

## 解析网络上的json数据

示例为实现查单词功能，解析来自有道词典的json数据

注意：

程序必须先申请可用的key

申请地址：<http://fanyi.youdao.com/openapi?path=data-mode>

数据接口：

<http://fanyi.youdao.com/openapi.do?keyfrom=<keyfrom>&key=<key>&type=data&doctype=<doctype>&version=1.1&q=要翻译的文本>

版本：1.1，请求方式：get，编码方式：utf-8

主要功能：中英互译，同时获得有道翻译结果和有道词典结果（可能没有）

参数说明：

type - 返回结果的类型，固定为data

doctype - 返回结果的数据格式，xml或json或jsonp

version - 版本，当前最新版本为1.1

q - 要翻译的文本，不能超过200个字符，需要使用utf-8编码

errorCode：

0 - 正常

20 - 要翻译的文本过长

30 - 无法进行有效的翻译

40 - 不支持的语言类型

50 - 无效的key

## xml数据格式举例

---

<http://fanyi.youdao.com/openapi.do?keyfrom=<keyfrom>&key=>

[<key>&type=data&doctype=xml&version=1.1&q=这里是有道翻译API](http://fanyi.youdao.com/openapi.do?keyfrom=<keyfrom>&key=<key>&type=data&doctype=xml&version=1.1&q=这里是有道翻译API)

`<?xml version="1.0" encoding="UTF-8"?>`

`<youdao-fanyi>`

`<errorCode>0</errorCode>`

`<!-- 有道翻译 -->`

`<query><![CDATA[这里是有道翻译API]]></query>`

`<translation>`

`<paragraph><![CDATA[Here is the youdao translation API]]></paragraph>`

`</translation>`

`</youdao-fanyi>`

## json数据格式举例

http://fanyi.youdao.com/openapi.do?keyfrom=<keyfrom>&key=

<key>&type=data&doctype=json&version=1.1&q=good

```
{
  "errorCode":0
  "query":"good",
  "translation":["好"], // 有道翻译
  "basic":{" // 有道词典-基本词典
    "phonetic":"gʊd"
    "uk-phonetic":"gʊd" //英式发音
    "us-phonetic":"gʊd" //美式发音
    "explains":[
      "好处",
      "好的"
      "好"
    ]
  },
  "web":[" // 有道词典-网络释义
    {
      "key":"good",
      "value":["良好","善","美好"]
    },
    {...}
  ]
}
```

## 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.         />
39.
40.     </ScrollView>
41.
42. </LinearLayout>

```

主程序：

```

1. package com.example.helloworld;
2.
3. import java.io.BufferedReader;
4. import java.io.InputStreamReader;
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.EntityUtils;
12. import org.json.JSONArray;
13. import org.json.JSONObject;
14.
15. import android.app.Activity;
16. import android.os.Bundle;
17. import android.os.Handler;
18. import android.os.Message;
19. import android.view.Menu;
20. import android.view.MenuItem;
21. import android.view.View;
22. import android.view.View.OnClickListener;
23. import android.widget.Button;
24. import android.widget.EditText;
25. import android.widget.TextView;
26. import android.widget.Toast;
27.

```

```

28. public class MainActivity extends Activity {
29.
30.     private static final int SHOW_RESPONSE = 0;
31.     private static final int SHOW_ERROR = 1;
32.     private EditText searchText;
33.     private Button btn_search;
34.     private TextView searchresult;
35.
36.     private String YouDaoBaseUrl = "http://fanyi.youdao.com/openapi.do";
37.     private String YouDaoKeyFrom = "y3u4q0329"; // 此处需到网上申请key
38.     private String YouDaokey = "825755169";
39.     private String YouDaoType = "data";
40.     private String YouDaoDoctype = "json";
41.     private String YouDaoVersion = "1.1";
42.
43.     private Handler handler = new Handler() {
44.         public void handleMessage(android.os.Message msg) {
45.             switch (msg.what) {
46.                 case SHOW_RESPONSE:
47.                     String response = (String) msg.obj;
48.                     // 在这里进行UI操作，将结果显示到界面上
49.                     searchresult.setText(response);
50.                     Toast.makeText(getApplicationContext(),
51.                         "here" + (String) msg.obj, Toast.LENGTH_SHORT).show();
52.                     break;
53.                 case SHOW_ERROR:
54.                     Toast.makeText(getApplicationContext(),
55.                         "here" + (String) msg.obj, Toast.LENGTH_SHORT).show();
56.                     break;
57.                 default:
58.                     break;
59.             }
60.         };
61.     };
62.
63.     @Override
64.     protected void onCreate(Bundle savedInstanceState) {
65.         super.onCreate(savedInstanceState);
66.         setContentView(R.layout.activity_main);
67.         ini();
68.     }
69.
70.     private void ini() {
71.         searchText = (EditText) findViewById(R.id.searchcontent);
72.         btn_search = (Button) findViewById(R.id.btn_search);
73.         searchresult = (TextView) findViewById(R.id.searchresult);
74.         btn_search.setOnClickListener(new searchListener());
75.
76.     }
77.
78.     class searchListener implements OnClickListener {
79.         @Override
80.         public void onClick(View v) {
81.             // TODO Auto-generated method stub

```

```

82.         String YouDaoSearchContent = searchText.getText().toString().trim();
83.         if (YouDaoSearchContent.equals("")) {
84.             Toast.makeText(MainActivity.this, "请输入要查询的内容!",
85.                 Toast.LENGTH_LONG).show();
86.         } else {
87.
88.             String YouDaoUrl = YouDaoBaseUrl + "?keyfrom=" + YouDaoKeyFrom
89.                 + "&key=" + YouDaokey + "&type=" + YouDaoType
90.                 + "&doctype=" + YouDaoDoctype + "&version="
91.                 + YouDaoVersion + "&q=" + YouDaoSearchContent;
92.
93.             sendRequestWithHttpClient(YouDaoUrl);
94.
95.         }
96.     }
97.
98.     private void sendRequestWithHttpClient(final String youDaoUrl) {
99.
100.        new Thread(new Runnable() {
101.
102.            @Override
103.            public void run() {
104.                // TODO Auto-generated method stub
105.
106.                try {
107.                    // httpget会自动处理重定向问题
108.                    HttpClient httpClient = new DefaultHttpClient();
109.                    HttpGet httpGet = new HttpGet(youDaoUrl);
110.                    HttpResponse httpResponse = httpClient.execute(httpGet);
111.                    // 200 成功 302 重定向(页面跳转) 404资源没找到 5xx 服务器内部错误
112.                    int code = httpResponse.getStatusLine().getStatusCode();
113.
114.                    if (code == 200) {
115.
116.                        BufferedReader input1 = new BufferedReader(
117.                            new InputStreamReader(httpResponse
118.                                .getEntity().getContent()));
119.                        StringBuilder sb = new StringBuilder();
120.
121.                        for (String s = input1.readLine(); s != null; s = input
122.                            .readLine()) {
123.                            sb.append(s);
124.                        }
125.                        AnalyzingOfJson(sb.toString());
126.                    } else {
127.
128.                        Message message = new Message();
129.                        message.what = SHOW_ERROR;
130.                        message.obj = code + "";
131.                        handler.sendMessage(message);
132.                    }
133.
134.                } catch (Exception e) {
135.                    e.printStackTrace();

```

```

136.         } finally {
137.         }
138.
139.     }
140.     }).start();
141. }
142.
143. private void AnalyzingOfJson(String result) throws Exception {
144.
145.     JSONArray jsonArray = new JSONArray "[" + result + "]";
146.     // String message=null;
147.     StringBuilder message = new StringBuilder();
148.     for (int i = 0; i < jsonArray.length(); i++) {
149.         JSONObject jsonObject = jsonArray.getJSONObject(i);
150.         if (jsonObject != null) {
151.             String errorCode = jsonObject.getString("errorCode");
152.
153.             if (errorCode.equals("0")) {
154.                 String query = jsonObject.getString("query");
155.                 // message=query;
156.                 message.append(query);
157.                 String translation = jsonObject
158.                     .getString("translation");
159.                 // message+="\t"+translation;
160.                 message.append("\t" + translation);
161.                 // 有道词典-基本词典
162.                 if (jsonObject.has("basic")) {
163.                     JSONObject basic = jsonObject
164.                         .getJSONObject("basic");
165.                     if (basic.has("phonetic")) {
166.                         String phonetic = basic.getString("phonetic");
167.                         // message+="\n\t"+phonetic;
168.                         message.append("\n\t音标: [" + phonetic + "]");
169.                     }
170.                     if (basic.has("explains")) {
171.                         String explains = basic.getString("explains");
172.                         // message+="\n\t"+explains;
173.                         message.append("\n\t" + explains);
174.                     }
175.                 }
176.                 if (jsonObject.has("web")) {
177.                     String web = jsonObject.getString("web");
178.                     JSONArray webstring = new JSONArray "[" + web + "]";
179.                     // message+="\n网络释义: ";
180.                     message.append("\n网络释义: ");
181.                     JSONArray webArray = webstring.getJSONArray(0);
182.                     int count = 0;
183.                     while (!webArray.isNull(count)) {
184.                         if (webArray.getJSONObject(count).has("key")) {
185.                             String key = webArray.getJSONObject(count)
186.                                 .getString("key");
187.                             // message+="\n\t<"+(count+1)+">"+key;
188.                             message.append("\n\t<" + (count + 1) + ">"
189.                                 + key);

```

```

190.
191.         }
192.         if (webArray.getJSONObject(count).has("value")) {
193.             String value = webArray
194.                 .getJSONObject(count).getString(
195.                     "value");
196.             // message+="\n\t "+value;
197.             message.append("\n\t " + value);
198.
199.         }
200.         count++;
201.     }
202.
203.     }
204.
205. }
206. if (errorCode.equals("20")) {
207.
208.     Message mes = new Message();
209.     mes.what = SHOW_ERROR;
210.     mes.obj = "要翻译的文本过长";
211.     handler.sendMessage(mes);
212. }
213. if (errorCode.equals("30")) {
214.     Message mes = new Message();
215.     mes.what = SHOW_ERROR;
216.     mes.obj = "无法进行有效的翻译 ";
217.     handler.sendMessage(mes);
218. }
219. if (errorCode.equals("40")) {
220.
221.     Message mes = new Message();
222.     mes.what = SHOW_ERROR;
223.     mes.obj = "不支持语言类型";
224.     handler.sendMessage(mes);
225. }
226. if (errorCode.equals("50")) {
227.     Message mes = new Message();
228.     mes.what = SHOW_ERROR;
229.     mes.obj = "无效的Key";
230.     handler.sendMessage(mes);
231. }
232. } else {
233.     Message mes = new Message();
234.     mes.what = SHOW_ERROR;
235.     mes.obj = "ERROR";
236.     handler.sendMessage(mes);
237. }
238. }
239. Message mes = new Message();
240. mes.what = SHOW_RESPONSE;
241. mes.obj = message.toString();
242. handler.sendMessage(mes);
243.

```

```
244.         }  
245.  
246.     }  
247.  
248. }
```

申请权限

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

运行：



# hello

---

SEARCH

hello ["你好"]

音标: [hə'ləʊ; he-]

["n. 表示问候, 惊奇或唤起注意时的用语","int. 喂; 哈罗",  
"n. (Hello)人名; (法)埃洛"]

网络释义:

<1>Hello

["你好","您好","Hello"]

<2>Hello Kitty

["凯蒂猫","昵称","匿称"]

<3>Hello Bebe

["哈乐哈乐","乐扣乐扣"]

