

## 通过Gson来处理json数据

### 首先贴上布局文件

```
1. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2.     xmlns:tools="http://schemas.android.com/tools"
3.     android:layout_width="match_parent"
4.     android:layout_height="match_parent"
5.     android:orientation="vertical"
6.     android:paddingBottom="@dimen/activity_vertical_margin"
7.     android:paddingLeft="@dimen/activity_horizontal_margin"
8.     android:paddingRight="@dimen/activity_horizontal_margin"
9.     android:paddingTop="@dimen/activity_vertical_margin"
10.    tools:context="com.example.helloworld.MainActivity" >
11.
12.    <EditText
13.        android:id="@+id/searchcontent"
14.        android:layout_width="wrap_content"
15.        android:layout_height="wrap_content"
16.
17.        android:hint="输入要查询的单词/文字....."
18.    />
19.    <Button
20.        android:id="@+id/btn_search"
21.        android:layout_width="wrap_content"
22.        android:layout_height="wrap_content"
23.
24.        android:text="Search"
25.
26.    />
27.    <ScrollView
28.        android:layout_width="fill_parent"
29.        android:layout_height="wrap_content"
30.        android:scrollbars="none"
31.    >
32.        <TextView
33.            android:id="@+id/searchresult"
34.            android:layout_width="fill_parent"
35.            android:layout_height="wrap_content"
36.        />
37.
38.    </ScrollView>
39.
40. </LinearLayout>
```

### 解析有道词典返回json所需的类

```
1. import java.util.List;
2.
3. public class Word {
4.
```

```

5.     private String errorCode;
6.     private String query;
7.     private List<String> translation;
8.     private Basic basic;
9.     public String getErrorCode() {
10.         return errorCode;
11.     }
12.     public void setErrorCode(String errorCode) {
13.         this.errorCode = errorCode;
14.     }
15.     public String getQuery() {
16.         return query;
17.     }
18.     public void setQuery(String query) {
19.         this.query = query;
20.     }
21.     public List<String> getTranslation() {
22.         return translation;
23.     }
24.     public void setTranslation(List<String> translation) {
25.         this.translation = translation;
26.     }
27.     public Basic getBasic() {
28.         return basic;
29.     }
30.     public void setBasic(Basic basic) {
31.         this.basic = basic;
32.     }
33.
34.     public List<Web> getWeb() {
35.         return web;
36.     }
37.
38.     @Override
39.     public String toString() {
40.         return "Word [errorCode=" + errorCode + ", query=" + query
41.             + ", translation=" + translation + ", basic=" + basic.toString()
42.             + ", web=" + web.toString() + "]";
43.     }
44.     public void setWeb(List<Web> web) {
45.         this.web = web;
46.     }
47.
48.     private List<Web> web;
49.
50. }

```

```

1.
2. import java.util.List;
3.
4. import com.google.gson.annotations.SerializedName;
5.

```

```

6. public class Basic {
7.     @SerializedName("uk-phonetic")
8.     private String uk_phonetic;
9.
10.    @Override
11.    public String toString() {
12.        return "Basic [uk_phonetic=" + uk_phonetic + ", us_phonetic="
13.            + us_phonetic + ", phonetic=" + phonetic + ", explains="
14.            + explains + "]";
15.    }
16.    @SerializedName("us-phonetic")
17.    private String us_phonetic;
18.    private String phonetic;
19.    public String getPhonetic() {
20.        return phonetic;
21.    }
22.    public void setPhonetic(String phonetic) {
23.        this.phonetic = phonetic;
24.    }
25.    private List<String> explains;
26.    public String getUk_phonetic() {
27.        return uk_phonetic;
28.    }
29.    public void setUk_phonetic(String uk_phonetic) {
30.        this.uk_phonetic = uk_phonetic;
31.    }
32.    public String getUs_phonetic() {
33.        return us_phonetic;
34.    }
35.    public void setUs_phonetic(String us_phonetic) {
36.        this.us_phonetic = us_phonetic;
37.    }
38.    public List<String> getExplains() {
39.        return explains;
40.    }
41.    public void setExplains(List<String> explains) {
42.        this.explains = explains;
43.    }
44. }

```

```

1.
2. import java.util.List;
3.
4. public class Web {
5.
6.     private String key;
7.     @Override
8.     public String toString() {
9.         return "Web [key=" + key + ", value=" + value + "]";
10.    }
11.    private List<String> value;
12.    public String getKey() {

```

```

13.         return key;
14.     }
15.     public void setKey(String key) {
16.         this.key = key;
17.     }
18.     public List<String> getValue() {
19.         return value;
20.     }
21.     public void setValue(List<String> value) {
22.         this.value = value;
23.     }
24. }

```

主程序：

```

1.  import java.io.BufferedReader;
2.  import java.io.InputStream;
3.  import java.io.InputStreamReader;
4.  import java.util.List;
5.
6.  import org.apache.http.HttpEntity;
7.  import org.apache.http.HttpResponse;
8.  import org.apache.http.client.HttpClient;
9.  import org.apache.http.client.methods.HttpGet;
10. import org.apache.http.impl.client.DefaultHttpClient;
11. import org.apache.http.util.EncodingUtils;
12. import org.apache.http.util.EntityUtils;
13. import org.json.JSONArray;
14. import org.json.JSONObject;
15.
16. import com.google.gson.Gson;
17. import com.google.gson.reflect.TypeToken;
18.
19. import android.app.Activity;
20. import android.os.Bundle;
21. import android.os.Handler;
22. import android.os.Message;
23. import android.view.Menu;
24. import android.view.MenuItem;
25. import android.view.View;
26. import android.view.View.OnClickListener;
27. import android.widget.Button;
28. import android.widget.EditText;
29. import android.widget.TextView;
30. import android.widget.Toast;
31.
32. public class MainActivity extends Activity {
33.
34.     private static final int SHOW_RESPONSE = 0;
35.     private static final int SHOW_ERROR = 1;
36.     private EditText searchText;
37.     private Button btn_search;

```

```

38. private TextView searchresult;
39. // ////
40. private String YouDaoBaseUrl = "http://fanyi.youdao.com/openapi.do";
41. private String YouDaoKeyFrom = "y3u3q0329";
42. private String YouDaokey = "825755169";
43. private String YouDaoType = "data";
44. private String YouDaoDoctype = "json";
45. private String YouDaoVersion = "1.1";
46.
47. private Handler handler = new Handler() {
48.     public void handleMessage(android.os.Message msg) {
49.         switch (msg.what) {
50.             case SHOW_RESPONSE:
51.                 String response = (String) msg.obj;
52.                 // 在这里进行UI操作，将结果显示到界面上
53.                 searchresult.setText(response);
54.                 Toast.makeText(getApplicationContext(),
55.                     "here" + (String) msg.obj, Toast.LENGTH_SHORT).show();
56.                 break;
57.             case SHOW_ERROR:
58.                 Toast.makeText(getApplicationContext(),
59.                     "here" + (String) msg.obj, Toast.LENGTH_SHORT).show();
60.                 break;
61.             default:
62.                 break;
63.         }
64.     };
65. };
66.
67. @Override
68. protected void onCreate(Bundle savedInstanceState) {
69.     super.onCreate(savedInstanceState);
70.     setContentView(R.layout.activity_main);
71.     ini();
72. }
73.
74. private void ini() {
75.     searchText = (EditText) findViewById(R.id.searchcontent);
76.     btn_search = (Button) findViewById(R.id.btn_search);
77.     searchresult = (TextView) findViewById(R.id.searchresult);
78.     btn_search.setOnClickListener(new searchListener());
79. }
80.
81.
82. class searchListener implements OnClickListener {
83.     @Override
84.     public void onClick(View v) {
85.         // TODO Auto-generated method stub
86.         String YouDaoSearchContent = searchText.getText().toString().trim();
87.         if (YouDaoSearchContent.equals("")) {
88.             Toast.makeText(MainActivity.this, "请输入要查询的内容！",
89.                 Toast.LENGTH_LONG).show();
90.         } else {
91.

```

```

92.         String YouDaoUrl = YouDaoBaseUrl + "?keyfrom=" + YouDaoKeyFrom
93.             + "&key=" + YouDaokey + "&type=" + YouDaoType
94.             + "&doctype=" + YouDaoDoctype + "&version="
95.             + YouDaoVersion + "&q=" + YouDaoSearchContent;
96.
97.         sendRequestWithHttpClient(YouDaoUrl);
98.
99.     }
100. }
101.
102. private void sendRequestWithHttpClient(final String youDaoUrl) {
103.
104.     new Thread(new Runnable() {
105.
106.         @Override
107.         public void run() {
108.             // TODO Auto-generated method stub
109.
110.             try {
111.                 // httpget会自动处理重定向问题
112.                 HttpClient httpClient = new DefaultHttpClient();
113.                 HttpGet httpGet = new HttpGet(youDaoUrl);
114.                 HttpResponse httpResponse = httpClient.execute(httpGet);
115.                 // 200 成功 302 重定向(页面跳转) 404资源没找到 5xx 服务器内部错
116.                 int code = httpResponse.getStatusLine().getStatusCode();
117.
118.                 if (code == 200) {
119.
120.                     BufferedReader input1 = new BufferedReader(
121.                         new InputStreamReader(httpResponse
122.                             .getEntity().getContent()));
123.                     StringBuilder sb = new StringBuilder();
124.
125.                     for (String s = input1.readLine(); s != null; s = input
126.                         .readLine()) {
127.                         sb.append(s);
128.                     }
129.                     parseJSONWithGSON(sb.toString());
130.                 } else {
131.
132.                     Message message = new Message();
133.                     message.what = SHOW_ERROR;
134.                     message.obj = code + "";
135.                     handler.sendMessage(message);
136.                 }
137.
138.             } catch (Exception e) {
139.                 e.printStackTrace();
140.             } finally {
141.
142.             }
143.         }
144.     }).start();
145. }

```

```
146.
147.     private void parseJSONWithGSON(String result)
148.     {
149.
150.         Gson gson=new Gson();
151.         Word word=gson.fromJson(result, new TypeToken<Word>(){}.getType());
152.
153.         Message mes = new Message();
154.         mes.what = SHOW_RESPONSE;
155.         mes.obj = word.toString();
156.         handler.sendMessage(mes);
157.
158.     }
159.
160. }
161.
162. }
```

别忘了网络权限：

```
1. <uses-permission android:name="android.permission.INTERNET"/>
```

运行效果：

# good

SEARCH

Word [errorCode=0, query=good, translation=[好], basic=Basic [uk\_phonetic=gʊd, us\_phonetic=gʊd, phonetic=gʊd, explains=[n. 好处; 善行; 慷慨的行为, adj. 好的; 优良的; 愉快的; 虔诚的, adv. 好, n. (Good)人名; (英)古德; (瑞典)戈德]], web=[Web [key=Good, value=[好, 善, 商品]], Web [key=public good, value=[公共物品, 公益事业, 公共财]], Web [key=Good Doctor, value=[굿 닥터, Good Doctor (TV series), 好医生]]]



