[**SpringMVC接收复杂集合参数**](http://jxd-zxf.iteye.com/blog/2072300)

**博客分类：**

* [Spring](http://jxd-zxf.iteye.com/category/272980)

[SpringMVC](http://www.iteye.com/blogs/tag/SpringMVC)[集合参数](http://www.iteye.com/blogs/tag/%E9%9B%86%E5%90%88%E5%8F%82%E6%95%B0)

        Spring MVC在接收集合请求参数时，需要在Controller方法的集合参数里前添加@RequestBody，而@RequestBody默认接收的enctype (MIME编码)是application/json，因此发送POST请求时需要设置请求报文头信息，否则Spring MVC在解析集合请求参数时不会自动的转换成JSON数据再解析成相应的集合。以下列举接收List<String>、List<User>、List<Map<String,Object>>、User[]、User(bean里面包含List)几种较为复杂的集合参数示例：

* 接收List<String>集合参数：

1、页面js代码：

**Js代码**



1. **var** idList = **new** Array();
2. idList.push(“1”);
3. idList.push(“2”);
4. idList.push(“3”);
5. **var** isBatch = **false**;
6. $.ajax({
7. type: "POST",
8. url: "<%=path%>/catalog.do?fn=deleteCatalogSchemes",
9. dataType: 'json',
10. data: {"idList":idList,"isBatch":isBatch},
11. success: **function**(data){
12. …
13. },
14. error: **function**(res){
15. …
16. }
17. });

 2、Controller方法：

**Java代码**



1. @Controller
2. @RequestMapping("/catalog.do")
3. **public** **class** CatalogController {
5. @RequestMapping(params = "fn=deleteCatalogSchemes")
6. @ResponseBody
7. **public** AjaxJson deleteCatalogSchemes(@RequestParam("idList[]") List<String> idList,Boolean isBatch) {
8. …
9. }
10. }

* 接收List<User>、User[]集合参数：

 1、User实体类：

**Java代码**



1. **public** **class** User {
2. **private** String name;
3. **private** String pwd;
4. //省略getter/setter
5. }

2、页面js代码：

**Js代码**



1. **var** userList = **new** Array();
2. userList.push({name: "李四",pwd: "123"});
3. userList.push({name: "张三",pwd: "332"});
4. $.ajax({
5. type: "POST",
6. url: "<%=path%>/catalog.do?fn=saveUsers",
7. data: JSON.stringify(userList),//将对象序列化成JSON字符串
8. dataType:"json",
9. contentType : 'application/json;charset=utf-8', //设置请求头信息
10. success: **function**(data){
11. …
12. },
13. error: **function**(res){
14. …
15. }
16. });

3、Controller方法：

**Java代码**



1. @Controller
2. @RequestMapping("/catalog.do")
3. **public** **class** CatalogController {
5. @RequestMapping(params = "fn=saveUsers")
6. @ResponseBody
7. **public** AjaxJson saveUsers(@RequestBody List<User> userList) {
8. …
9. }
10. }

    如果想要接收User[]数组，只需要把saveUsers的参数类型改为@RequestBody User[] userArray就行了。

* 接收List<Map<String,Object>>集合参数：

 1、页面js代码（不需要User对象了）：

**Js代码**



1. **var** userList = **new** Array();
2. userList.push({name: "李四",pwd: "123"});
3. userList.push({name: "张三",pwd: "332"});
4. $.ajax({
5. type: "POST",
6. url: "<%=path%>/catalog.do?fn=saveUsers",
7. data: JSON.stringify(userList),//将对象序列化成JSON字符串
8. dataType:"json",
9. contentType : 'application/json;charset=utf-8', //设置请求头信息
10. success: **function**(data){
11. …
12. },
13. error: **function**(res){
14. …
15. }
16. });

2、Controller方法：

**Java代码**



1. @Controller
2. @RequestMapping("/catalog.do")
3. **public** **class** CatalogController {
5. @RequestMapping(params = "fn=saveUsers")
6. @ResponseBody
7. **public** AjaxJson saveUsers(@RequestBody List<Map<String,Object>> listMap) {
8. …
9. }
10. }

* 接收User(bean里面包含List)集合参数：

 1、User实体类：

**Java代码**



1. **public** **class** User {
2. **private** String name;
3. **private** String pwd;
4. **private** List<User> customers;//属于用户的客户群
5. //省略getter/setter
6. }

2、页面js代码：

**Js代码**



1. **var** customerArray = **new** Array();
2. customerArray.push({name: "李四",pwd: "123"});
3. customerArray.push({name: "张三",pwd: "332"});
4. **var** user = {};
5. user.name = "李刚";
6. user.pwd = "888";
7. user. customers = customerArray;
8. $.ajax({
9. type: "POST",
10. url: "<%=path%>/catalog.do?fn=saveUsers",
11. data: JSON.stringify(user),//将对象序列化成JSON字符串
12. dataType:"json",
13. contentType : 'application/json;charset=utf-8', //设置请求头信息
14. success: **function**(data){
15. …
16. },
17. error: **function**(res){
18. …
19. }
20. });

  3、Controller方法：

**Java代码**



1. @Controller
2. @RequestMapping("/catalog.do")
3. **public** **class** CatalogController {
5. @RequestMapping(params = "fn=saveUsers")
6. @ResponseBody
7. **public** AjaxJson saveUsers(@RequestBody User user) {
8. List<User> customers = user.getCustomers();
9. …
10. }
11. }