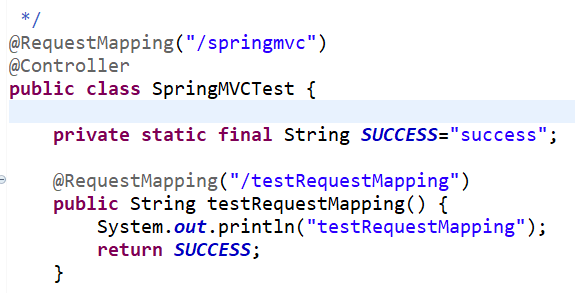
@RequestMaping注解

1.@RequestMapping：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：提供一个细分信息，相对于类定义处的url，如果类定义处没有定义，则相对于WEB应用的根目录

所以这样访问的路径就是/springmvc/testRequestMapping

[2.@RequestMapping](mailto:2.@RequestMapping)的一些属性

1. 默认的一个属性，value属性，用来指定所匹配的url

method属性，可以指定访问的方式（get和post）



这样指定了就只接受post请求，而发送一个get请求到这个url的话会报错

③ \* 可以用params属性来指定请求参数里面所包含的信息，而且还可以多个，headers属性来指定请求头里面所包含的信息

\* 但是这两个属性不常被使用





④RequestMapping里面写所匹配的路径可以使用ant形式的通配符来表示

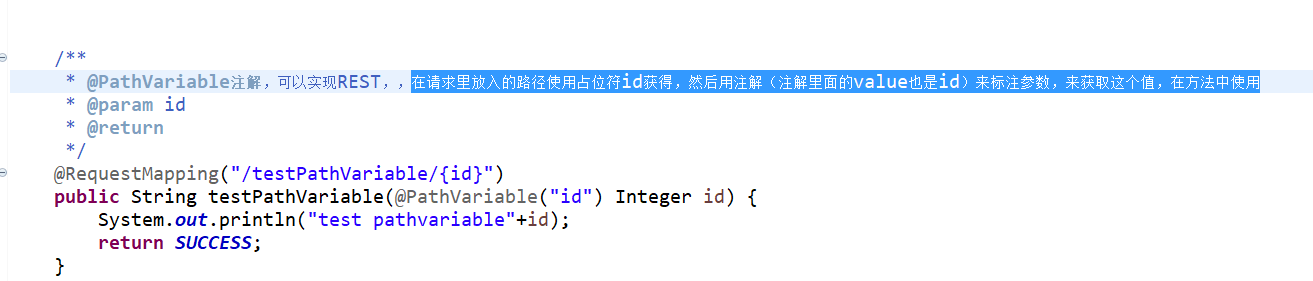






⑤还可以使用@PathVariable注解来实现REST在请求里放入的路径使用占位符id获得，然后用注解（注解里面的value也是id）来标注参数，来获取这个值，在方法中使用

通过 @PathVariable 可以将 URL 中占位符参数绑定到控•制器处理方法的入参



补充：REST：多的也不会，，先了解一点

，就是 HTTP 协议里面，四个表示操作方式的动 词：GET、POST、PUT、DELETE。它们分别对应四种基本操作：GET 用来获 取资源，POST 用来新建资源，PUT 用来更新资源，DELETE 用来删除资源。

REST风格的URL

\* 以CRUD为例

\* 新增：POST

\* 修改：PUT

\* 获取：GET

\* 删除：DELETE

\*

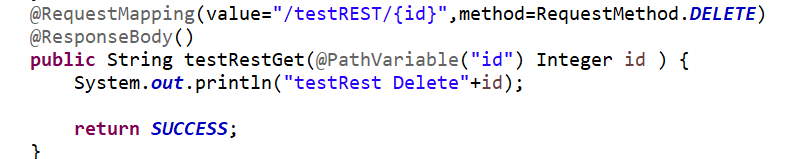
\* 如何发送PUT请求和DELETE请求呢？

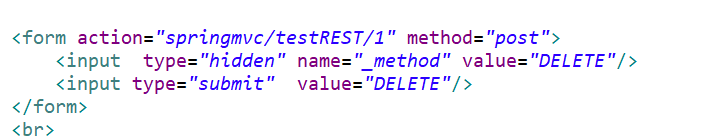
\* 1.需要配置HiddenHttpMethodFilter

\* 2.需要发送Post请求

\* 3.需要在发送post请求时携带一个隐藏域，name=\_method,value=PUT或DELETE

\* 4.注意tomcat如果在7.0版本以上，需要在方法前面加上@ResponseBody注解（原因请去CSDN搜索）





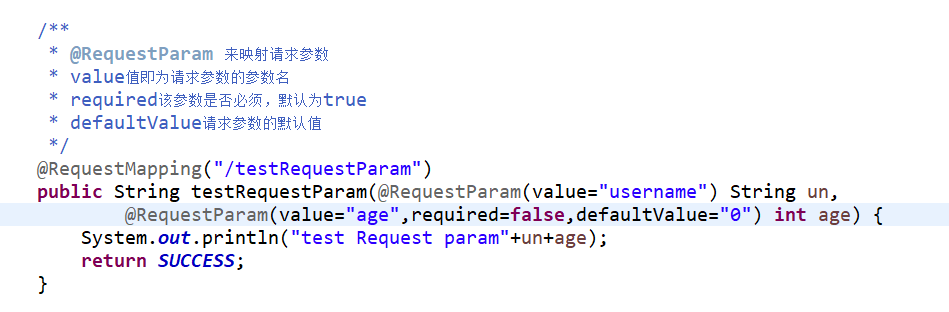
3.使用@RequestParam注解标注入参，可以在方法中搞到请求参数，该注解重要，在开发中使用的机会较多

**@RequestParam** 来映射请求参数

\* value值即为请求参数的参数名

\* required该参数是否必须，默认为true

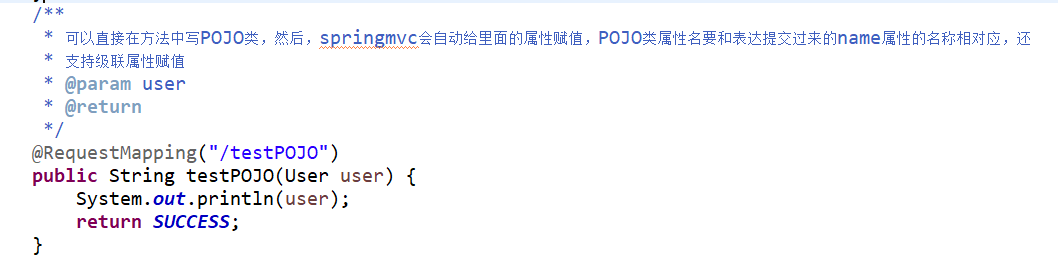
\* defaultValue请求参数的默认值

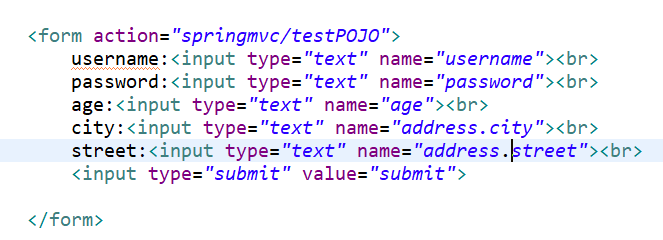


[4.使用@RequestHeader](mailto:4.使用@RequestHeader)注解可以在方法中搞到请求头信息，使用机会较少，了解即可，使用方法和上面的@RequestParam注解相同

[5.@CookievValue](mailto:5.@CookievValue):映射一个Cookie值，属性同@RequestParam

6．\* 可以直接在方法中写POJO类，然后，springmvc会自动给里面的属性赋值，POJO类属性名要和表达提交过来的name属性的名称相对应，还支持级联属性赋值





还可以接受某些servlet的api作为参数，这个挺重要的，因为在开发过程中难免要用到Servlet原生api

HttpServletRequest：即原生的。。。

• HttpServletResponse：即原生的。。

• HttpSession：即session

• java.security.Principal：

• Locale：

• InputStream：

• OutputStream：

• Reader：request的getreader方法返回的

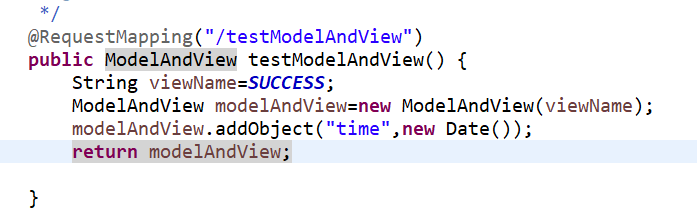
• Writer：response的getwriter方法返回的

7. 处理模型数据

Spring MVC 提供了以下几种途径输出模型数据：

把model数据放到请求域当中

a)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，然后这些数据就可以用EL在jsp页面上直接显示



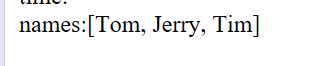




b)Map 及 Model: 入参为 org.springframework.ui.Model、org.springframework.ui. ModelMap 或 java.uti.Map 时，处理方法返回时，Map 中的数据会自动添加到模型中。

• Spring MVC 在调用方法前会创建一个隐– 含的模型对象作为模型数据的存储容器。 如果方法的入参为 Map 或 Model 类– 型，Spring MVC 会将隐含模型的引用传 递给这些入参。在方法体内，开发者可以 通过这个入参对象访问到模型中的所有数 据，也可以向模型中添加新的属性数据





将model数据放入到session域中

a）@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

@SessionAttributes 除了可以通过属性名（通过value属性）指定需要放到会话中的属性外，还可以通过模型属性的对象类型（通过types属性）指定哪些 模型属性需要放到会话中

注意：这个注解是放在Controller类上，而不能放在方法上，而且可以指定value和types属性，value属性是一个字符串是数组，只要下面方法的模型对象的键与这个属性相同就把这个属性放入到request和session域

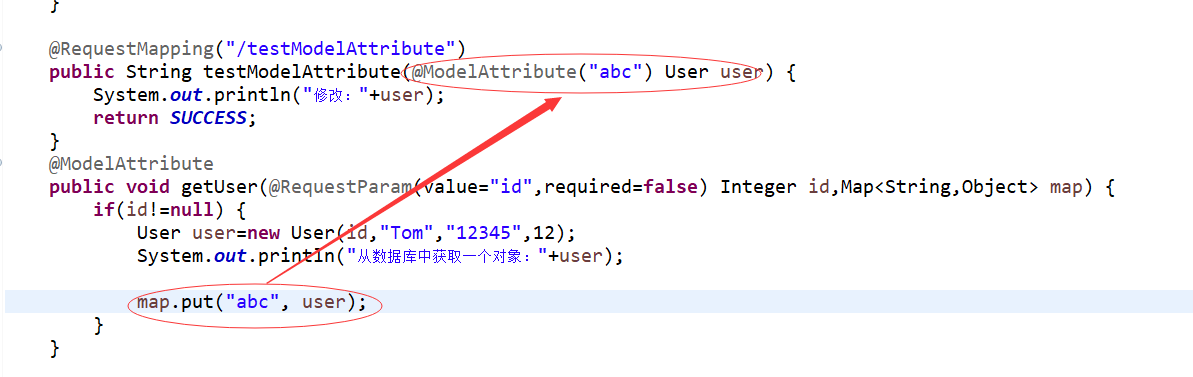
types属性时一个class类型的数组，只要下面的方法的模型对象的class类型和这个属性所指定的一致，就把它放入到session和request域中

b）@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

好吧，，下面两句1.2.你先看一遍然后往下看看完代码再看一遍就差不多能懂表示的是什么意思

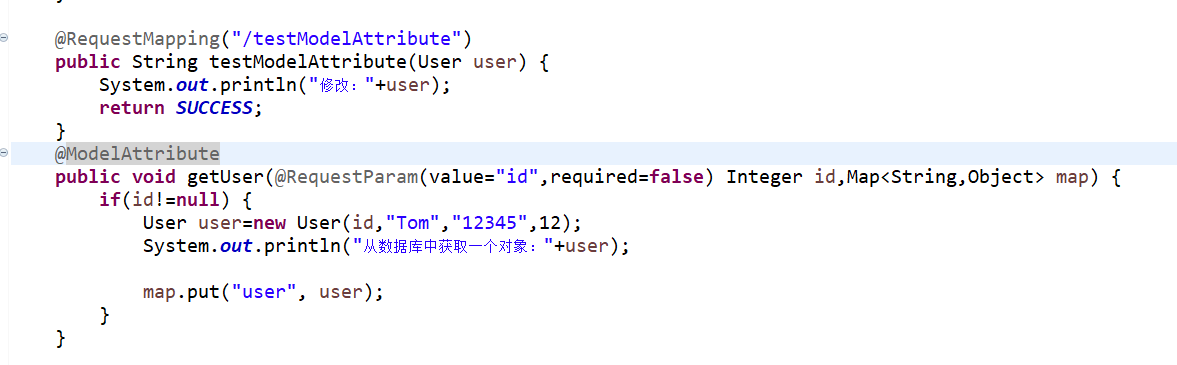
1.在方法定义上使用 @ModelAttribute 注解：Spring MVC 在调用目标处理方法前，会先逐个调用在方法级上标注了 @ModelAttribute 的方法。

2.在方法的入参前使用 @ModelAttribute 注解：可以从隐含对象中获取隐含的模型数据中获取对象，再将请求参数绑定到对象中，再传入入参 将方法入参对象添加到模型

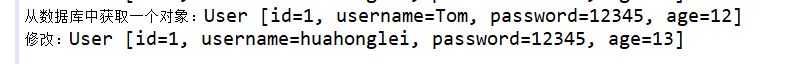


（这个使用场景如果一时看不懂先别看，毕竟都是模拟，而且这个写的也只是一个示例，先写个简单示例明白这个注解是干什么的更为重要）

经典使用场景：从数据库获取一个对象，然后利用传入的一部分值修改这个对象。其中在修改这个对象之前需要从数据中获取一个对象，，这个时候需要用此注解







可以看出在jsp页面里并没有为password赋值，而修改的password里面有值，这就说明当前修改的这个user是从数据库中取出来的那个user

运行流程

1.执行@ModelAttribute注解注释的方法：从数据库中取出对象，把对象放入到Map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

3.springmvc把上述对象传入目标方法的参数

注意：在将@ModelAttribute修饰的方法中，放入到Map时的键需要和目标方法入参类型的第一个字母小写的字符串一致

此处略去一大段源码分析过程，，，是真滴看不懂啊，，视频中有，自己看吧，，，真的是受不了，太长了，以后学到一定程度后，想起来这个一定要看看这个源码

源码分析之后的两个结论

1.这个注解也可以修饰目标方法POJO类型的入参，其value属性值有如下作用，springmvc会使用value属性值在implicitModel中查找对应的对象，若存在则会直接传入到目标方法的入参中

2.springmvc会以value为键，POJO类型的对象为值，存入到request中

（这一段话纯属自己理解，理解的简单点好像也就这回事）默认的话是以类名小写第一个字母，当然也可以使用注解的value属性来指定，如果都没有则springmvc会为你新建一个user对象，往里面赋值，所以这样修改的就不是刚刚从数据中获取的那个user，而是往新创建的user对象里面赋值。而如果当前的模型对象里面没有对应的user，则程序会去找@SessionAttribute