



链滴

Json 数据解析之 FastJson 解析

作者: [zml2015](#)

原文链接: <https://ld246.com/article/1484219763571>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

之前写了关于org.json 以及gson解析 json格式数据的相关博文，今天再补充一下使用里巴巴工程师类库fastjson解析json数据，fastjson和谷歌的gson很像，但是比谷歌的gson做的优化好，因为至少我在[测试](https://ld246.com/forward?goto=http%3A%2F%2Flib.csdn.net%2Fbase%2Fsoftwaretest "软件测试知识库")的时候并没有出现谷歌gson包的[Java](https://ld246.com/forward?goto=http%3A%2F%2Flib.csdn.net%2Fbase%2Fjavase "Java SE知识库")泛型擦除的问题，

即 com.google.gson.JsonSyntaxException:

 java.lang.IllegalStateException:

Expected Person but was BEGIN_OBJECT 等类似的问题

和之前一样，仍然将fastjson解析常用的四种格式的方法写成了一个工具类，直接上工类源码：

```
package com.zml.utils;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import com.alibaba.fastjson.JSON;
import com.alibaba.fastjson.TypeReference;
import com.google.gson.Gson;
import com.zml.pojo.Person;
/**


- @author 郑明亮
- @Time: 2016年7月18日上午9:53:16
- @version 1.0


*/
public class FastJsonTools {
    public FastJsonTools() {
        throw new UnsupportedOperationException("不能被实例化");
    }
    /**TODO 转换成 json 格式的字符串
    

- @param object 要转换的对象
- @return


*/
    public static String createJsonString(Object object) {
        String jsonString = JSON.toJSONString(object);
        return jsonString;
    }
    /**将 json 字符串转换为指定的 bean 对象
    

- @param jsonString
- @param cls 要转换为对象的类型
- @return


*/
    public static <T> T createJsonBean(String jsonString, Class<T> cls) {
```

```

T t = JSON.parseObject(jsonString, cls);<br>
return t;<br>
}/li>
</ul>
<p>/**将 json 字符串转换为 List<T>;</p>
<ul>
<li>@param jsonString</li>
<li>@param cls</li>
<li>@return<br>
*/<br>
public static <T> List<T> createJsonToListBean(String jsonString,<br>
Class<T> cls) {<br>
List<T> list = null;<br>
list = JSON.parseArray(jsonString, cls);<br>
return list;<br>
}</li>
</ul>
<p>/**将 json 字符串转换为 List<Map<String, Object>>;</p>
<ul>
<li>@param jsonString</li>
<li>@return<br>
*/<br>
public static List<Map<String, Object>> createJsonToListMap(<br>
String jsonString) {<br>
List<Map<String, Object>> list2 = JSON.parseObject(jsonString,<br>
new TypeReference<List<Map<String, Object>>>() {<br>
});<br>
return list2;<br>
}</li>
</ul>
<p>/**将 json 字符串转换为 List<String>;</p>
<ul>
<li>@param jsonString</li>
<li>@return<br>
*/<br>
public static List<String> createJsonToListString(String jsonString) {<br>
List<String> list2 = JSON.parseObject(jsonString,<br>
new TypeReference<List<String>>() {<br>
});<br>
return list2;<br>
}</li>
</ul>
</li>
</ul>
<p><br>
</p></pre>
<p><span><a href="https://ld246.com/forward?goto=http%3A%2F%2Fdownload.csdn.net%2Fdetail%2Fzml_2015%2F9578748" target="_blank" rel="nofollow ugc">Json_Gson_FastJson析服务器端源码 下载</a></span></p>
<p><span>&nbsp;</span></p>
<p>&nbsp;</p>
<p><span><a href="https://ld246.com/forward?goto=http%3A%2F%2Fdownload.csdn.net%2Fdetail%2Fzml_2015%2F9578799" target="_blank" rel="nofollow ugc">FastJson解析</a><a href="https://ld246.com/forward?goto=http%3A%2F%2Flib.csdn.net%2Fbase%2Fandroid" cla

```

s="replace_word" title="Android知识库" target="_blank" rel="nofollow ugc">Android客
端源码下载</p>
<p> </p>
<p>关于Gson解析的博文 请点击这里</p>
<p> </p>
<p>关于org.json解析的博文 请点击这里</p>
<p> </p>
<p> </p>
<p> </p>
<p> </p>