

Json转换利器Gson之实例五-实际开发中的特殊需求处理

标签: [Gson](#) [json](#) [JSON](#) [Json](#) [实例](#)

2012-06-22 22:02

15953人阅读

[评论\(10\)](#)

[收藏](#)

[举报](#)

分类:

[Json \(5\)](#)

版权声明：
本

文为博主原创文章，未经博主允许不得转载。

前面四篇博客基本上可以满足我们处理的绝大多数需求,但有时项目中对json有特殊的格式规定.比如下面的json串解析:

```
[{"tableName":"students","tableData":[{"id":1,"name":"李坤","birthDay":"Jun 22, 2012 9:54:49 PM"}, {"id":2,"name":"曹贵生","birthDay":"Jun 22, 2012 9:54:49 PM"}, {"id":3,"name":"柳波","birthDay":"Jun 22, 2012 9:54:49 PM"}]}, {"tableName":"teachers","tableData":[{"id":1,"name":"米老师","title":"教授"}, {"id":2,"name":"丁老师","title":"讲师"}]}
```

分析之后我们发现使用前面博客中用到的都不好处理上面的json串.请看本文是如何处理的吧:

实体类:

[java]

```
01. import java.util.Date;
02.
03. public class Student {
04.     private int id;
05.     private String name;
06.     private Date birthDay;
07.
08.     public int getId() {
09.         return id;
10.     }
11.
12.     public void setId(int id) {
13.         this.id = id;
14.     }
15.
16.     public String getName() {
17.         return name;
18.     }
19.
20.     public void setName(String name) {
21.         this.name = name;
22.     }
23.
24.     public Date getBirthDay() {
25.         return birthDay;
26.     }
```

下载

```
27.  
28. public void setBirthDay(Date birthDay) {  
29.     this.birthDay = birthDay;  
30. }  
31.  
32. @Override  
33. public String toString() {  
34.     return "Student [birthDay=" + birthDay + ", id=" + id + ", name=" +  
35.         + name + " ]";  
36. }  
37.  
38. }
```

[java]

```
01. public class Teacher {  
02.     private int id;  
03.  
04.     private String name;  
05.  
06.     private String title;  
07.  
08.     public int getId() {  
09.         return id;  
10.     }  
11.  
12.     public void setId(int id) {  
13.         this.id = id;  
14.     }  
15.  
16.     public String getName() {  
17.         return name;  
18.     }  
19.  
20.     public void setName(String name) {  
21.         this.name = name;  
22.     }  
23.  
24.     public String getTitle() {  
25.         return title;  
26.     }  
27.  
28.     public void setTitle(String title) {  
29.         this.title = title;  
30.     }  
31.  
32. @Override  
33. public String toString() {  
34.     return "Teacher [id=" + id + ", name=" + name + ", title=" + title  
35.         + " ]";  
36. }  
37.  
38. }
```

📄 载

注意这里定义了一个TableData实体类:

```
[java]

01. import java.util.List;
02.
03. public class TableData {
04.
05.     private String tableName;
06.
07.     private List tableData;
08.
09.     public String getTableName() {
10.         return tableName;
11.     }
12.
13.     public void setTableName(String tableName) {
14.         this.tableName = tableName;
15.     }
16.
17.     public List getTableData() {
18.         return tableData;
19.     }
20.
21.     public void setTableData(List tableData) {
22.         this.tableData = tableData;
23.     }
24. }
```

测试类:

(仔细看将json转回为对象的实现,这里经过两次转化,第一次转回的结果是map不是我们所期望的对象,对map再次转为json后再转为对象,我引用的是Gson2.1的jar处理正常,好像使用Gson1.6的jar会报错,所以建议用最新版本)

```
[java]

01. import java.util.ArrayList;
02. import java.util.Date;
03. import java.util.List;
04.
05. import com.google.gson.Gson;
06. import com.google.gson.reflect.TypeToken;
07.
08. public class GsonTest5 {
09.
10.     /**
11.      * @param args
12.      */
13.     public static void main(String[] args) {
14.         // 对象转为Json-->start
15.         Student student1 = new Student();
16.         student1.setId(1);
17.         student1.setName("李坤");
```

```
18.         student1.setBirthDay(new Date());
19.
20.         Student student2 = new Student();
21.         student2.setId(2);
22.         student2.setName("曹贵生");
23.         student2.setBirthDay(new Date());
24.
25.         Student student3 = new Student();
26.         student3.setId(3);
27.         student3.setName("柳波");
28.         student3.setBirthDay(new Date());
29.
30.         List<Student> stulist = new ArrayList<Student>();
31.         stulist.add(student1);
32.         stulist.add(student2);
33.         stulist.add(student3);
34.
35.         Teacher teacher1 = new Teacher();
36.         teacher1.setId(1);
37.         teacher1.setName("米老师");
38.         teacher1.setTitle("教授");
39.
40.         Teacher teacher2 = new Teacher();
41.         teacher2.setId(2);
42.         teacher2.setName("丁老师");
43.         teacher2.setTitle("讲师");
44.         List<Teacher> teacherList = new ArrayList<Teacher>();
45.         teacherList.add(teacher1);
46.         teacherList.add(teacher2);
47.
48.         TableData td1 = new TableData();
49.         td1.setTableName("students");
50.         td1.setTableData(stulist);
51.
52.         TableData td2 = new TableData();
53.         td2.setTableName("teachers");
54.         td2.setTableData(teacherList);
55.
56.         List<TableData> tdList = new ArrayList<TableData>();
57.         tdList.add(td1);
58.         tdList.add(td2);
59.         Gson gson = new Gson();
60.         String s = gson.toJson(tdList);
61.
62.         System.out.println(s);
63.         // 结果:[{"tableName":"students","tableData":[{"id":1,"name":"李
        坤","birthDay":"Jun 22, 2012 10:44:16 AM"}, {"id":2,"name":"曹贵
        生","birthDay":"Jun 22, 2012 10:44:16 AM"}, {"id":3,"name":"柳
        波","birthDay":"Jun 22, 2012 10:44:16 AM"}]}, {"tableName":"teachers","tableData":
        [{"id":1,"name":"米老师","title":"教授"}, {"id":2,"name":"丁老师","title":"讲师"}]}]
64.         // 对象转为Json-->end
65.
66.         // //////////////////////////////////////
67.
68.         // 将json转为数据-->start
69.         List<TableData> tableDatas2 = gson.fromJson(s,
70.             new TypeToken<List<TableData>>() {
```

```

71.         }.getType());
72.         for (int i = 0; i < tableDatas2.size(); i++) {
73.             TableData entityData = tableDatas2.get(i);
74.             String tableName = entityData.getTableName();
75.             List tableData = entityData.getTableData();
76.             String s2 = gson.toJson(tableData);
77.             // System.out.println(s2);
78.             // System.out.println(entityData.getData());
79.             if (tableName.equals("students")) {
80.                 System.out.println("students");
81.                 List<Student> retStuList = gson.fromJson(s2,
82.                     new TypeToken<List<Student>>() {
83.                         }.getType());
84.                 for (int j = 0; j < retStuList.size(); j++) {
85.                     System.out.println(retStuList.get(j));
86.                 }
87.
88.             } else if (tableName.equals("teachers")) {
89.                 System.out.println("teachers");
90.                 List<Teacher> retTchrList = gson.fromJson(s2,
91.                     new TypeToken<List<Teacher>>() {
92.                         }.getType());
93.                 for (int j = 0; j < retTchrList.size(); j++) {
94.                     System.out.println(retTchrList.get(j));
95.                 }
96.             }
97.         }
98.
99.         // Json转为对象-->end
100.     }
101. }

```

输出结果:

[plain]

```

01. [{"tableName":"students","tableData":[{"id":1,"name":"李
02. 坤","birthDay":"Jun 22, 2012 10:04:12 PM"}, {"id":2,"name":"曹贵
03. 生","birthDay":"Jun 22, 2012 10:04:12 PM"}, {"id":3,"name":"柳
04. 波","birthDay":"Jun 22, 2012 10:04:12 PM"}]}, {"tableName":"teachers","tableData":
05. [{"id":1,"name":"米老师","title":"教授"}, {"id":2,"name":"丁老师","title":"讲师"}]}]
06. students
07. Student [birthDay=Fri Jun 22 22:04:12 CST 2012, id=1, name=李坤]
08. Student [birthDay=Fri Jun 22 22:04:12 CST 2012, id=2, name=曹贵生]
09. Student [birthDay=Fri Jun 22 22:04:12 CST 2012, id=3, name=柳波]
10. teachers
11. Teacher [id=1, name=米老师, title=教授]
12. Teacher [id=2, name=丁老师, title=讲师]

```

1 载 4

Json转换利器Gson之实例一-简单对象转化和带泛型的List转化

(http://blog.csdn.net/lk_blog/article/details/7685169)

Json转换利器Gson之实例二-Gson注解和GsonBuilder

(http://blog.csdn.net/lk_blog/article/details/7685190)

Json转换利器Gson之实例三-Map处理(上)

(http://blog.csdn.net/lk_blog/article/details/7685210)

Json转换利器Gson之实例四-Map处理(下)

(http://blog.csdn.net/lk_blog/article/details/7685224)

Json转换利器Gson之实例五-实际开发中的特殊需求处理

(http://blog.csdn.net/lk_blog/article/details/7685237)

Json转换利器Gson之实例六-注册TypeAdapter及处理Enum类型

(http://blog.csdn.net/lk_blog/article/details/7685347)

实例代码下载: http://download.csdn.net/detail/lk_blog/4387822