Spring-mvc-servlet.xml配置文件

1. <!-- mvc的注解驱动 -->
2. **<mvc:annotation-driven** **/>**
4. <!--只要定义了扫描器，注解驱动就不需要，扫描器已经有了注解驱动的功能 -->
5. **<context:component-scan** base-package="org.study1.mvc.controller" **/>**
7. <!-- 前缀+ viewName +后缀 -->
8. **<bean**
9. class="org.springframework.web.servlet.view.InternalResourceViewResolver"**>**
10. <!-- WebContent(WebRoot)到某一指定的文件夹的路径 ,如下表示/WEB-INF/view/\*.jsp -->
11. **<property** name="prefix" value="/WEB-INF/view/"**></property>**
12. <!-- 视图名称的后缀 -->
13. **<property** name="suffix" value=".jsp"**></property>**
14. **</bean>**

**3. @controller:标识当前类是控制层的一个具体的实现**  
**4. @requestMapping（“url”,RequestMethod.POST）：放在方法上面用来指定某个方法的路径，当它放在类上的时候相当于命名空间需要组合方法上的requestmapping来访问。**

**1. HttpServletRequest可以直接定义在参数的列表，通过该请求可以传递参数**

url：http://localhost/mvc/test/toPerson.do?name=zhangsan

1. @RequestMapping("/toPerson.do")
2. **public** String toPerson(HttpServletRequest request) {
3. String result = request.getParameter("name");
4. System.out.println(result);
5. **return** "jsp1/index";
6. }

**2.1传递多种不同类型的参数：**

url：http://localhost/mvc/test/toPerson1.do？name=zhangsan&age=14&address=china&birthday=2000-2-11

1. /\*\*
2. \* 传递的参数的名字必须要与实体类的属性set方法后面的字符串匹配的上才能接收到参数，首字符的大小写不区分
3. \* 请求中传的参数只要是能和参数列表里面的变量名或者实体里面的set后面的字符串匹配的上就能接收到 a
4. \*
5. \*/
6. @RequestMapping("/toPerson1.do")
7. **public** String toPerson1(String name, Integer age, String address,
8. Date birthday) {
9. System.out.println(name + " " + age + " " + address + " " + birthday);
10. **return** "jsp1/index";
11. }
13. /\*\*
14. \* 注册时间类型的属性编辑器，将String转化为Date
15. \*/
16. @InitBinder
17. **public** **void** initBinder(ServletRequestDataBinder binder) {
18. binder.registerCustomEditor(Date.**class**, **new** CustomDateEditor(
19. **new** SimpleDateFormat("yyyy-MM-dd"), **true**));
20. }

**2.2传递数组：**

url：http://localhost/mvc/test/toPerson2.do?name=tom&name=jack

1. @RequestMapping("/toPerson2.do")
2. **public** String toPerson2(String[] name) {
3. **for** (String result : name) {
4. System.out.println(result);
5. }
6. **return** "jsp1/index";
7. }

**2.3传递自定义对象（可多个）：**

url：http://localhost/mvc/test/toPerson3.do？name=zhangsan&age=14&address=china&birthday=2000-2-11

 User 定义的属性有：name,age，并且有各自属性的对应的set方法以及toString方法

 Person定义的属性有：name,age.address,birthday，并且有各自属性的对应的set方法以及toString方法

1. /\*\*
2. \*
3. \* 传递的参数的名字必须要与实体类的属性set方法后面的字符串匹配的上才能接收到参数，首字符的大小写不区分
4. \* 请求中传的参数只要是能和参数列表里面的变量名或者实体里面的set后面的字符串匹配的上就能接收到
5. \*
6. \*/
7. @RequestMapping("/toPerson3.do")
8. **public** String toPerson3(Person person, User user) {
9. System.out.println(person);
10. System.out.println(user);
11. **return** "jsp1/index";
12. }

Person [name=zhangsan, age=14, address=china, birthday=Fri Feb 11 00:00:00 CST 2000]  
User [name=zhangsan, age=14]

自动封装了对象，并且被分别注入进来！

三、结果返回

1. /\*\*
2. \* HttpServletRequest可以直接定义在参数的列表，并且带回返回结果
3. \*
4. \*/
5. @RequestMapping("/toPerson41.do")
6. **public** String toPerson41(HttpServletRequest request) **throws** Exception {
7. request.setAttribute("p", newPesion());
8. **return** "index";
9. }
10. /\*\*
11. \*
12. \* 在参数列表中直接定义Model，model.addAttribute("p", person);
13. \* 把参数值放到request类里面去，建议使用
14. \*
15. \*/
16. @RequestMapping("/toPerson44.do")
17. **public** String toPerson44(Model model) **throws** Exception {
18. // 把参数值放到request类里面去
19. model.addAttribute("p", newPesion());
20. **return** "index";
21. }
23. /\*\*

**四、表单提交和重定向**

**1、表单提交：**

**请求方式的指定：@RequestMapping( method=RequestMethod.POST )可以指定请求方式，前台页面就必须要以它制定好的方式来访问，否则出现405错误**

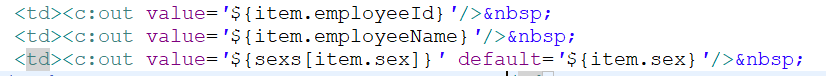
 表单jsp页面：

1. /\*\*
2. \*
3. \* @RequestMapping( method=RequestMethod.POST)
4. \* 可以指定请求方式，前台页面就必须要以它制定好的方式来访问，否则出现405错误 a
5. \*
6. \*/
7. @RequestMapping(value = "/toPerson5.do", method = RequestMethod.POST)
8. **public** String toPerson5(Person person) {
9. System.out.println(person);
10. **return** "jsp1/index";
11. }
12. /\*\*
13. \*
14. \* controller内部重定向
15. \* redirect:加上同一个controller中的requestMapping的值
16. \*
17. \*/
18. @RequestMapping("/redirectToForm.do")
19. **public** String redirectToForm() {
20. **return** "redirect:toform.do";
21. }
23. /\*\*
24. \*
25. \* controller之间的重定向：必须要指定好controller的命名空间再指定requestMapping的值，
26. \* redirect：后必须要加/,是从根目录开始
27. \*/
28. @RequestMapping("/redirectToForm1.do")
29. **public** String redirectToForm1() {
30. //test1表示另一个Controller的命名空间
31. **return** "redirect:/test1/toForm.do";
32. }

返回的数据给jsp界面：

Jstl标签：

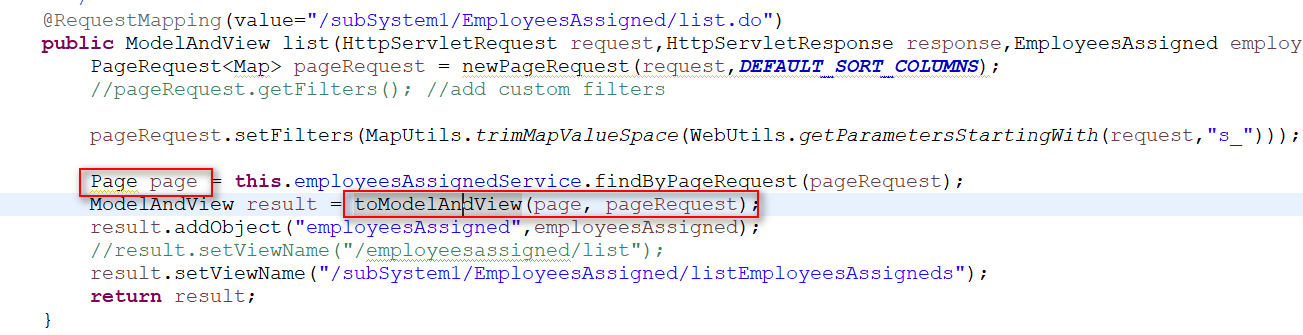


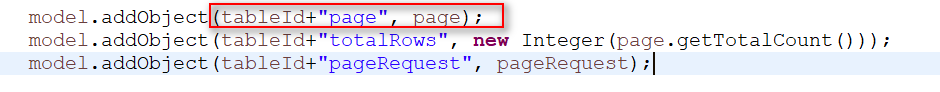


遍历输出page.result中的每个元素

对应controller代码：

Return 一个modelAndView对象，给jsp界面；相当于request.setAttriubte(“param\_name”,”object”);





Ajax接收controller返回的json数据

**@ResponseBody**

作用：

      该注解用于将Controller的方法返回的对象，通过适当的HttpMessageConverter转换为指定格式后，写入到Response对象的body数据区。（把java对象转换成json,xml字符串返回给client端）

后台代码如下，输出一个json格式的字符串返回给前台界面

