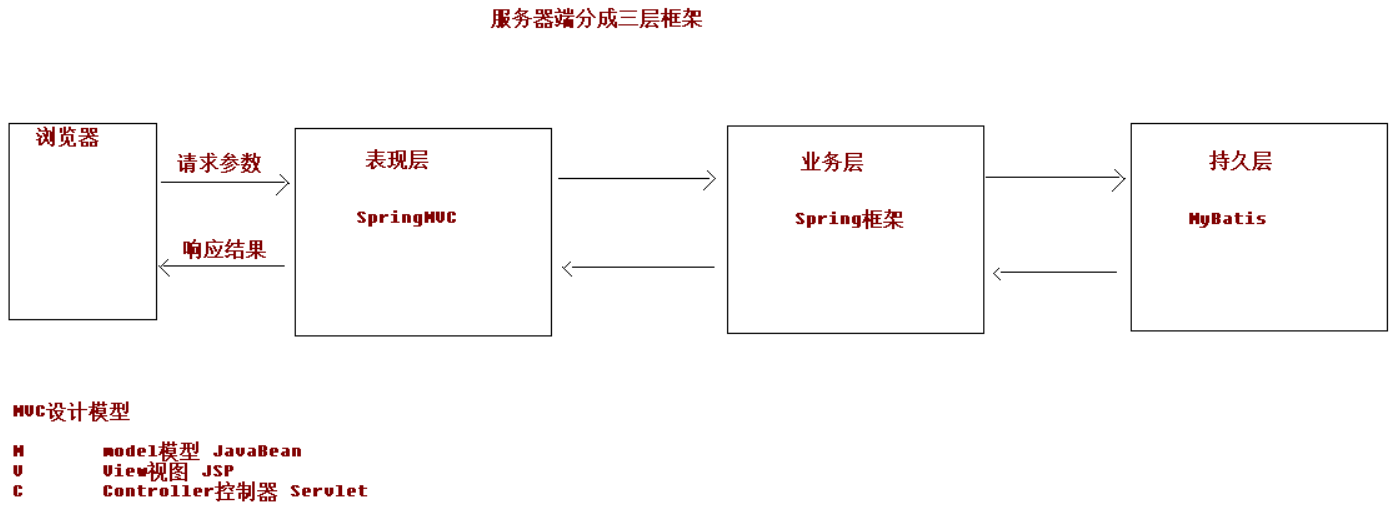
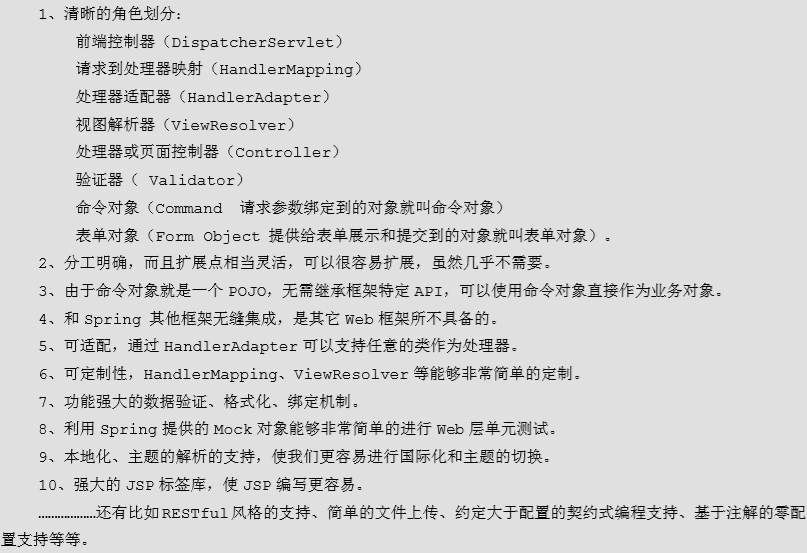
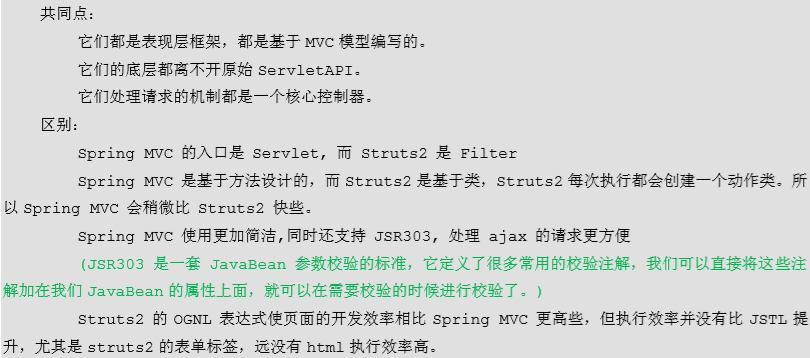
# SpringMVC介绍



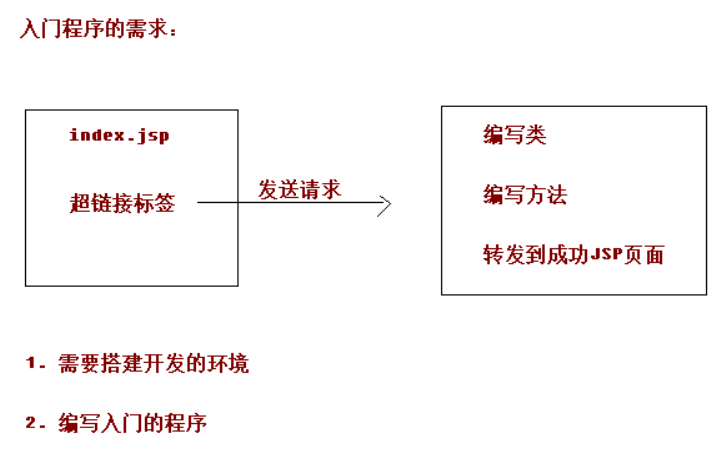
## SpringMVC的优势



## SpringMVC和Struts2区别



# 入门案例

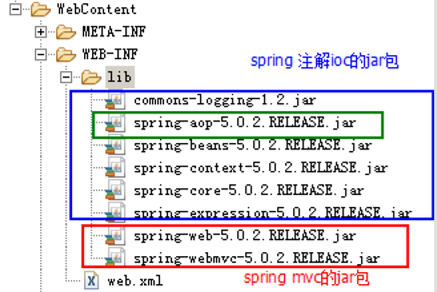


## 搭建开发环境

Based on SpringMVC\_day01: Link

### 创建带有web.xml的web project

### 引入jar包



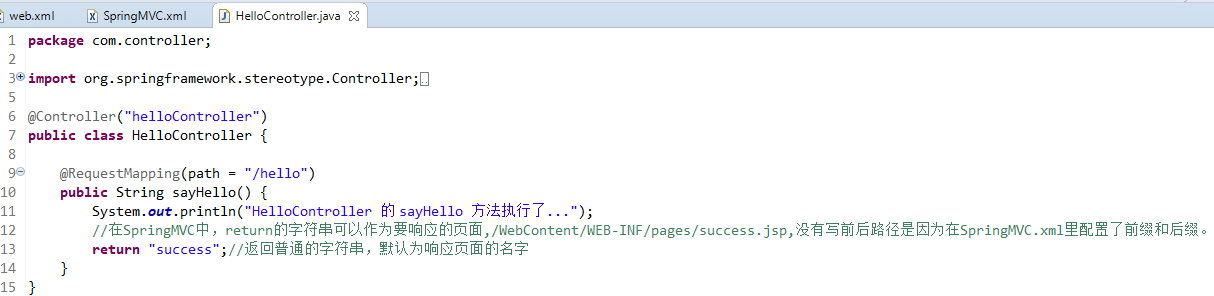
### 配置前端控制器



### 配置SpringMVC的xml配置文件



### Controller写法

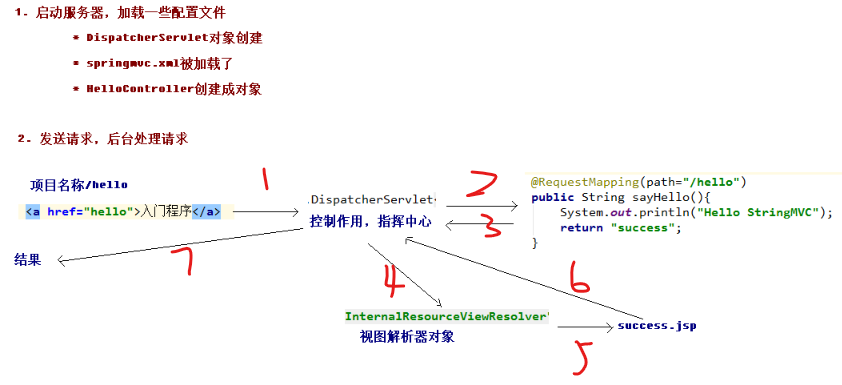


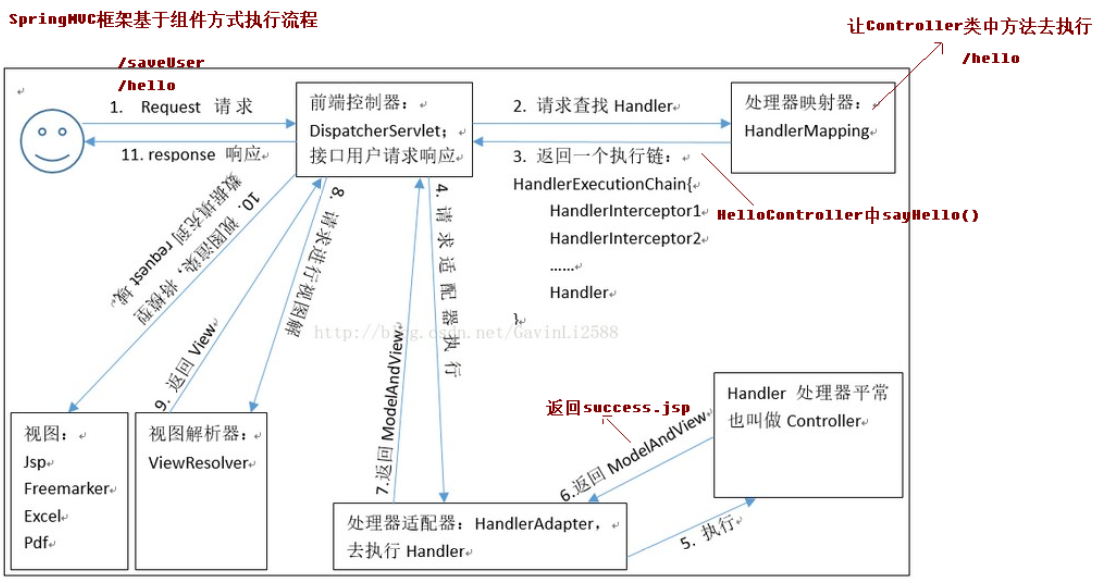
### 测试

<a href=*"hello"*>Test</a>

The “success.jsp” located at WebContent\WEB-INF\pages\success.jsp

### 流程总结





# RequestMapping 注解

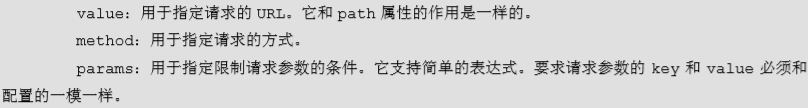
## RequestMapping使用位置

RequestMapping可以写在类和方法上。



<a href=*"helloController/testRequestMapping"*>RequestMapping注解</a>

## RequestMapping注解属性



### Value属性

用于指定请求的 URL。它和 path 属性的作用是一样的。如果只有一个属性，则value和path可以省略不写。

@RequestMapping("/testRequestMapping")

### Method属性

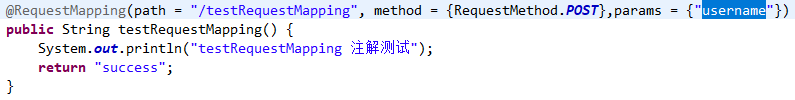
用于指定请求的方式。下例是只接受POST提交方式的写法，其它提交方式会报错。



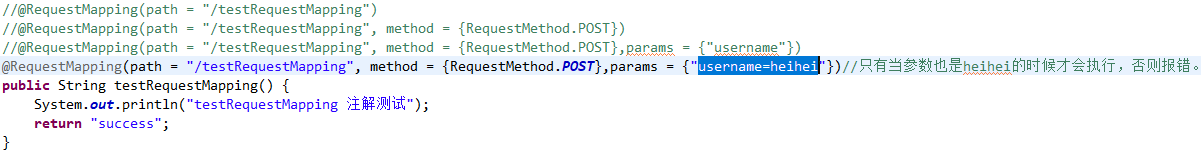


### params属性

用于指定限制请求参数的条件。它支持简单的表达式。要求请求参数的 key 和 value 必须和 配置的一模一样。

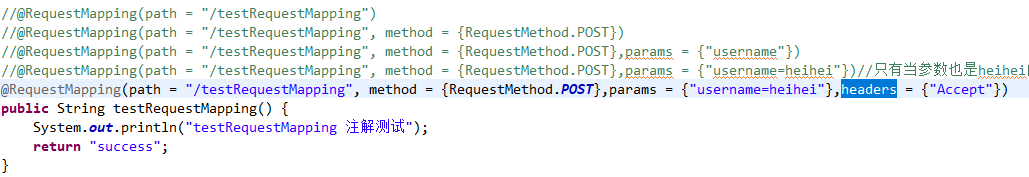






### headers属性

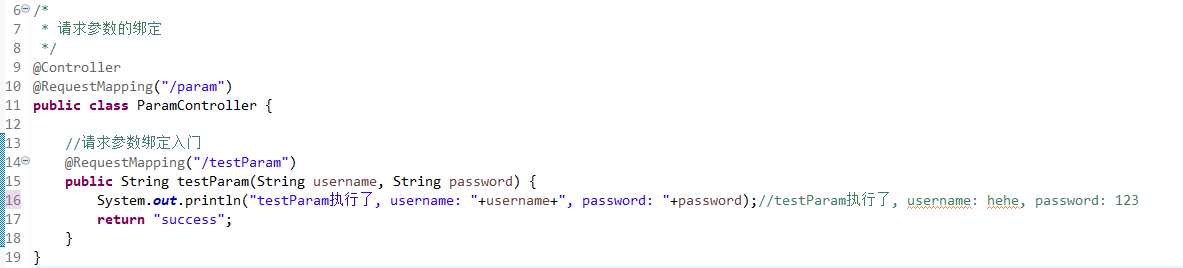
用于指定限制请求消息头的条件。如果不包含请求头，也会报错。



# 请求参数的绑定

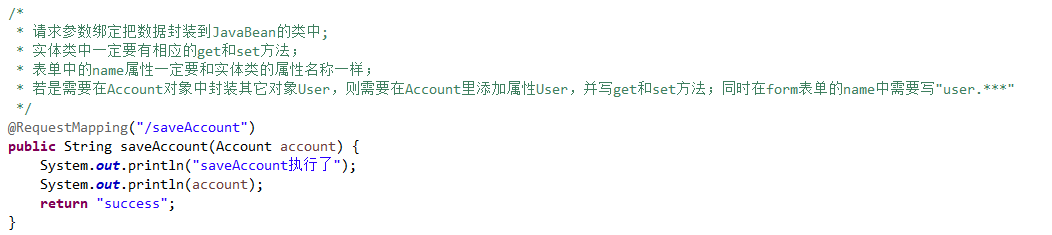
## 封装零散属性

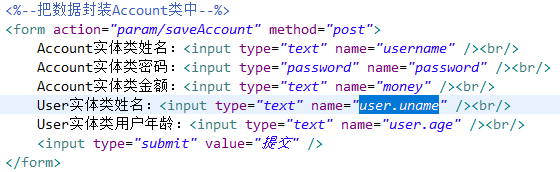




## 把请求参数封装到实体类中

SpringMVC会自动进行数据类型转换，并将属性封装到实体类中。



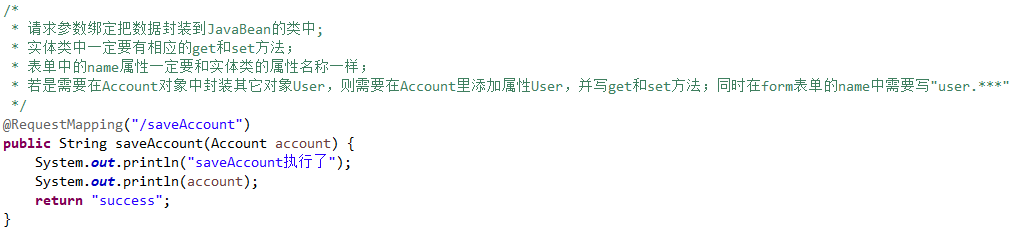


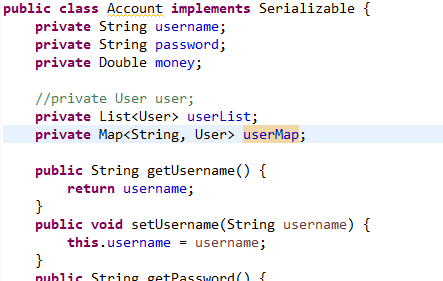
## 利用过滤器解决中文乱码



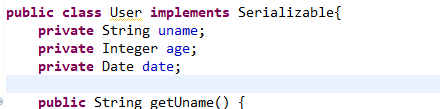
## 把请求参数封装到List和Map中

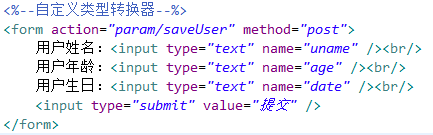




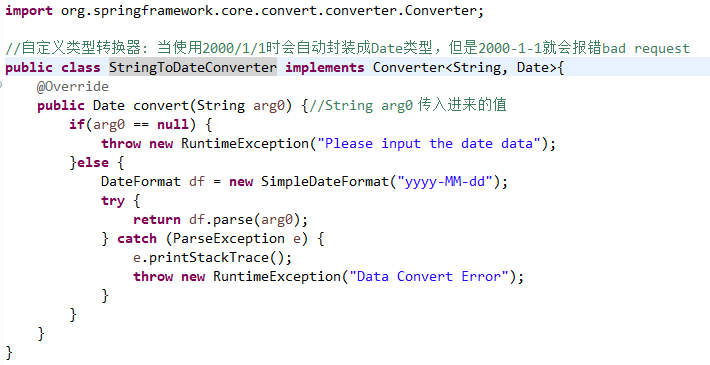


## 自定义类型转换器





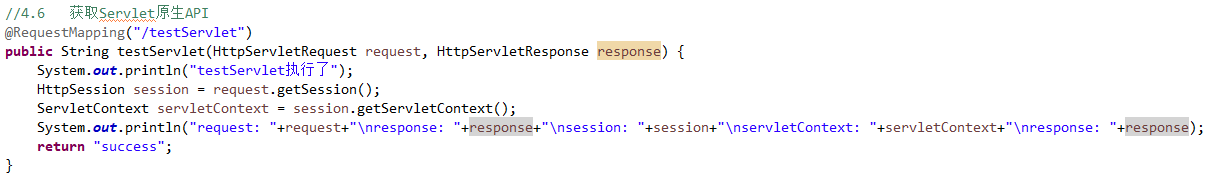






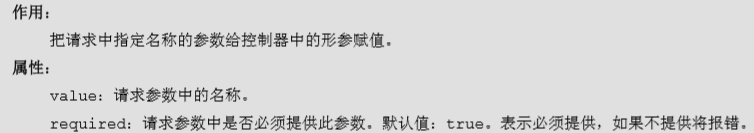
## 获取Servlet原生API





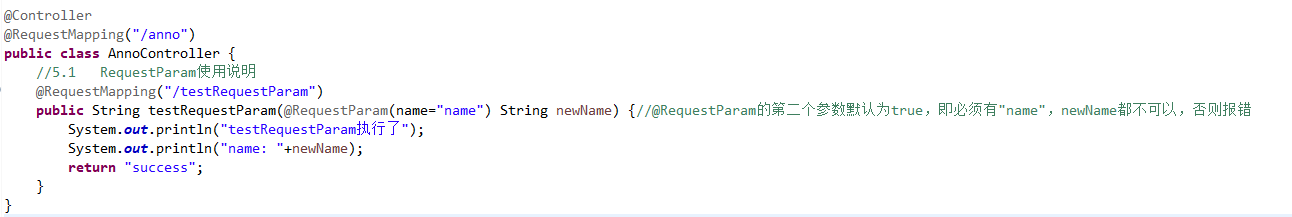
# 常用注解

## RequestParam注解使用说明

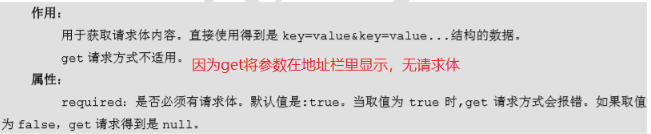


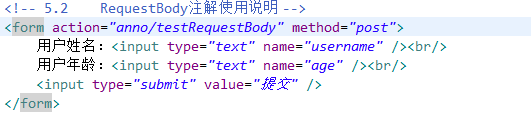
比如零散属性封装的时候形参必须和form表单的name相同，才会自动赋值。当不一样时，可以使用RequestParam来解决。

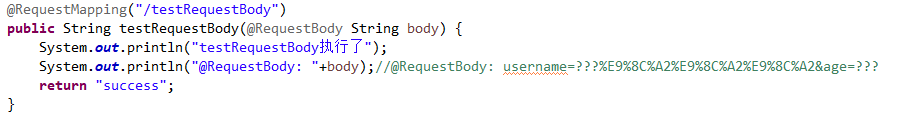




## RequestBody注解使用说明

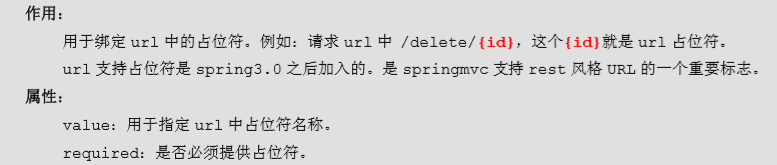




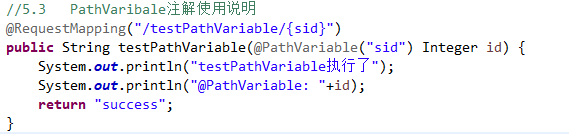


## PathVaribale注解使用说明

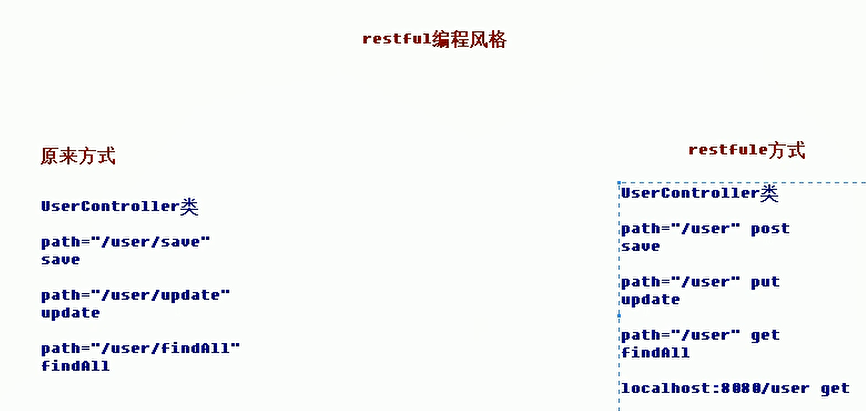
### PathVaribale使用说明

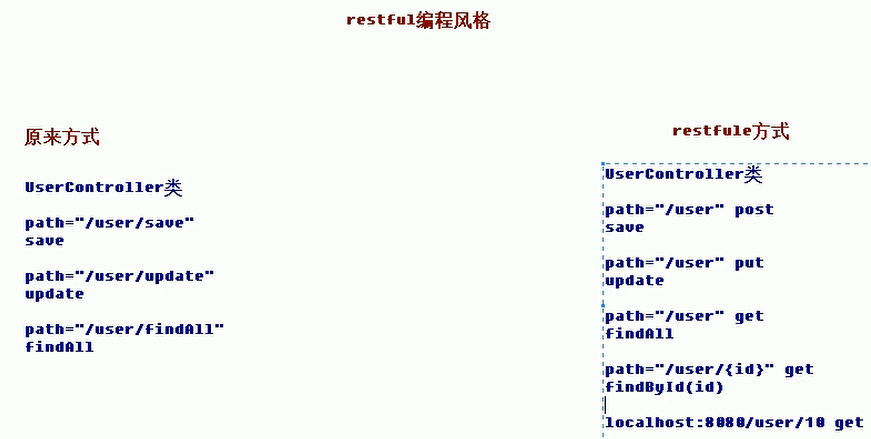




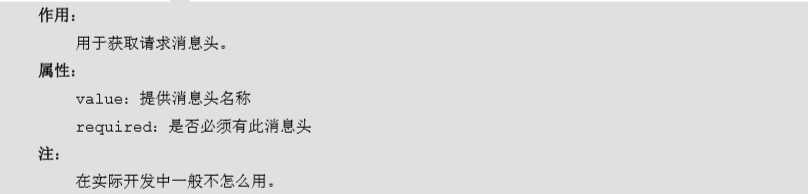


### Restful风格

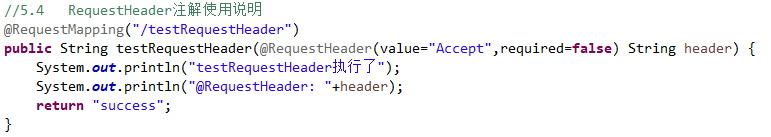




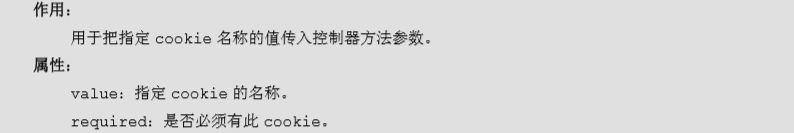
## RequestHeader注解使用说明



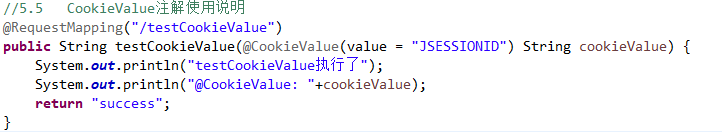




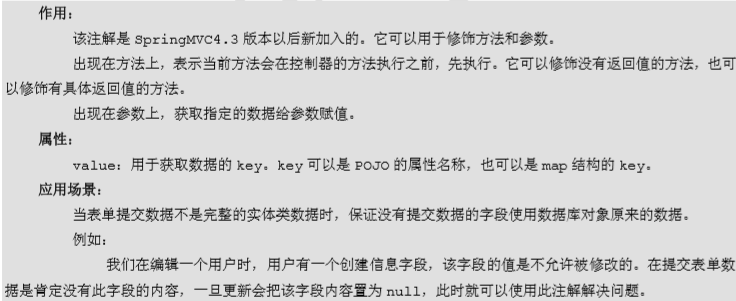
## CookieValue注解使用说明







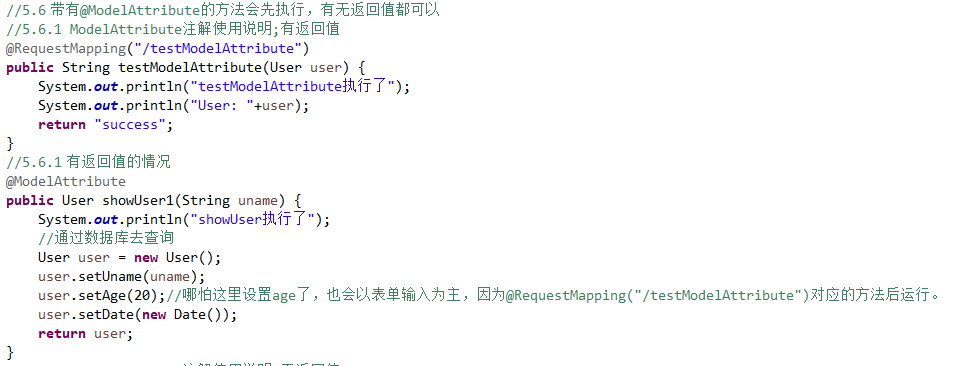
## ModelAttribute注解使用说明



假如一个实体类有3个属性，但form只有两个属性，封装成对象后，有一个属性会为空。可以使用ModelAttribute先执行其它方法，使实体类属性完整。



### ModelAttribute有返回值的情况。

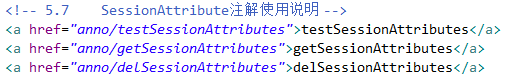


### ModelAttribute无返回值的情况。



## SessionAttribute注解使用说明

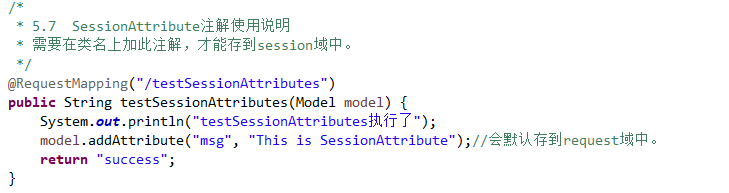




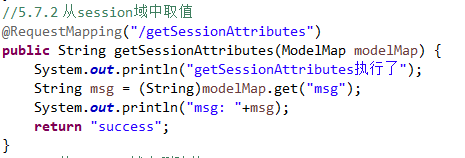


### 向session域中存值

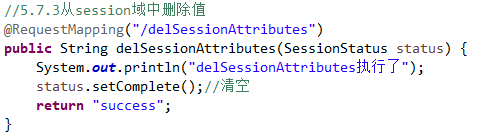




### 从session域中取值



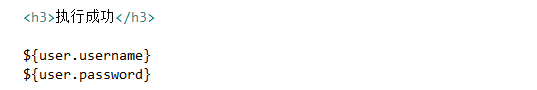
### 从session域中删除值



# 响应数据和结果视图

## 返回字符串



## 返回值是void



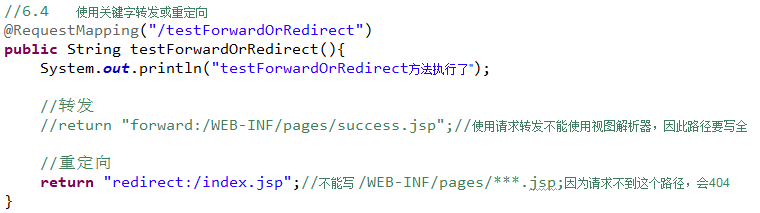
## 返回值是ModelAndView对象





## 使用关键字转发或重定向

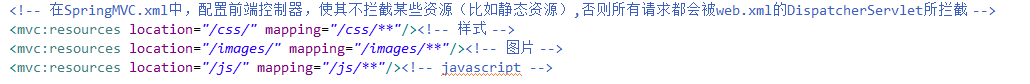


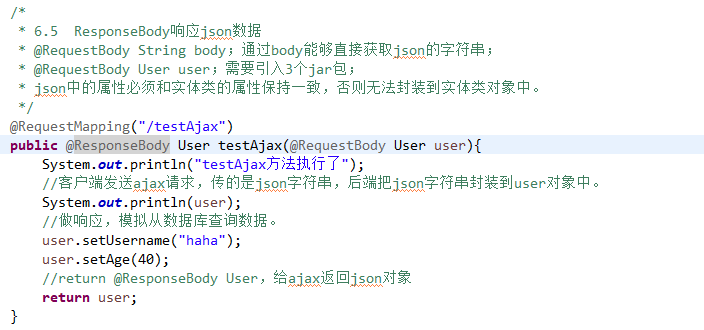


## ResponseBody响应json数据











# 文件上传

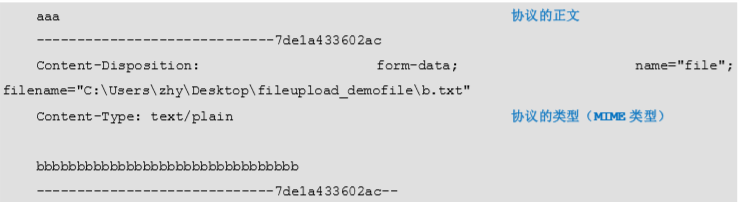


## 传统方法

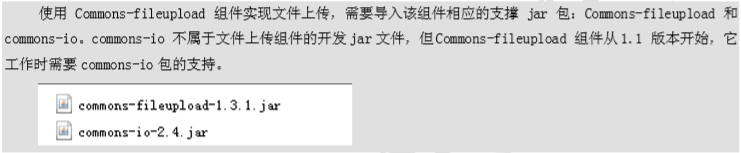
### 文件上传的必要前提



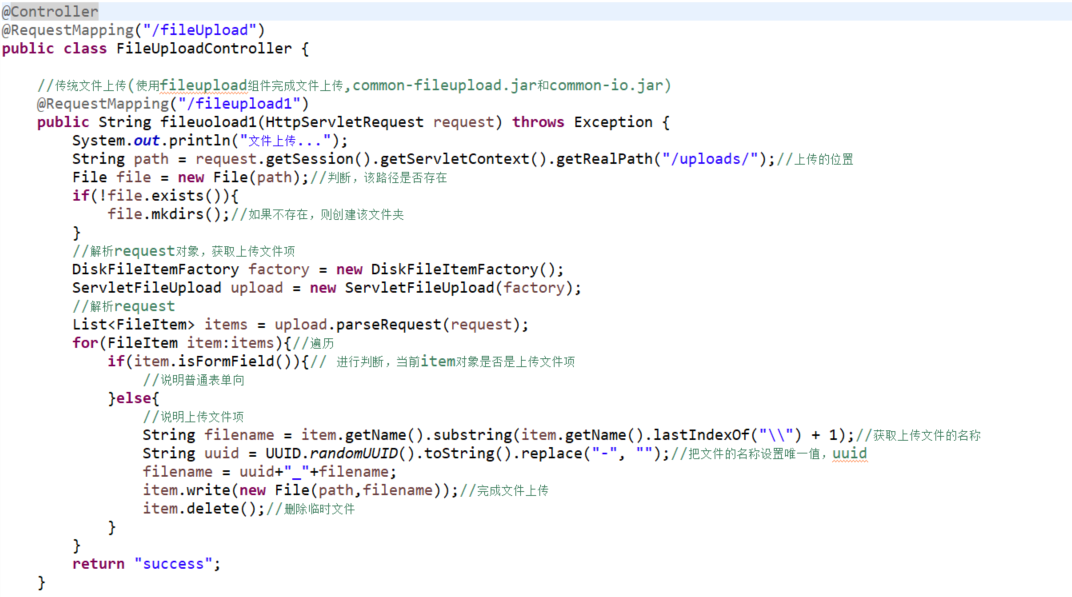
### 文件上传的原理分析



### 借助第三方组件实现文件上传

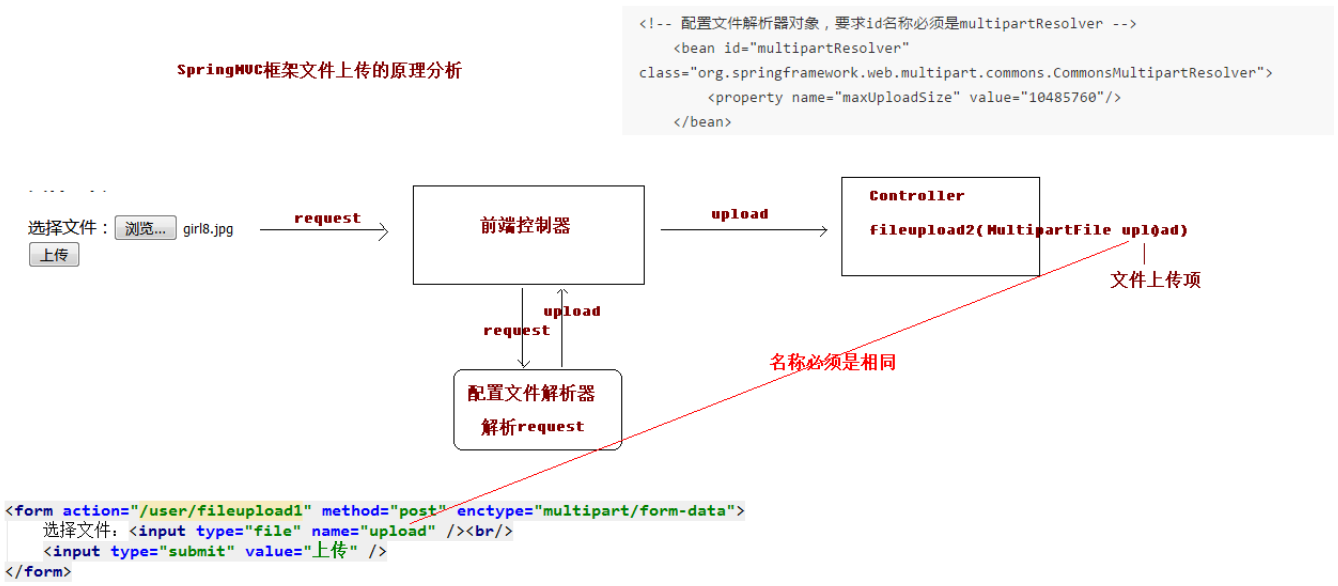


### 实现过程



## SpringMVC传统方式上传

### 原理



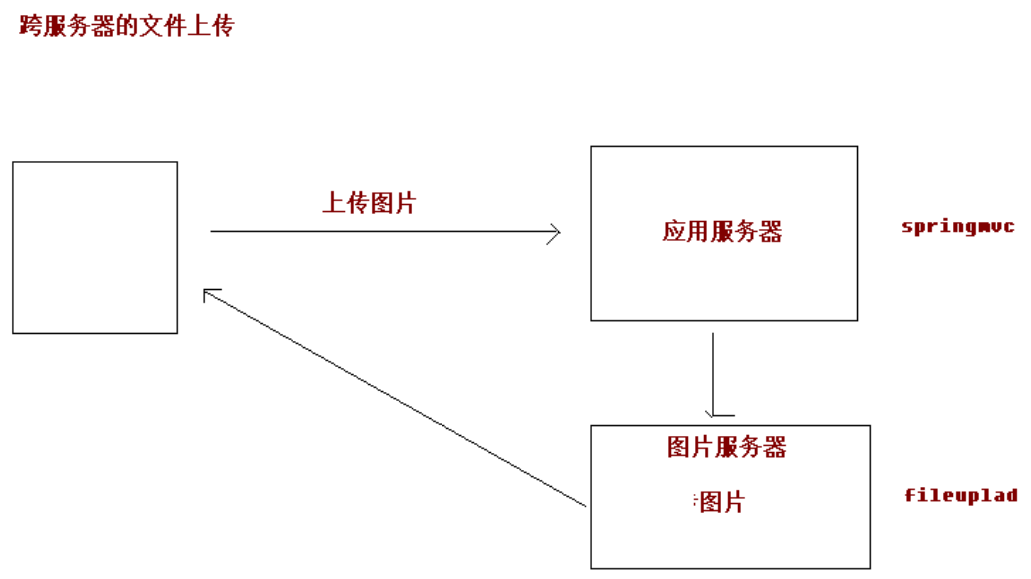
### 代码



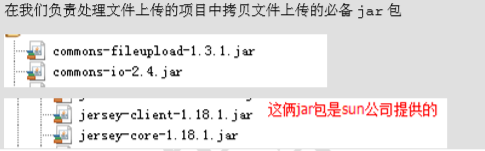


## SpringMVC跨服务器方式上传

### 原理



### 代码

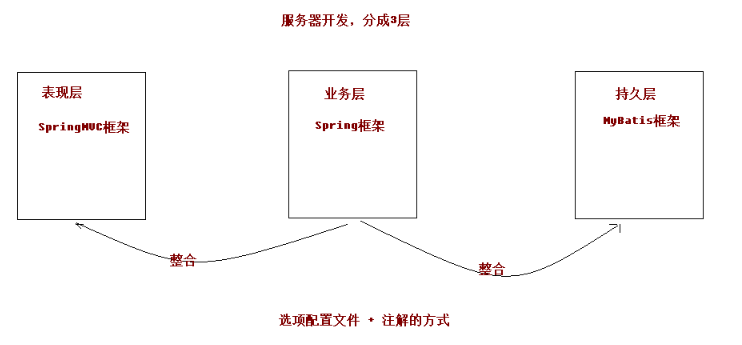




# SpringMVC的异常处理及拦截器

略，以后再看

# Spring整合SpringMVC



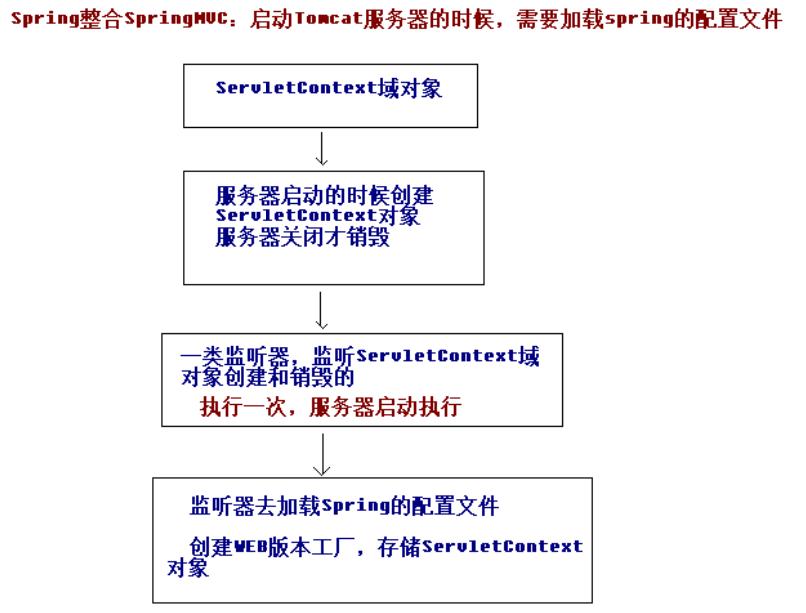
## 引入jar文件



## 编写web.xml

### 配置Spring监听器

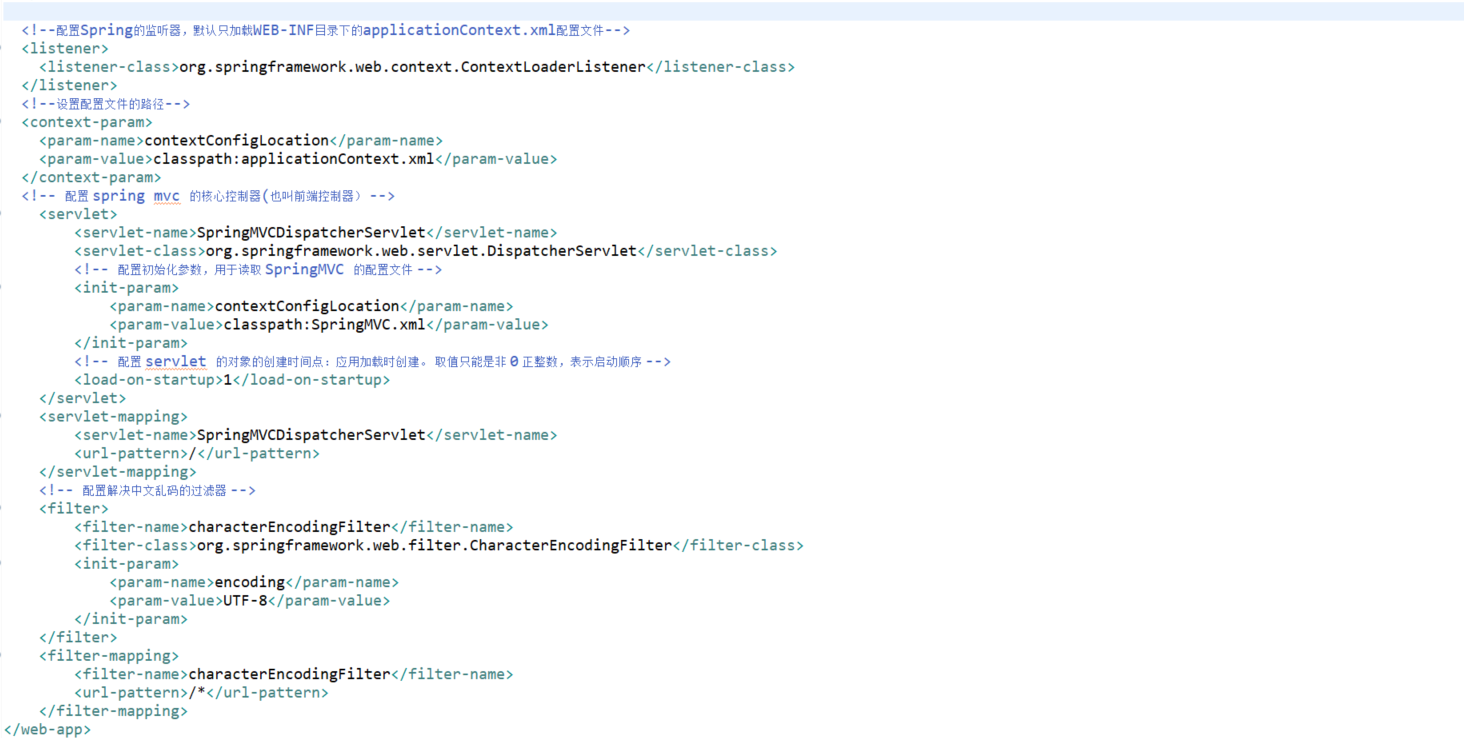




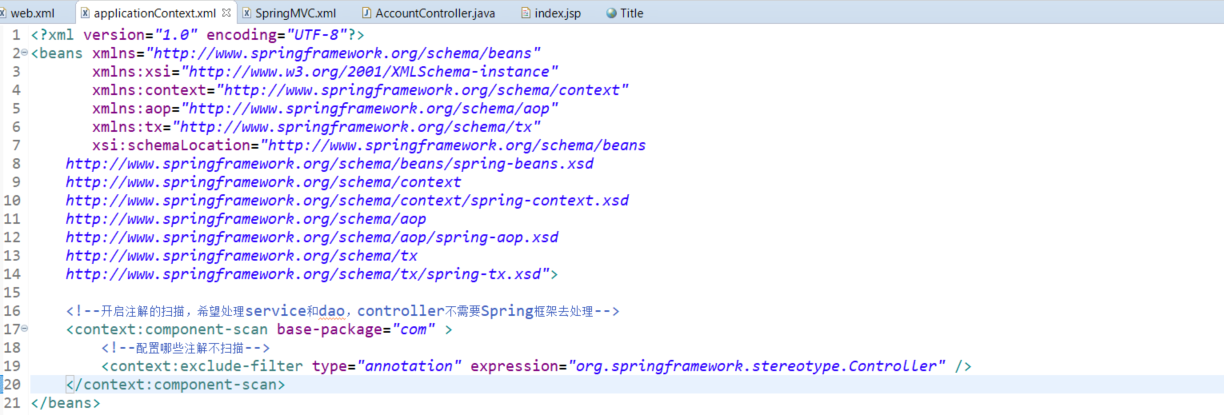
### 配置SpringMVC核心控制器



### web.xml总览



## 编写applicationContext.xml



## 编写SpringMVC.xml



## Spring整合SpringMVC