**Google Gson的使用方法,实现Json结构的相互转换**

原创 2015年01月16日 12:03:45

* 26956

在Java开发中，有时需要保存一个数据结构成字符串，可能你会考虑用Json，但是当Json字符串转换成Java对象时，转换成的是JsonObject，并不是你想要的Class类型的对象，操作起来就很不是愉悦，下面说的就可以解决了这种问题。

首先，需要把Google的Gson的Jar包导入到项目中，这个导入包的简单步骤就不展示了，Gson的下载链接：http://download.csdn.net/detail/qxs965266509/8367275

现在，我先自定义一个Class类

**[java]** [view plain](http://blog.csdn.net/qxs965266509/article/details/42774691) [copy](http://blog.csdn.net/qxs965266509/article/details/42774691)

1. **public** **class** Student {
2. **public** **int** id;
3. **public** String nickName;
4. **public** **int** age;
5. **public** ArrayList<String> books;
6. **public** HashMap<String, String> booksMap;
7. }

案例一，案例二，案例三都是把Java的Class对象使用Gson转换成Json的字符串

案例一：

仅包含基本数据类型的数据结构

**[java]** [view plain](http://blog.csdn.net/qxs965266509/article/details/42774691) [copy](http://blog.csdn.net/qxs965266509/article/details/42774691)

1. Gson gson = **new** Gson();
2. Student student = **new** Student();
3. student.id = 1;
4. student.nickName = "乔晓松";
5. student.age = 22;
6. student.email = "965266509@qq.com";
7. Log.e("MainActivity", gson.toJson(student));

输出结果是 ：

**[java]** [view plain](http://blog.csdn.net/qxs965266509/article/details/42774691) [copy](http://blog.csdn.net/qxs965266509/article/details/42774691)

1. {"email":"965266509@qq.com","nickName":"乔晓松","id":1,"age":22}

案例二：

除了基本数据类型还包含了List集合

**[java]** [view plain](http://blog.csdn.net/qxs965266509/article/details/42774691) [copy](http://blog.csdn.net/qxs965266509/article/details/42774691)

1. Gson gson = **new** Gson();
2. Student student = **new** Student();
3. student.id = 1;
4. student.nickName = "乔晓松";
5. student.age = 22;
6. student.email = "965266509@qq.com";
7. ArrayList<String> books = **new** ArrayList<String>();
8. books.add("数学");
9. books.add("语文");
10. books.add("英语");
11. books.add("物理");
12. books.add("化学");
13. books.add("生物");
14. student.books = books;
15. Log.e("MainActivity", gson.toJson(student));

输出结果是 ：

**[html]** [view plain](http://blog.csdn.net/qxs965266509/article/details/42774691) [copy](http://blog.csdn.net/qxs965266509/article/details/42774691)

1. {"books":["数学","语文","英语","物理","化学","生物"],"email":"965266509@qq.com","nickName":"乔晓松","id":1,"age":22}

案例三：

除了基本数据类型还包含了List和Map集合

**[java]** [view plain](http://blog.csdn.net/qxs965266509/article/details/42774691) [copy](http://blog.csdn.net/qxs965266509/article/details/42774691)

1. Gson gson = **new** Gson();
2. Student student = **new** Student();
3. student.id = 1;
4. student.nickName = "乔晓松";
5. student.age = 22;
6. student.email = "965266509@qq.com";
7. ArrayList<String> books = **new** ArrayList<String>();
8. books.add("数学");
9. books.add("语文");
10. books.add("英语");
11. books.add("物理");
12. books.add("化学");
13. books.add("生物");
14. student.books = books;
15. HashMap<String, String> booksMap = **new** HashMap<String, String>();
16. booksMap.put("1", "数学");
17. booksMap.put("2", "语文");
18. booksMap.put("3", "英语");
19. booksMap.put("4", "物理");
20. booksMap.put("5", "化学");
21. booksMap.put("6", "生物");
22. student.booksMap = booksMap;
23. Log.e("MainActivity", gson.toJson(student));

输出结果是 ：

**[java]** [view plain](http://blog.csdn.net/qxs965266509/article/details/42774691) [copy](http://blog.csdn.net/qxs965266509/article/details/42774691)

1. {"books":["数学","语文","英语","物理","化学","生物"],"booksMap":{"3":"英语","2":"语文","1":"数学","6":"生物","5":"化学","4":"物理"},"email":"965266509@qq.com","nickName":"乔晓松","id":1,"age":22}

案例四：  
把案例三输出的字符串使用Gson转换成Student对象

**[java]** [view plain](http://blog.csdn.net/qxs965266509/article/details/42774691) [copy](http://blog.csdn.net/qxs965266509/article/details/42774691)

1. Gson gson = **new** Gson();
2. Student student = **new** Student();
3. student.id = 1;
4. student.nickName = "乔晓松";
5. student.age = 22;
6. student.email = "965266509@qq.com";
7. ArrayList<String> books = **new** ArrayList<String>();
8. books.add("数学");
9. books.add("语文");
10. books.add("英语");
11. books.add("物理");
12. books.add("化学");
13. books.add("生物");
14. student.books = books;
15. HashMap<String, String> booksMap = **new** HashMap<String, String>();
16. booksMap.put("1", "数学");
17. booksMap.put("2", "语文");
18. booksMap.put("3", "英语");
19. booksMap.put("4", "物理");
20. booksMap.put("5", "化学");
21. booksMap.put("6", "生物");
22. student.booksMap = booksMap;
23. String result = gson.toJson(student);
25. Student studentG = gson.fromJson(result, Student.**class**);
27. Log.e("MainActivity", "id:" + studentG.id);
28. Log.e("MainActivity", "nickName:" + studentG.nickName);
29. Log.e("MainActivity", "age:" + studentG.age);
30. Log.e("MainActivity", "email:" + studentG.email);
31. Log.e("MainActivity", "books size:" + studentG.books.size());
32. Log.e("MainActivity", "booksMap size:" + studentG.booksMap.size());

输出结果是 ：

**[html]** [view plain](http://blog.csdn.net/qxs965266509/article/details/42774691) [copy](http://blog.csdn.net/qxs965266509/article/details/42774691)

1. id:1
2. nickName:乔晓松
3. age:22
4. email:965266509@qq.com
5. books size:6
6. booksMap size:6

通过这4个案例我解决你一定就把Gson的基本用法学会了，当然我们的需求可能需要把List或者Map等集合的泛型换成我们自定义个class，这也是可以解决的，请看案例

案例五：

泛型的使用

**[html]** [view plain](http://blog.csdn.net/qxs965266509/article/details/42774691) [copy](http://blog.csdn.net/qxs965266509/article/details/42774691)

1. public HashMap**<String**,Book**>** booksMap;
3. public class Book{
4. public int id;
5. public String name;
6. }

把booksMap转换成字符串和上面的案例是一样的，但是booksMap的Json字符串换成booksMap的实例对象就有点不同了，因为booksMap有自定义的泛型

**[html]** [view plain](http://blog.csdn.net/qxs965266509/article/details/42774691) [copy](http://blog.csdn.net/qxs965266509/article/details/42774691)

1. HashMap**<String**, Book**>** booksMap = gson.fromJson(result, new TypeToken**<HashMap<String**, Book**>>**() { }.getType());

如果什么疑问，请到我的博客列表页面，QQ或者邮件联系我！如有转载请著名来自<http://blog.csdn.net/qxs965266509>

[源代码下载链接：http://download.csdn.net/detail/qxs965266509/8367689](http://download.csdn.net/detail/qxs965266509/8367689)

版权声明：本文为博主原创文章，未经博主允许不得转载。来自：http://blog.csdn.net/qxs965266509