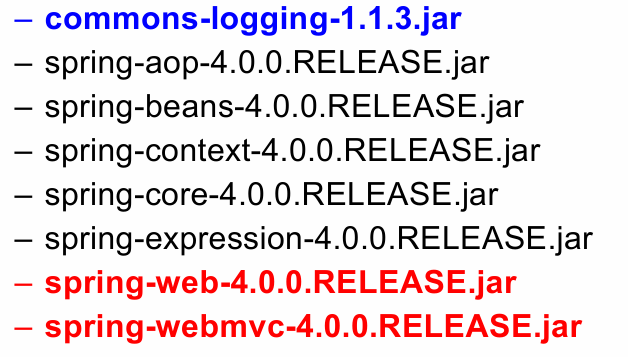
# Spring MVC（基于spring4.x）

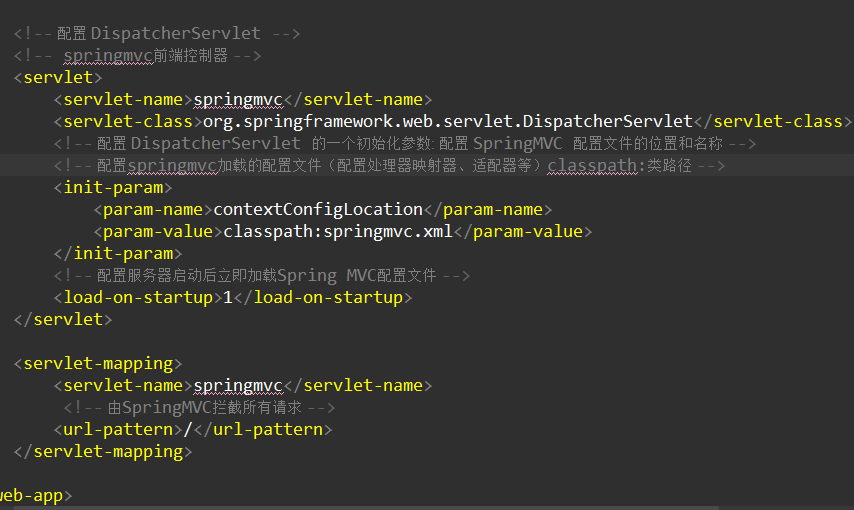
## HelloWorld

步骤： •

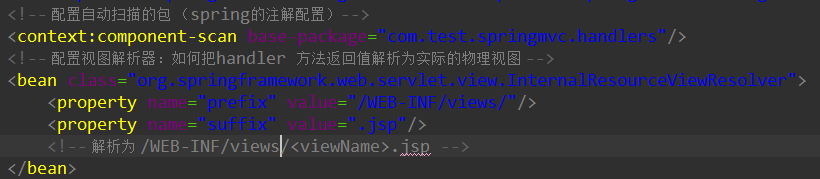
1. 加入 jar 包 –



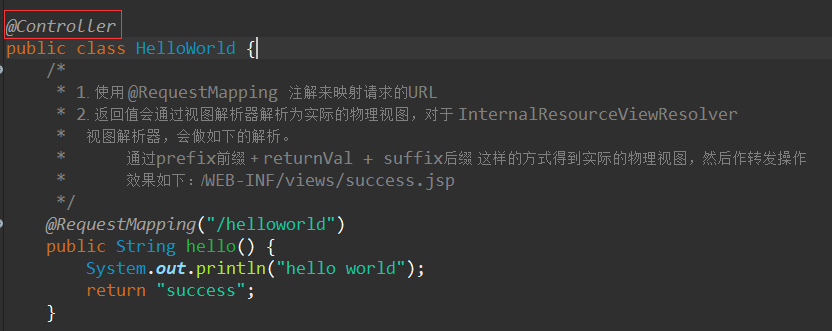
1. 在 web.xml 中配置 DispatcherServlet –



1. 加入 Spring MVC 的配置文件 –



1. 编写处理请求的处理器，并标识为处理器 用Controller注解标识

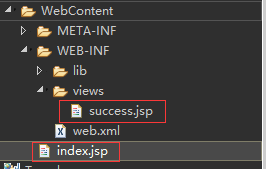


1. 编写视图

请求连接



结构



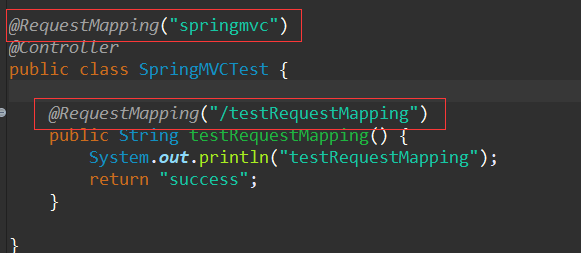
## @RequestMapping

@RequestMapping 不仅可以修饰方法，还可以修饰类

类定义处：提供初步的请求映射信息。相对于 WEB 应用的根目录。

方法定义处：提供进一步的细分映射信息。相对于类定义处的 URL。若类定义处未标注 @RequestMapping，则方法处标记的 URL 相对于WEB 应用的根目录。

处理器



处理器类可以定义多个处理方法，处理来自springmvc下的请求

视图



@RequestMapping 除了可以使用请 请求 求 URL 映射请求外，还 可以使用 请 求方法、 请 求参数及 请 求 头 映射 请 求

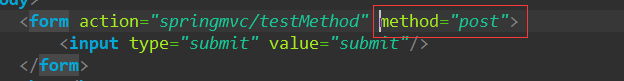
@RequestMapping 的 value、method、params 及 heads 分别表示请URL、请求方法、 请求参数及 请求头的映射条件，他们之间是与的关系，联合使用多个条件可 让请 求映射更加精确化。

1. 使用 method 属性来指定请求的方式处理器

处理器中内容

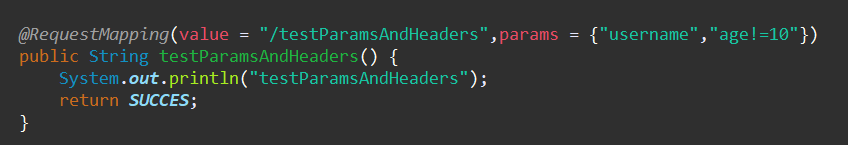


Post请求



2. 了解：可以使用params和headers来更加精确的映射请求。params和headers 支持简单的表达式

例如 连接中必须包含username 和age属性并且age的值不为10.才能正常访问。



正常可以访问到的



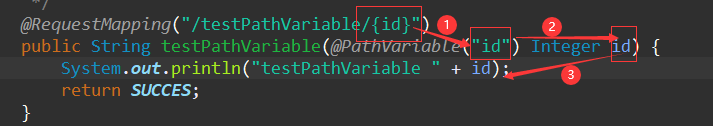
访问不到的



## 映射 请求参数 求参数 & 请 请 求参数

### 3.1 @PathVariable

@PathVariable 可以来映射URL中的占位符到目标方法的参数中



前端



### 3.2 Rest 风格的Url

以CRUD为例：

/order/1 HTTP GET ：得到 id = 1 的 order get?id=1

/order/1 HTTP DELETE ：删除 id = 1的 order delete?id=1

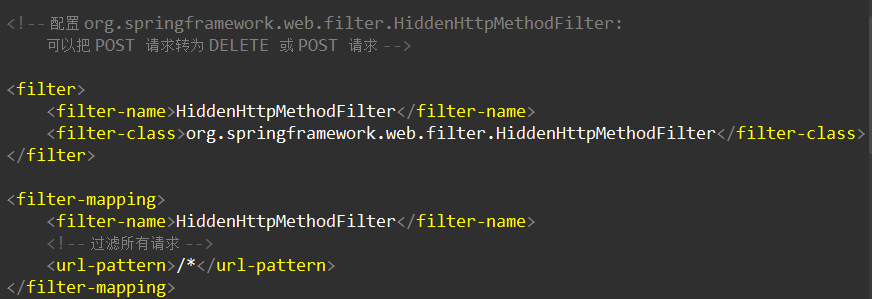
/order/1 HTTP PUT：更新id = 1的 order pudate?id=1s

/order HTTP POST：新增 order

如何发送PUT和DELETE请求？

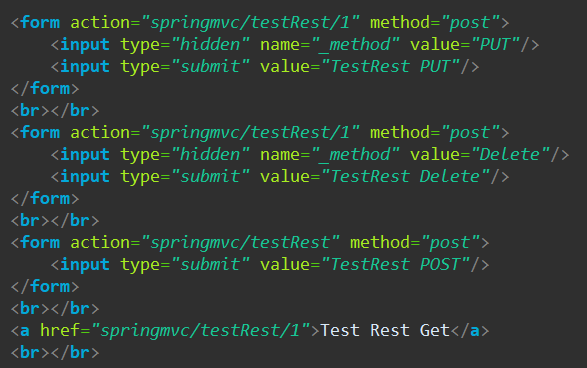
1． 需要配置一个 HiddenHttpMethodFilter

浏览器 form 表单只支持 GET 与 POST 请求，而DELETE、PUT 等 method 并不支持，Spring3.0 添加了一个过滤器，可以将这些请求转换为标准的 http 方法，使得支持 GET、POST、PUT 与DELETE 请求。



2. 需要发送POST请求

1. 需要在发送POST请求时携带一个隐藏域name=”\_method”的隐藏域值为DELETE或者PUT。



4. 在Spring MVC 的目标方法中如何得到id？

使用method声明出来的请求

使用@PathVariable注解



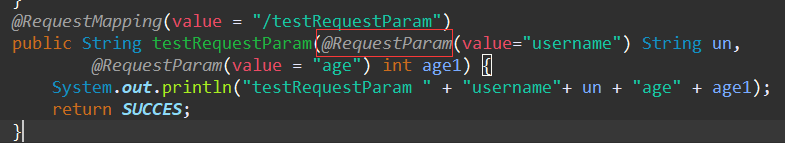
### 3.3 @RequestParam

用使用@RequestParam 可以把 可以把请求参数传递给请求方法。

1. 前端页面传入一个username和age两个属性值



1. 在处理器中使用RequestParam获取前台传入的username和age的值分别给un和age1。在处理器中可以直接输出un各age1的值。

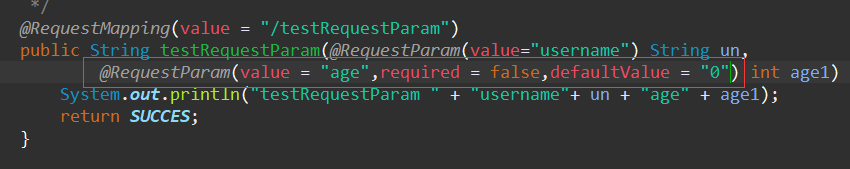


使用@RequestParam来映射请求参数，其中

value 值即为请求参数的参数名

required 该参数是否必须，默认为true使必须的。表示请求参数中必须包含对应 的参数，若不存在，将抛出异常。

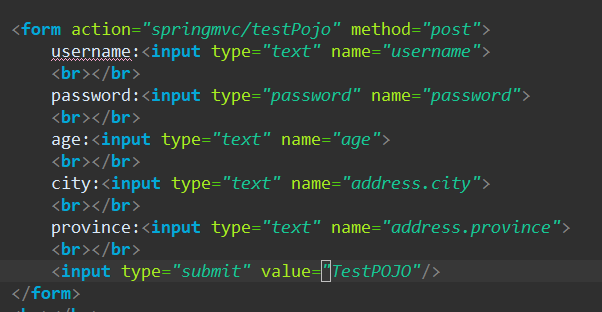
DeafltValue 请求参数的默认值



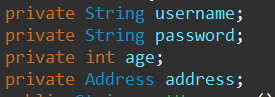
3.4 使用POJO对象绑定请求参数值

Spring MVC 会按请求参数名和 求参数名和 POJO 属性名属性名进行自动匹配，自动为该对象填充属性值。支持级联属性，address.city,address.province。

前端页面

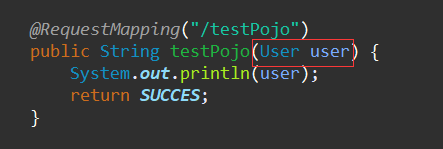


User和Address类





处理器



3.4 使用

## 4. 处理模型数据

Spring MVC 提供了以下几种途径输出模型数据：

ModelAndView: 处理方法返回值类型为 ModelAndView 时, 方法体即可通过该对象添加模型数据

* Map 及 Model: 入参为org.springframework.ui.Model 、org.springframework.ui.
* ModelMap 或 java.uti.Map 时，处理方法返回时，Map中的数据会自动添加到模型中。
* @SessionAttributes: 将模型中的某个属性暂存到HttpSession 中，以便多个请求之间可以共享这个属性。
* @ModelAttribute: 方法入参标注该注解后, 入参的对象就会放到数据模型中。

### 4.1 ModelAndView

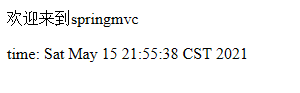
处理器



返回jsp页面

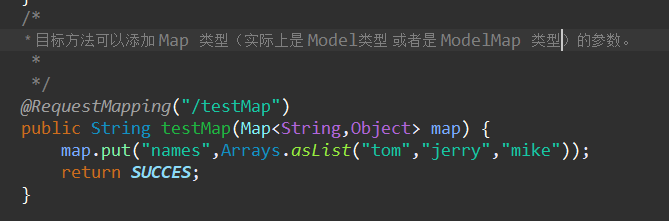


效果

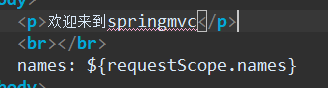


### Map 及 Model

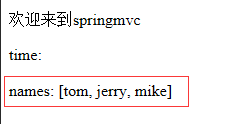
控制器



返回jsp页面

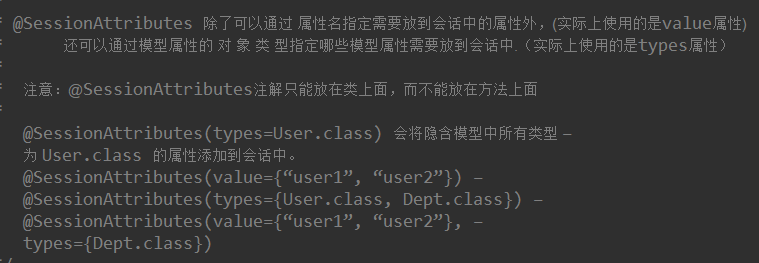


效果

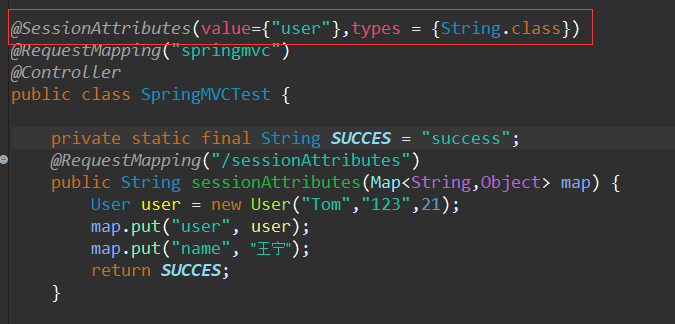


### @SessionAttributes

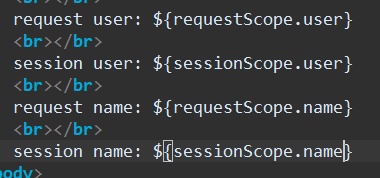
**把ModelMap里面的数据放入Session里面**



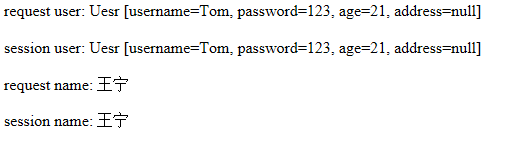
示例



返回jsp页面



效果

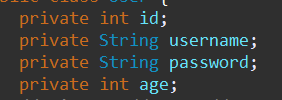


### @ModelAttribute、

在方法定义上使用@ModelAttribute注解：Spring MVC 在调用目标处理方法前，会先逐个调用在方法级上标注了@ModelAttribute 的方法。

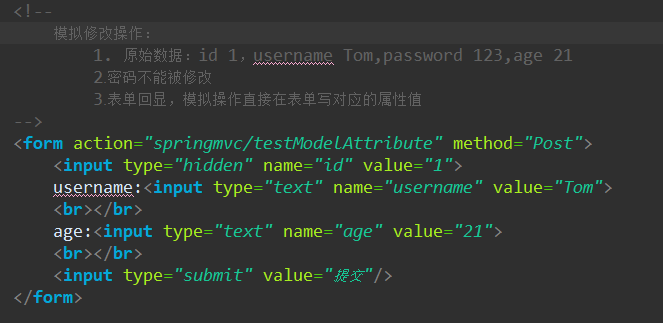
（有 @ModelAttribute 标记的方法 ，会在每个目标方法执行之前被 SpringMVC 调用!）

User类

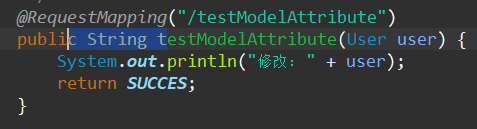


Jsp页面

在修改数据时候，密码不修改。Id属性隐藏



不使用 @ModelAttribute 注解



结果



问题 ：password 属性为空，在修改时会把null传入数据库是password 为空

使用 @ModelAttribute 注解



运行流程：

1. 首先会执行 @ModelAttribute 注解修饰的方法。从数据库中取出id为1的对象，将查询出来的对象封装在Map集合中 键为user。
2. SpringMVC 从Map中取出User 对象,并把表单的请求参数赋给该 User 对象的对应属性。（表单中的属性会覆盖map中Map 中user对象的属性）
3. SpringMVC 把上述对象传入目标方法的参数。

注意：在 @ModelAttribute 修饰的方法中 放入到 Map 的时候键，需要和目标方法入参类型的第一个字母小写的字符串中。

运行结果

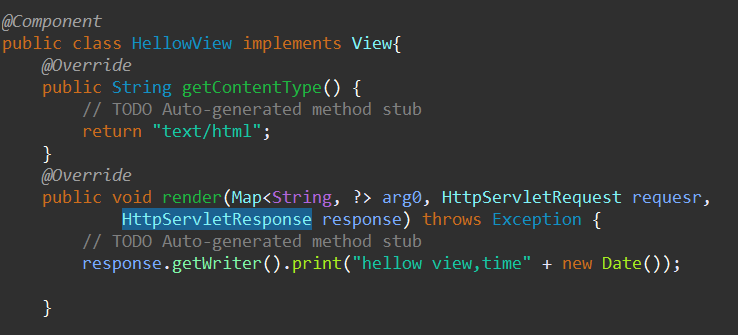


了解：前面所说在 @ModelAttribute 修饰的方法中 放入到 Map 的时候，键需要和目标方法入参类型的第一个字母小写的字符串中。这是在默认情况下。如果Map 中的键不是和目标方法入参类型的第一个字母小写的字符串。可以在目标方法中使用@ModelAttribute 注解。

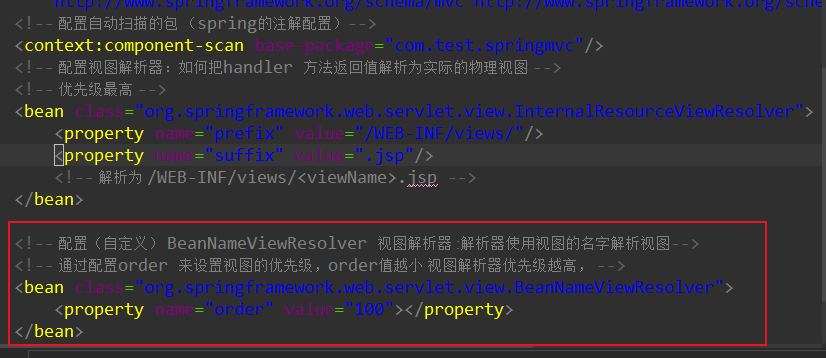


## 视图和视图解析器

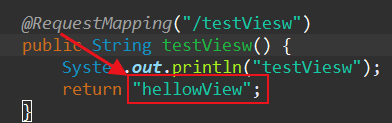
1. 配置一个自定义类实现View接口



2． 编写全局配置文件 BeanNameViewResolver



3. 测试类（返回值类名小写）



4. 测试结果



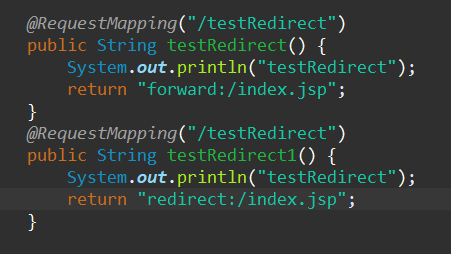
## 6. 重定向

一般情况下，控制器方法返回字符串类型的值会被当成逻辑视图名处理如果返回的字符串中带 forward: 或 redirect: 前缀 时，SpringMVC 会对他们进行特殊处理：将 forward: 和redirect: 当成指示符，其后的字符串作为 URL 来处理

redirect:success.jsp：会完成一个到 success.jsp 的重定向的操作

forward:success.jsp：会完成一个到 success.jsp 的转发操作

如图

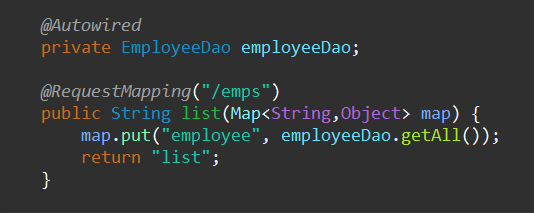


该方法被调用后会直接跳转到index.jsp页面。

## 7. RESTful CRUD

### 1. 模拟查询员工信息（查询所有）

将数据库中所有的员工信息放入map集合中



list.jsp将查询到的信息展示出来页

当map中的key为空，则数据库中没有对应信息

### 2. 模拟添加员工信息以及（员工信息中包含部门信息）



**控制器**



**前台使用Spring中的表单标签（from标签）**

思考为什么要使用form标签？

可以更加快速的开发表单页面，而且可以更方便的进行表单值的回显。

1. 声明标签



* 一般情况下， 通过GET 请求获取表单页面，而通过POST 请求提交表单页 面， 因此获取表单页面和提交表单页面的URL是相同的。 只要满足该最佳条件的契约，<form:form> 标签就无需通过 action 属性指定表单提交的 URL.
* 可以通过 modelAttribute 属性指定绑定的模型属性，若没有指定该属性，则默认从 request 域对象中读取command 的表单 bean，如果该属性值也不存在，则会发生错误。（所以这里在控制器中为啥加，下图中红框中的数据）



modelAttribute：是设置该表单，对那个bean进行操作。表单中的属性必须给该bean中的属性。（我们是对员工信息进行添加，所以是对EmployeeDao这个bean操作）



* 表单标签

SpringMVC提供了多个表单标签如：<form:input>,<form:select>等，可以绑定单表字段的属性值，它们有共同的属性如下：

--path：表单字段，对应html元素的name，支持级联属性。

-- htmlEscape：是否对表单值的 HTML 特殊字符进行转换，默认值为 true

--cssClass：表单组件对应的 CSS 样式类名

--cssErrorClass：表单组件的数据存在错误时，采取的 CSS 样式

form:input、form:password、form:hidden、form:textarea 对应 HTML 表单的 text、password、hidden、textarea标签。

--form:radiobutton：单选框组件标签，当表单 bean 对应的属性值和 value 值相等时，单选框被选中。

--form:radiobuttons：单选框组标签，用于构造多个单选框。

--items：可以是一个 List、String[] 或 Map

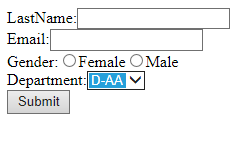
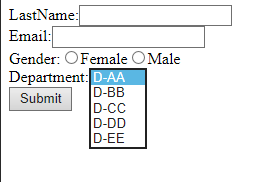
--itemValue：指定radio的value值。可以是集合中bean的一个属性值

--itemLabel：指定 radio 的 label 值。

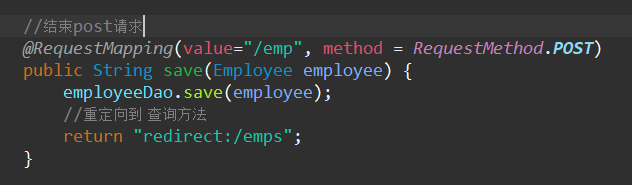
--delimiter：多个单选框可以通过 delimiter 指定分隔符

前端效果页面

（这里的部门信息是从数据库中查询出来的数据，而不是自己写的数据）

点击Submit后调用控制器中的方法



注意再次，在点击添加员工信息连接时，和输入数据点击提交按钮时传入的连接都是emp。（但是请求方式不同，所以不会影响）。

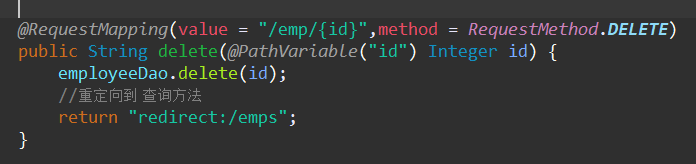


### 3. 删除操作

jsp页面删除标签（该操作使用的是一个get请求链接）



控制器（控制器接受的是一个DELETE请求）



注：DELETE是在form标签中的。form默认使用POST请 求（这里通过配置web.xml 将POST请求转为 DELETE请求）

这里需要用到Spring MVC 处理静态资源

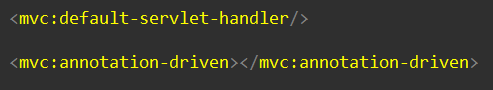
SpringMVC 处理静态资源：

1. 为什么会有这样的问题

优雅的 REST 风格的资源URL 不希望带 .html 或 .do 等后缀若将 DispatcherServlet 请求映射配置为 /，则 Spring MVC 将捕获 WEB 容器的所有请求，包括静态资源的请求，SpringMVC 会将他会将他们当成一个普通请求处理，因找不到对应处理器将导致错误。

2.解决：在 SpringMVC 的配置文件中配置 <mvc:default-servlet-handler/>

springmvc.xml 中的配置



* mvc:annotation-driven 将在 SpringMVC 上下文中定义一个DefaultServletHttpRequestHandler，它会对进入 DispatcherServlet 的请求进行筛查，如果发现是没有经过映射的请求，就将该请求交由 WEB应用服务器默认的 Servlet 处理，如果不是静态资源的请求，才由DispatcherServlet 继续处理.
* 一般 WEB 应用服务器默认的 Servlet 的名称都是 default。若所使用的 WEB 服务器的默认 Servlet 名称不是 default，则需要通过 default-servlet-name 属性显式指定.

jsp页面使用js进行转换.(这里和删除链接同页面)

