该程序参考自网络

首先贴上json文件

test.txt文件----放在res/raw文件夹下

```
{
    "error": 0,
    "status": "success",
    "date": "2014-05-10",
    "results": [
        {
            "currentCity": "南京",
            "weather_data": [
                {
                    "date": "周六(今天, 实时：19℃)",
                    "dayPictureUrl": "http://api.map.baidu.com/images/weather/day/c
                    "nightPictureUrl": "http://api.map.baidu.com/images/weather/nig
                    "weather": "大雨",
                    "wind": "东南风5-6级",
                    "temperature": "18℃"
                },
                {
                    "date": "周日",
                    "dayPictureUrl": "http://api.map.baidu.com/images/weather/day/z
                    "nightPictureUrl": "http://api.map.baidu.com/images/weather/nig
                    "weather": "阵雨转多云",
                    "wind": "西北风4-5级",
                    "temperature": "21 ~ 14℃"
                }
            ]
        }
    ]
}
```

读取该文件的程序片段

```
        public String readFileFromRaw(int resourceId) {
            if(resourceId < 0 ){
                return null;
            }

            String result = null;
            try {
                InputStream inputStream = getResources().openRawResource( resou
                // 获取文件的字节数
                int length = inputStream.available();
                // 创建byte数组
                byte[] buffer = new byte[length];
                // 将文件中的数据读到byte数组中
                inputStream.read(buffer);
                result = EncodingUtils.getString(buffer, "gbk");
```

```
                    //result=new String(buffer,0,length);
                } catch (Exception e) {
                    e.printStackTrace();
                }

                return result;
            }
```

进入正题，使用gson

首先gson将json解析为对象所需的类

```
import java.util.List;

public class Status
{
        private String error;
        private String status;
        private String date;
        private List<Results> results;
        public String getError()
        {
            return error;
        }
        public void setError(String error)
        {
            this.error = error;
        }

        public String getStatus()
        {
            return status;
        }
        public void setStatus(String status)
        {
            this.status = status;
        }
        public String getDate()
        {
            return date;
        }
        public void setDate(String date)
        {
            this.date = date;
        }
        public List<Results> getResults()
        {
            return results;
        }
        public void setResults(List<Results> results)
        {
            this.results = results;
        }
```

```
        @Override
        public String toString()
        {
            return "Status [error=" + error + ", status=" + status
                    + ", date=" + date + ", results=" + results + "]";
        }

}
```

```
import java.util.List;

public class Results
{
    private String currentCity;
    private List<Weather> weather_data;
    public String getCurrentCity()
    {
        return currentCity;
    }
    public void setCurrentCity(String currentCity)
    {
        this.currentCity = currentCity;
    }
    public List<Weather> getWeather_data()
    {
        return weather_data;
    }
    public void setWeather_data(List<Weather> weather_data)
    {
        this.weather_data = weather_data;
    }
    @Override
    public String toString()
    {
        return "Results [currentCity=" + currentCity + ", weather_data="
                + weather_data + "]";
    }
}
```

```
public class Weather {
    private String date;
                private String dayPictureUrl;
                private String nightPictureUrl;
                private String weather;
                private String wind;
                private String temperature;
                public String getDate() {
                    return date;
                }
```

```
    public void setDate(String date) {
        this.date = date;
    }
    public String getDayPictureUrl() {
        return dayPictureUrl;
    }
    public void setDayPictureUrl(String dayPictureUrl) {
        this.dayPictureUrl = dayPictureUrl;
    }
    public String getNightPictureUrl() {
        return nightPictureUrl;
    }
    public void setNightPictureUrl(String nightPictureUrl) {
        this.nightPictureUrl = nightPictureUrl;
    }
    public String getWeather() {
        return weather;
    }
    public void setWeather(String weather) {
        this.weather = weather;
    }
    public String getWind() {
        return wind;
    }
    public void setWind(String wind) {
        this.wind = wind;
    }
    public String getTemperature() {
        return temperature;
    }
    public void setTemperature(String temperature) {
        this.temperature = temperature;
    }
    @Override
    public String toString() {
        return "Weather [date=" + date + ", dayPictureUrl="
                + dayPictureUrl + ", nightPictureUrl="
                + nightPictureUrl + ", weather=" + weather
                + ", wind=" + wind + ", temperature=" + temperature
                + "]";
    }

}
```

主程序：

```
package com.example.helloworld;

import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.List;
```

```java

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.util.EncodingUtils;
import org.apache.http.util.EntityUtils;
import org.json.JSONArray;
import org.json.JSONObject;

import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;

import android.app.Activity;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends Activity {

    private static final int SHOW_RESPONSE = 0;
    private static final int SHOW_ERROR = 1;

    private Button btn_read;
    private TextView result;

    private Handler handler = new Handler() {
        public void handleMessage(android.os.Message msg) {
            switch (msg.what) {
            case SHOW_RESPONSE:
                String response = (String) msg.obj;
                // 在这里进行UI操作，将结果显示到界面上
                result.setText(response);

                break;
            case SHOW_ERROR:
                Toast.makeText(getApplicationContext(),
                        "here" + (String) msg.obj, Toast.LENGTH_SHORT).show();
                break;
            default:
                break;
            }
        };
    };

```

```
61.     @Override
62.     protected void onCreate(Bundle savedInstanceState) {
63.         super.onCreate(savedInstanceState);
64.         setContentView(R.layout.activity_main);
65.         ini();
66.     }
67. 
68.     private void ini() {
69.         btn_read = (Button) findViewById(R.id.btn_read);
70.         result = (TextView) findViewById(R.id.result);
71.         btn_read.setOnClickListener(new searchListener());
72. 
73.     }
74. 
75.     class searchListener implements OnClickListener {
76.         @Override
77.         public void onClick(View v) {
78.             // TODO Auto-generated method stub
79. 
80. 
81.         }
82. 
83.         private void parseJSONWithGSON()
84.         {
85.             //List<Word> wordList = new TypeToken<List<Word>>() {}.getType();
86.             //List<CallBackVo>> list = gson.from(jsonstr,listtype);
87. 
88.             String test=readFileFromRaw(R.raw.test);
89.             Gson gson=new Gson();
90.             Status status=gson.fromJson(test, Status.class);
91. 
92.             Message mes = new Message();
93.             mes.what = SHOW_RESPONSE;
94.             mes.obj = word.toString();
95.             handler.sendMessage(mes);
96. 
97.         }
98.          public String readFileFromRaw(int resourceId) {
99.                 if(resourceId < 0 ){
100.                     return null;
101.                 }
102. 
103.                 String result = null;
104.                 try {
105.                     InputStream inputStream = getResources().openRawResource( resou
106.                     // 获取文件的字节数
107.                     int length = inputStream.available();
108.                     // 创建byte数组
109.                     byte[] buffer = new byte[length];
110.                     // 将文件中的数据读到byte数组中
111.                     inputStream.read(buffer);
112.                     result = EncodingUtils.getString(buffer, "gbk");
113.                     //result=new String(buffer,0,length);
114.                 } catch (Exception e) {
```

```
                    e.printStackTrace();
                }

                return result;
            }

    }

}
```