方法参数

1. 使用POJO作为参数

SpringMVC会按照请求参数名和POJO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。如：dept.deptId、dept.address.tel等

1. **写一个表单**

<form action=*"hello/testPOJO"* method=*"post"*>

username：<input type=*"text"* name=*"username"*/><br>

password：<input type=*"password"* name=*"password"*/><br>

age：<input type=*"text"* name=*"age"*/><br>

city：<input type=*"text"* name=*"address.city"* /><br>

street：<input type=*"text"* name=*"address.street"* />

<input type=*"submit"* value=*"submit"*/>

</form>

1. **根据表单的name名写两个对应的Java类**

一个user类，对应属性username，password，age，address

一个address类，对应属性city，street

1. **使用**

把对象当参数传入接受请求的方法，就会自动根据name名和属性名的对应，填充该对象的属性值

@RequestMapping("/testPOJO")

**public** String testPOJO(User user) {

System.***out***.println(user);

**return** ***SUCCESS***;

}

POJO入参步骤

1. 先确定key
   1. 若目标参数没有被@ModelAttribute注解修饰，或该注解中没有一个vaule属性，则key值取类名首字母小写，
   2. 如果有则取注解的value值
2. 确定key对应的对象
   1. 先看处理器类中是否有@ModelAttribute注解标识的方法

如果有，则看看@ModelAttribute注解标识的方法里是否有map参数，如果有，看看map方法里是否有确定的key，如果有则使用这个key对应的对象做参数，如果没有，往下走

* 1. 看看这个类是否被@SessionAttributes注解标识，如果被标识，则看看注解value值中是否有确定的key，如果有，使用key对应的对象做参数，如果属性值为空，则抛出异常
  2. 如果没有被@SessionAttributes注解标识或注解的vlaue属性中不包含这个key，则利用反射创建一个对象，传入到目标方法做参数

1. 在此方法结束后，SpringMVC会把该对象自动传入到request域对象中
2. 使用serlvet原生API做参数

支持的api

* **HttpSerlvetRequest**
* **HttpServletResponse**
* **HttpSession**
* **Java.security.Principal**
* **Locale**
* **InputStream**：HttpSerlvetRequest的InputStream ()方法
* **OuputStream**：HttpServletResponse的getOutputStream()方法
* **Reader**：HttpSerlvetRequest的get Reader ()方法
* **Writer**：HttpServletResponse的getWriter()方法

使用时直接写入参数即可

@RequestMapping("/testRequest")

**public** String testRequest(HttpServletRequest request,HttpServletResponse response) {

System.***out***.println("request："+request+"\nresonse："+response);

**return** ***SUCCESS***;

}

1. 添加Map参数★

**添加一个Map类型的参数，向这个Map里添加的键值对都会传入到request域对象中**

@RequestMapping("/testMap")

**public** String testMap(Map<String,Object> map,HttpServletRequest request) {

map.put("time", **new** Date());

**return** ***SUCCESS***;

}