

# hexiongyun001

昵称：残阳、  
园龄：1年7个月  
粉丝：0  
关注：6  
+加关注

< 2017年9月 >						
日	一	二	三	四	五	六
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

搜索

## 常用链接

我的随笔  
我的评论  
我的参与  
最新评论  
我的标签  
更多链接

## 我的标签

java学习路线(1)

## 随笔档案

2017年6月 (5)  
2017年5月 (25)  
2017年4月 (16)  
2017年3月 (13)  
2017年2月 (1)

## 阅读排行榜

1. json字符串、json对象、数组 三者之间的转换(2864)  
2. css菜鸟学习之text-align属性，行内元素，块级元素居中详解(2286)  
3. javascript 中的JSON.stringify - 将对象和数组转换为json格式(来源于网络)(1997)

博客园 首页 新随笔 联系 管理 订阅 XML

随笔- 60 文章- 1 评论- 0

## SpringMVC 之@RequestBody 接收Json数组对象

### 1. 摘要

程序流程：

1. 前台使用ajax技术，传递json字符串到后台；
2. 后台使用Spring MVC注解@RequestBody 接受前台传递的json字符串，并返回新的json字符串到前台；
3. 前台接受后台传递过来的json数据，并显示。

大纲

[1. 摘要](#)  
[3. 后台java代码](#)  
[4. 配置](#)  
[web.xml](#)  
[spring-servlet.xml](#)  
[applicationContext.xml](#)  
[5. 结果](#)  
[6. 其他](#)

```
</head>
<body>
    <div style="padding:5px 0;">
        <p>SpringMVC @RequestBody 接收Json数组对象</p>
        <a href="#" class="easyui-Linkbutton" onclick="loadData()">
    </div>
</body>
</html>
```

### 2. 前台界面和js

1. <%@ page language="java" pageEncoding="UTF-8"%>
2. <%
3. String path = request.getContextPath();
4. String basePath = request.getScheme() + "://" + request.getServerName() + ":" + request.getServerPort() + path + "/";
5. response.setHeader("Pragma", "no-cache");
6. response.setHeader("Cache-Control", "no-cache");
7. response.setDateHeader("Expires", 0);
8. %>

4. SpringMVC 之@RequestBody 接收Json数组对象(1326)
5. css菜鸟之HTML 中块级元素设置 height:100% 的实现(1297)

```

9. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

10. <html>

11. <head>

12. <title>SpringMVC+ajax+json</title>

13. <script type="text/javascript">var basePath = "<%=basePath%>";</script>

14. <!-- <link rel="stylesheet" type="text/css" href="<%=basePath%>js/easyui/demo.css" -->

15. <script type="text/javascript" src="<%=basePath%>js/JQuery/jquery.min.js"></script>

16. <script type="text/javascript" src="<%=basePath%>js/test/ajaxAndJson.js"></script>

17.

18. </head>

19. <body>

20. <div style="padding:5px 0;">

21. <p>SpringMVC @RequestBody 接收Json数组对象</p>

22. <a href="#" class="easyui-linkbutton" onclick="loadData()" data-options="iconCls:'icon-add'">@RequestBody 接收Json数组对象</a>

23. </div>

24. </body>

25. </html>

```

```

function loadData() {
    var actionUrl = basePath + "test/isonDataReq.action";
    var params = JSON.stringify([ {
        username : "zhangSan",
        passwd : "123456"
    }, {
        username : "LiSi",
        passwd : "8888"
    } ]);
    $.ajax({
        type: 'POST',
        url : actionUrl,
        data : params,
        dataType : "json", 后台返回的数据类型
        cache : false,
        contentType: "application/json", 必须, 不可少 @RequestBody需要# 需要使用jsonHt
        error : function(textStatus, errorThrown) {
            alert("系统请求错误: " + textStatus);
        },
        success : function(data, textStatus) {
            alert("请求成功==>姓名: "+data.username+" , 密码: "+data.p
        }
    });
}

```

### 3. 后台java代码

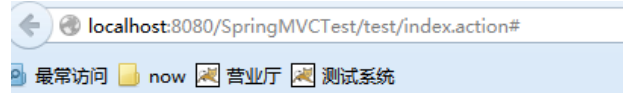
```

@Controller
@RequestMapping(value = "/test")
public class AjaxController {

    /**
     * 返回主页
     * @return
     */
    @RequestMapping(value = "/index.action")
    public String index() {
        return "index";
    }
}

```

<http://localhost:8080/SpringMVCTest/test/index.action> 请求后，返回index.jsp界面，如下图：



SpringMVC @RequestBody 接收Json数组对象

**@RequestBody 接收Json数组对象**

点击后会请求jsonDataReq

```

/**
 * SpringMVC @RequestBody 接收Json数组对象
 * @param persons
 */
@RequestMapping(value = "/jsonDataReq", method = { RequestMethod
@ResponseBody 完成person对象到json数据的转换
public Person jsonDataReq(@RequestBody Person[] persons) {
    for (Person person : persons) {
        System.out.println("用户名: " + person.getUsername() + ",
        + person.getPasswd());
    }

    return new Person("王五", "666666");
}
}

```

这是person.java类

```

package com.ll.model;

public class Person {
    private String username;
    private String passwd;

    public Person() {
        super();
    }
}

```

#### 4. 配置

##### web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://j
  <!-- 从类路径下加载Spring配置文件, classpath关键字特指从类路径下加载 -->
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:applicationContext.xml</param-value>
  </context-param>
  <!-- 负责启动Spring容器的监听器 -->
  <listener><listener-class>org.springframework.web.context.Context
  </listener-class></listener>

  <!-- 配置Spring MVC, 其对应的配置文件为: servlet-name-servlet.xml, 本项目
  <servlet>
    <servlet-name>spring</servlet-name>
    <servlet-class> org.springframework.web.servlet.DispatcherSe
    <load-on-startup>3</load-on-startup>
  </servlet>

  <servlet-mapping>
    <servlet-name>spring</servlet-name>
    <url-pattern>*.action</url-pattern>
  </servlet-mapping>
</web-app>
```

1. <?xml version="1.0" encoding="UTF-8"?>
2. <web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3. xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-
 app\_2\_5.xsd">
4. <!-- 从类路径下加载Spring配置文件, classpath关键字特指从类路径下加载 -->
5. <context-param>
6. <param-name>contextConfigLocation</param-name>
7. <param-value>classpath:applicationContext.xml</param-value>
8. </context-param>
9. <!-- 负责启动Spring容器的监听器 -->
10. <listener><listener-class>org.springframework.web.context.ContextLoaderListener
11. </listener-class></listener>
- 12.
13. <!-- 配置Spring MVC, 其对应的配置文件为: servlet-name-servlet.xml, 本项目的为: spring-
 servlet.xml -->
14. <servlet>
15. <servlet-name>spring</servlet-name>
16. <servlet-class> org.springframework.web.servlet.DispatcherServlet</servlet-class>
17. <load-on-startup>3</load-on-startup>
18. </servlet>
- 19.
20. <servlet-mapping>
21. <servlet-name>spring</servlet-name>
22. <url-pattern>\*.action</url-pattern>
23. </servlet-mapping>
24. </web-app>

## spring-servlet.xml

```

<beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:
        xmlns:context="http://www.springframework.org/schema/context"
        xmlns:mvc="http://www.springframework.org/schema/mvc"
        xsi:schemaLocation="http://www.springframework.org/schema/b
            http://www.springframework.org/schema/beans/spring-beans
            http://www.springframework.org/schema/context
            http://www.springframework.org/schema/context/spring-con
            http://www.springframework.org/schema/mvc
            http://www.springframework.org/schema/mvc/spring-mvc-3.0

        <!-- 扫描web包，应用Spring的注解 -->
        <context:component-scan base-package="com.ll.web"/>

        <mvc:annotation-driven/>

        <!-- 配置视图解析器，将ModelAndView及字符串解析为具体的页面，默认优先级最
        <bean
            class="org.springframework.web.servlet.view.InternalReso
            p:viewClass="org.springframework.web.servlet.view.JstlV
            p:prefix="/jsp/"
            p:suffix=".jsp" />

    </beans>

```

## applicationContext.xml

```

<?xml version="1.0" encoding="UTF-8" ?>
<beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:p=
        xmlns:context="http://www.springframework.org/schema/context"
        xmlns:aop="http://www.springframework.org/schema/aop" xmlns:tx
        xsi:schemaLocation="http://www.springframework.org/schema/bean
            http://www.springframework.org/schema/beans/spring-beans-3.
            http://www.springframework.org/schema/context
            http://www.springframework.org/schema/context/spring-contex
            http://www.springframework.org/schema/tx
            http://www.springframework.org/schema/tx/spring-tx-3.0.xsd
            http://www.springframework.org/schema/aop
            http://www.springframework.org/schema/aop/spring-aop-3.0.xs

        <!-- 扫描类包，将标注Spring注解的类自动转化Bean，同时完成Bean的注入 -->
        <context:component-scan base-package="com.ll.service"/>
        <context:component-scan base-package="com.ll.dao"/>

    </beans>

```

## 5. 结果

tomcat7Server [Remote Java Application] D:\Program Files\Java\jdk1.7.0\bin\javaw.exe (2)

用户名: zhangSan, 密码: 123456

用户名: LiSi, 密码: 8888

SpringMVC @RequestBody 接收Json数组对象

@RequestBody 接收Json数组对象

请求成功==>姓名

好文要顶 关注我 收藏该文



残阳、  
关注 - 6  
粉丝 - 0

0

0

+加关注

« 上一篇: [js数组与 json 的区别](#)  
» 下一篇: [SpringMVC @RequestBody 接收Json数组对象](#)

posted @ 2017-03-16 00:56 残阳、 阅读(1326) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#)网站首页。

最新IT新闻:

- 微软宣布Windows开发者日将于10月10日召开
  - 给Android开发者的31个Pro版进阶小贴士
  - 复旦大学附属肿瘤医院与腾讯战略合作打造“智慧医疗生态”
  - 盖茨怒怼特朗普：削减援助预算会导致更多艾滋病死亡
  - 百度发布“AI科普计划”：百余个关键词特效上线
- » 更多新闻...

最新知识库文章:

- Google 及其云智慧
  - 做到这一点，你也可以成为优秀的程序员
  - 写给立志做码农的大学生
  - 架构腐化之谜
  - 学会思考，而不只是编程
- » 更多知识库文章...

Copyright ©2017 残阳、