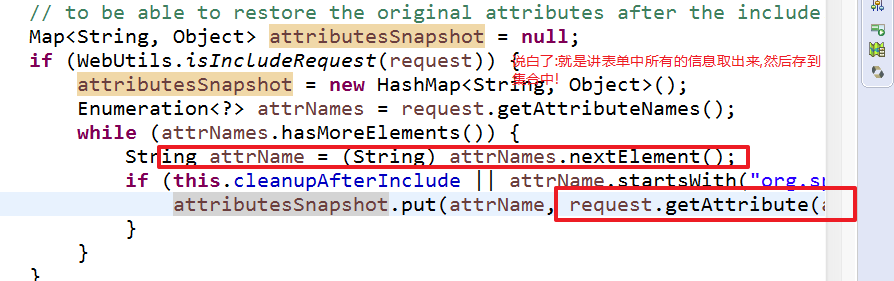


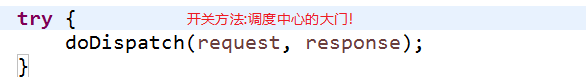


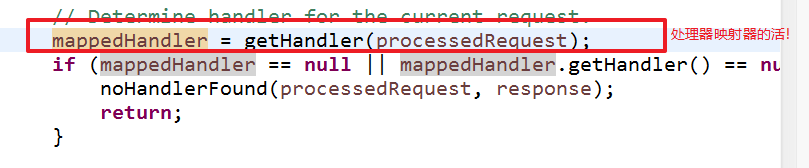


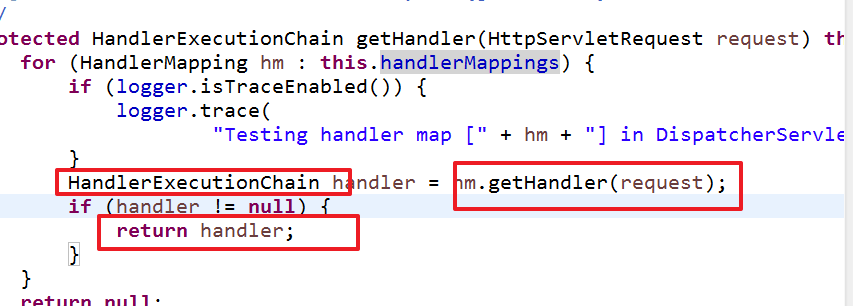






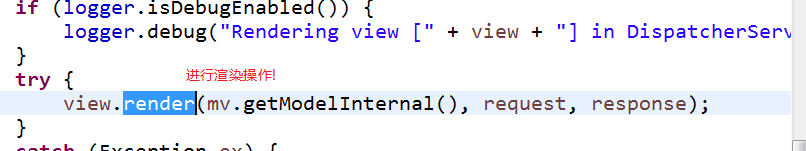












作业:

