**将json字符串中数组节点转换为泛型list集合(gson)**

原创 2016年03月15日 10:39:31

* 2932

开发中需要将json中节点单独取出某个节点，解析获取数据，此处对json字符串节点中数组节点，转换为泛型list集合，使用gson。

google的gson的pom文件和jar：[gson](http://mvnrepository.com/artifact/com.google.code.gson/gson/2.6.2)

**代码如下**：

**[java]** [view plain](http://blog.csdn.net/typa01_kk/article/details/50894563) [copy](http://blog.csdn.net/typa01_kk/article/details/50894563)

1. **package** my.demo.java.z.generic;
3. **import** java.util.ArrayList;
4. **import** java.util.List;
6. **import** org.apache.commons.lang3.StringUtils;
8. **import** com.google.gson.GsonBuilder;
9. **import** com.google.gson.reflect.TypeToken;
11. /\*\*
12. \* Gson获取json中的数组节点.<br/>
13. \* 将数组节点转换为list集合<泛型>.<br/>
14. \* 泛型类，是在实例化类的时候指明泛型的具体类型.<br/>
15. \* 泛型方法，是在调用方法的时候指明泛型的具体类型.<br/>
16. \* @author Tony\_tian
17. \* @time 2016-03-15 09:44:32
18. \* \*/
19. **public** **class** GenericMethodsJson2List {
21. **public** **static** **final** String START\_ARRAY = "[";
22. **public** **static** **final** String END\_ARRAY = "]";
24. /\*\*
25. \* Gson获取json中的数组节点.转换为list集合<泛型>.<br/>
26. \* @param <T> 声明一个泛型T
27. \* @param List<T> 指明该方法的返回值为泛型T代表类型的list集合
28. \* @param jsonStr json中的数组节点
29. \* @param tC 指明泛型T的具体类型，用来创建泛型T代表的类的对象
30. \* @return 1:正常返回泛型代表具体类型的list集合，2:转换错误则返回null
31. \* \*/
32. @SuppressWarnings("unchecked")
33. **public** **static** **final** <T> List<T> json2listT(String jsonStr, Class<T> tC) {
34. //json字符串不能为空
35. **if**(StringUtils.isBlank(jsonStr)) **return** **null**;
36. //json字符串必须为数组节点类型
37. **if**(!(jsonStr.startsWith(START\_ARRAY) && jsonStr.endsWith(END\_ARRAY))) **return** **null**;
38. List<T> listT = **null**;
39. **try** {
40. //创建泛型对象
41. T t =  tC.newInstance();
42. //利用类加载加载泛型的具体类型
43. Class<T> classT = (Class<T>) Class.forName(t.getClass().getName());
44. List<Object> listObj = **new** ArrayList<Object>();
45. //将数组节点中json字符串转换为object对象到Object的list集合
46. listObj = **new** GsonBuilder().create().fromJson(jsonStr, **new** TypeToken<List<Object>>(){}.getType());
47. //转换未成功
48. **if**(listObj == **null** || listObj.isEmpty()) **return** **null**;
49. listT = **new** ArrayList<T>();
50. //将Object的list中的的每一个元素中的json字符串转换为泛型代表的类型加入泛型代表的list集合返回
51. **for** (Object obj : listObj) {
52. T perT = **new** GsonBuilder().create().fromJson(obj.toString(), classT);
53. listT.add(perT);
54. }
55. } **catch** (Exception e) {
56. e.printStackTrace();
57. }
58. **return** listT;
59. }

62. **public** **static** **void** main(String[] args) {
63. String jsonStr = "[{\"username\":\"点点\",\"memberNo\":1},{\"username\":\"didi\",\"memberNo\":2},{\"username\":\"卡卡\",\"memberNo\":3}]";
64. List<Member> listp = json2listT(jsonStr, Member.**class**);
65. **for** (Member member : listp) {
66. System.out.println(member.toString());
67. }
68. }
69. }
71. **class** Member{
72. **private** Integer memberNo;
73. **private** String username;
75. **public** Integer getMemberNo() {
76. **return** memberNo;
77. }
79. **public** String getUsername() {
80. **return** username;
81. }
83. **public** **void** setMemberNo(Integer memberNo) {
84. **this**.memberNo = memberNo;
85. }
87. **public** **void** setUsername(String username) {
88. **this**.username = username;
89. }
91. @Override
92. **public** String toString() {
93. StringBuilder su = **new** StringBuilder();
94. su.append("\"memberNo\"").append(":").append(**this**.getMemberNo());
95. su.append(",").append("\"username\"").append(":").append(**this**.getUsername());
96. **return** su.toString();
97. }
98. }

**main执行结果**：

**[java]** [view plain](http://blog.csdn.net/typa01_kk/article/details/50894563) [copy](http://blog.csdn.net/typa01_kk/article/details/50894563)

1. "memberNo":1,"username":点点
2. "memberNo":2,"username":didi
3. "memberNo":3,"username":卡卡

版权声明：本文为博主原创文章，未经博主允许不得转载。