



移动平台应用软件开发

解析JSON格式数据

主讲：张齐勋

zhangqx@ss.pku.edu.cn

《移动平台应用软件开发》课程建设小组

北京大学

二零一五年



北京大学

JSON格式数据

- 在网络上传输数据时最常用的格式有两种：
 - XML和JSON，均是通用数据交换格式
- JSON的主要优势是体积小。

```
<root>
  <success>1</success>
  <result>
    <weaid>1</weaid>
    <days>2015-03-18</days>
    <week>星期三</week>
    <cityno>beijing</cityno>
    <citynm>北京</citynm>
    <cityid>101010100</cityid>
    <temperature>15℃/5℃</temperature>
    <temperature_curr>15℃</temperature_curr>
    <humidity>0% / 0%</humidity>
    <weather>多云</weather>
    <weather_icon>http://api.k780.com:88/upload/weather/d/1.gif</weather_i
    <weather_icon1>http://api.k780.com:88/upload/weather/d/1.gif</weather_
    <wind>无持续风向</wind>
    <winp>微风</winp>
    <temp_high>15</temp_high>
    <temp_low>5</temp_low>
    <temp_curr>15</temp_curr>
    <humi_high>0</humi_high>
    <humi_low>0</humi_low>
    <weatid>2</weatid>
    <weatidl>2</weatidl>
    <windid>124</windid>
    <winpid>125</winpid>
  </result>
</root>
```

```
{
  success: "1",
  result: {
    weaid: "1",
    days: "2015-03-18",
    week: "星期三",
    cityno: "beijing",
    citynm: "北京",
    cityid: "101010100",
    temperature: "15℃/5℃",
    temperature_curr: "15℃",
    humidity: "0% / 0%",
    weather: "多云",
    weather_icon: "http://api.k780.com:88/upload/weather/d/1.gif",
    weather_icon1: "http://api.k780.com:88/upload/weather/d/1.gif",
    wind: "无持续风向",
    winp: "微风",
    temp_high: "15",
    temp_low: "5",
    temp_curr: "15",
    humi_high: "0",
    humi_low: "0",
    weatid: "2",
    weatidl: "2",
    windid: "124",
    winpid: "125"
  }
}
```



北京大學

Android 中 JSON 数据解析

- 在Android中解析JSON的方法有多种：
 - 官方提供的JSONObject
 - 谷歌开源的GSON
 - 第三方开源库Jackson、FastJSON
- 本讲仅介绍GSON的使用方法



<https://code.google.com/p/google-gson/>



google-gson

A Java library to convert JSON to Java objects and vice-versa



gson-2.3.1.jar



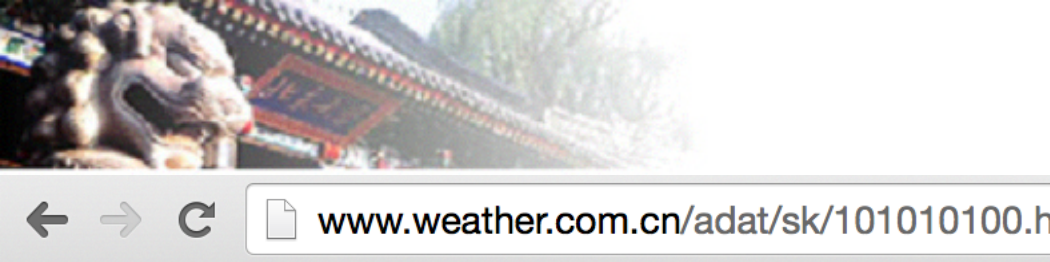
清华大学



GSON的使用

- Gson的解析规则是必须有一个bean文件
- 这个bean文件的内容与Json数据类型一一对应
- Gson有两个重要的方法，一个就是toJson，一个就是fromJson。
- toJson是将bean里面的内容转换为json内容
- fromjson是从Json对象封装出bean对象

- 把gson-*.*.jar放入libs文件夹
- 右击它,然后点击 'Add as library'
- 确保build.gradle文件中的依赖文件是正确的。
 - `compile files('libs/gson-2.3.1.jar')`



```
{  
  - weatherinfo: {  
    city: "北京",  
    cityid: "101010100",  
    temp: "10",  
    WD: "东南风",  
    WS: "2级",  
    SD: "26%",  
    WSE: "2",  
    time: "10:25",  
    isRadar: "1",  
    Radar: "JC_RADAR_AZ9010_JB",  
    njd: "暂无实况",  
    qy: "1012"  
  }  
}
```



```

public class TodayWeather {
    WeatherInfo weatherinfo;

    public void setWeatherinfo(WeatherInfo weatherinfo) {
        this.weatherinfo = weatherinfo;
    }

    public WeatherInfo getWeatherinfo() {
        return weatherinfo;
    }
}

```

```

public class WeatherInfo {
    private String city;
    private String cityid;
    private String temp;
    private String WD;
    private String WS;
    private String SD;
    private String WSE;
    private String time;
    private String isRadar;
    private String Radar;
    private String njd;
    private String qy;
}

```

```

private void parseJSON(String jsondata){
    mGson = new Gson();
    TodayWeather todayweather = mGson.fromJson(jsondata, TodayWeather.class);
    mCurWeatherinfo = todayweather.getWeatherinfo();
    Log.d("MyAPPGson", mCurWeatherinfo.getCity());
}
}

```

Log level: Debug



myap

logcat

```

03-18 15:27:50.097 3624-3642/com.example.zhangqx.myapplication D/MyAPPGson: 北京
03-18 15:28:14.463 3624-3650/com.example.zhangqx.myapplication D/MyAPPGson: 北京

```



Q&A

本讲结束！



北京大学