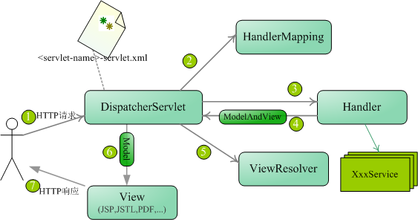
# 一、SpringMVC快速入门

## 1.配置web.xml文件

* SpringMVC的核心就是DispatcherServlet，DispatcherServlet实质也是一个HttpServlet。DispatcherSevlet负责将请求分发，所有的请求都有经过它来统一分发。
* 因此我们需要在web.xml中配置DispatcherServlet，并为contextConfigLocaltion 赋值，指定SpringMVC.xml配置文件的路径

eg:

## 2.配置springmvc.xml

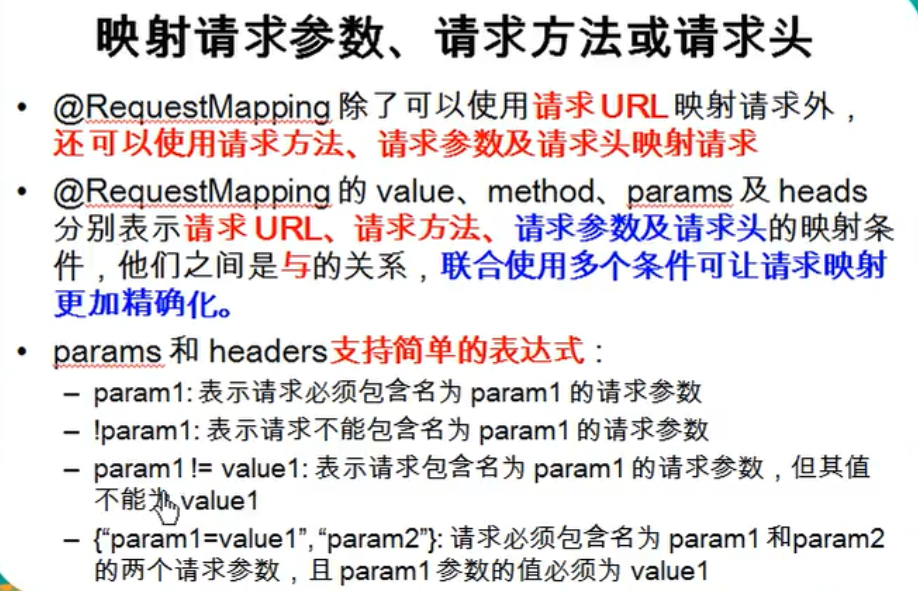
* 在springmvc.xml中我们需要配置自定义扫描包（context:component-scan）和视图解析器（InternalResourceViewResolver）
* context:component-scan:自定义扫描的包用来扫描base-package中的包，查看里面标注的注解 – 当我们使用注解进行开发时，必须配置自定义的扫描包，用来检查包下的注解
* InternalResourceViewResolver : 视图解析器使用来将handler返回的值解析为实际的物理视图;
* 使用 contenxt 必须添加xmlns:context="http://www.springframework.org/schema/context"

并且在schemaLocation中添加

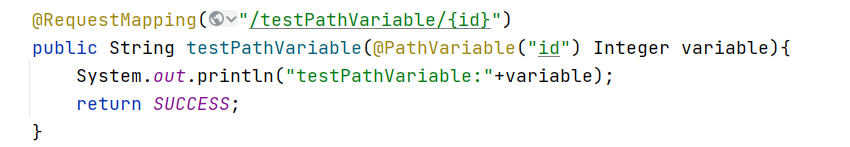
http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd "

# 二、SpringMVC的注解开发

## 1.RequestMapping

* RequestMapping (“/hello”)，此注解与视图解析器一起配合使用，将该路径转化为实际路径
* RequestMapping 可以用在方法与类上，当方法与类上都有时，其路径为：类路径/方法路径;方法上的路径相对于WEB-info的根目录下，而方法上的url相对于类路径下，如果类路径没有定义，则url相对于web-info下的根目录;
* RequestMapping 下的参数：
  + ****method : 用来指定请求方法 ->
  + ****params 和 headers

## 2.PathVariable

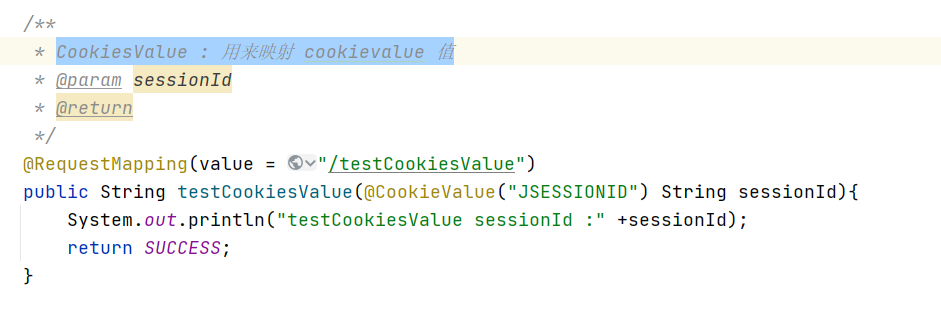
* PathVariable**:用来映射URL绑定的占位符**

## 3.RquestParam

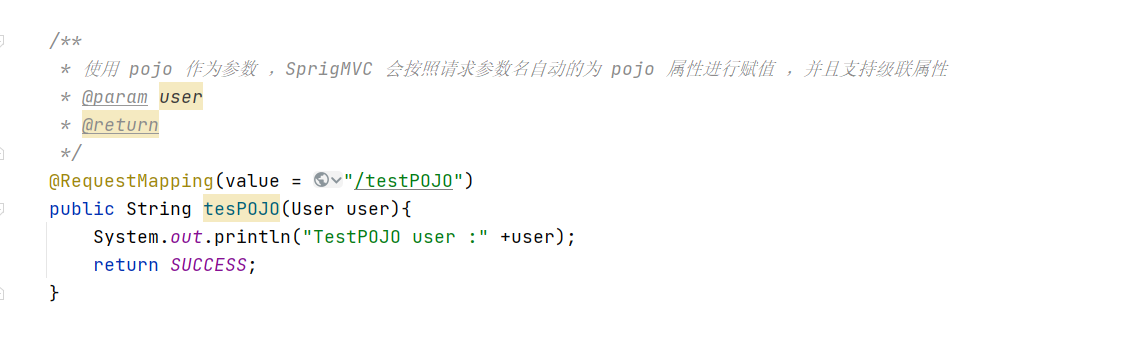
* **处理方法处使用此注解可以把请求参数传递给方法**
  + **-value : 参数名**
  + **-required : 是否必须。默认为true；**



## 4.CookieValue

* **用来映射cookie的值**

## 5.POJO作为参数

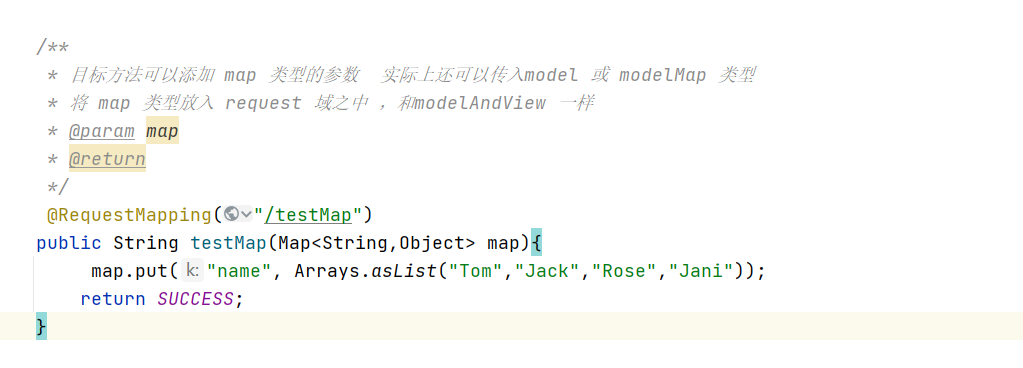
* **SpringMVC 会根据属性名自动为 pojo 进行赋值 ，并且支持级联操作**

## 6.使用原生Servlet API



## **7.ModelAndView**

* **同时我们也可以传入map类型的参数**



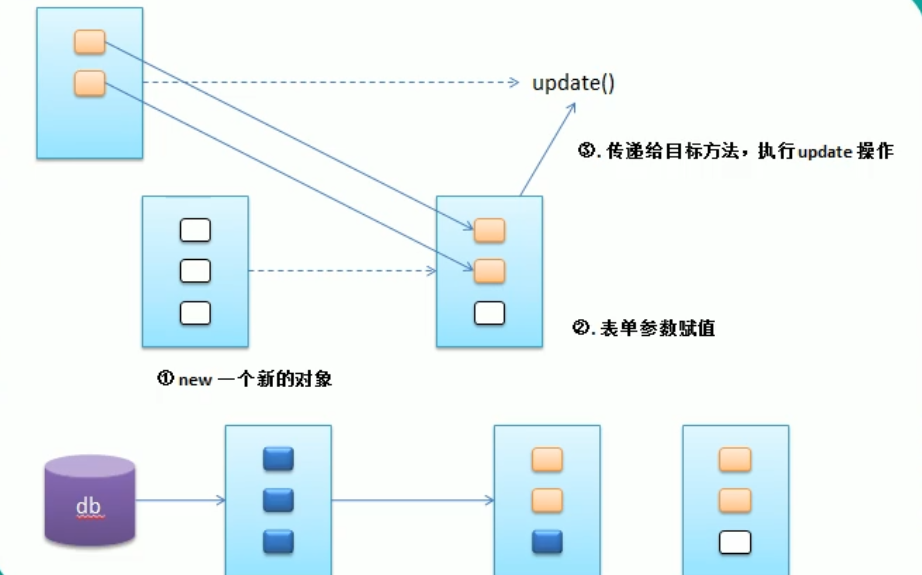
## **8.SessionAttribute**

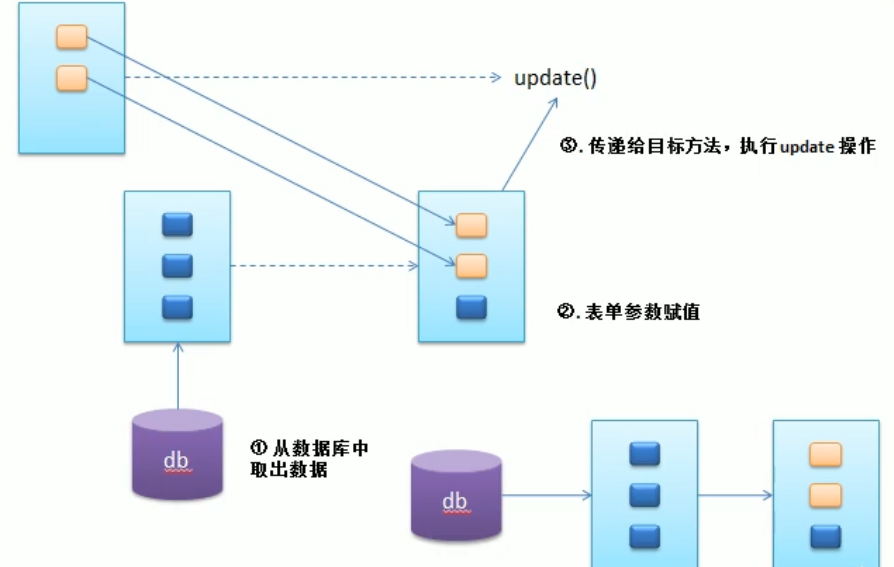
* @SessionAttribute : 作用在类上 有两个参数Value 和 type
  + Value map中的key
  + Type ： map中value的参数类型
* 范例:

# 三、ModelAttribute

## 1. ModelAttribute的使用场景

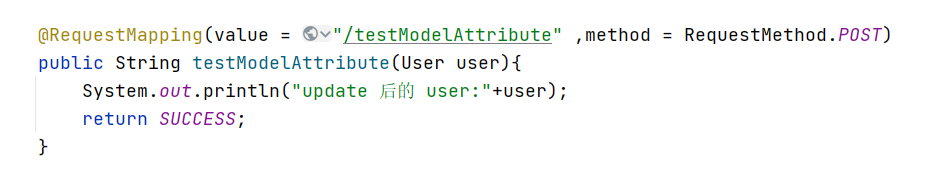
* 当我们想要更新数据时，我们只跟新了其中的一部分，而另一部分的值为空，此时我们如果我们直接将数据更新到数据库里面会导致，没有更新的数据缺失！！



* 此时如果我们不 new 一个新的对象，而从数据库之中拿数据，那么我们原始的对象就会有初始值，跟新时便不会发生数据丢失。
* 以上就是我们 ModelAttribute 的使用场景

## 2.ModelAttribute的使用方法

* 使用 ModelAttribute 标注的方法，会在每个目标方法执行之前被 SpringMVC 调用

我们为了解决之前的场景问题，可以在执行目标方法之前将 User 数据先从数据库之中取出放入 request域之中，这样在执行以下目标方法的时候，SpringMVC 根据名字自动赋值的对象已经拥有了数据！这样更新操作就不会丢失数据！

* 注意我们在将user放入 map.put() 时的key 必须和入参类型的第一个字母小写一致：也就是说当我的 入参类型为 Address 时，我们的map.put(“address”,address);其中key 必须为 address才能使得我们SpringMVC 在map中查找时将数据库中查找到的值与入参对应！