Evan
Only let oneself become strong enough, good enough, can afford the life that you want to.

android之解析json数据格式详解

标签： android 数据 json

2015-12-01 18:24 202人阅读 评论(0) 收藏 举报

分类： 移动开发（38）

1.JSON解析

(1).解析Object之一:

view sourceprint?

```
1 {"url":"http://www.cnblogs.com/qianxudetianxia"}
```

解析方法：

view sourceprint?

```
1 JSONObject demoJson = new JSONObject(jsonString);
2 String url = demoJson.getString("url");
```

(2).解析Object之二:

view sourceprint?

```
1 {"name":"android","name":"iphone"}
```

解析方法:

view sourceprint?

```
1 JSONObject demoJson = new JSONObject(jsonString);
2 String name = demoJson.getString("name");
3 String version = demoJson.getString("version");
4 System.out.println("name:"+name+",version:"+version);
```

(3).解析Array之一:

view sourceprint?

```
1 {"number":[1,2,3]}
```

解析方法：

view sourceprint?

```
1 JSONObject demoJson = new JSONObject(jsonString);
2 JSONArray numberList = demoJson.getJSONArray("number");
3 for(int i=0; i<numberList.length(); i++){
4         //因为数组中的类型为int，所以为getInt，其他getString，getLong同用
```

```
5         System.out.println(numberList.getInt(i));
6 }
```

(4).解析Array之二:

view sourceprint?

```
1 {"number":[[1],[2],[3]]}
```

解析方法:

view sourceprint?

```
1 //嵌套数组遍历
2 JSONObject demoJson = new JSONObject(jsonString);
3 JSONArray numberList = demoJson.getJSONArray("number");
4 for(int i=0; i<numberList.length(); i++){
5             //获取数组中的数组
6             System.out.println(numberList.getJSONArray(i).getInt(0));
7 }
```

(5).解析Object和Array:

view sourceprint?

```
1 {"mobile":[{"name":"android"},{"name":"iphone"}]}
```

解析方法:

view sourceprint?

```
1 JSONObject demoJson = new JSONObject(jsonString);
2 JSONArray numberList = demoJson.getJSONArray("mobile");
3 for(int i=0; i<numberList.length(); i++){
4             System.out.println(numberList.getJSONObject(i).getString("name"));
5 }
```

所以，我们发现get后面接着的是你想要的得到的结果的类型：getType，这个对理解很有帮助。

(6).使用optType:

上面的例子，使用getType在碰到查找不到节点的时候，会抛出异常。

如果使用optType，找不到节点，则返回null或者默认值。

view sourceprint?

```
1 //无url节点，抛出异常
2 String url = demoJson.getString("url");
3 //无url节点，返回空，如果为基本类型，则返回默认值
4 String url = demoJson.optString("url");
```

(7).UTF-8的BOM头导致解析JSON异常的问题

到json文件保存为utf-8的时候，在windows平台下，会产生bom头"EF BB EF"字节在文本的最前面（需要用十六进制工具打开才能看的到）。

有两种解决方法：

a.使用UltraEdit打开json文件， 另存为的时候，选择格式UTF-8,无BOM头，如果还不行，在用记事本打开，另存为UTF-8下，多i

b.使用代码处理，截取json主体内容：

view sourceprint?

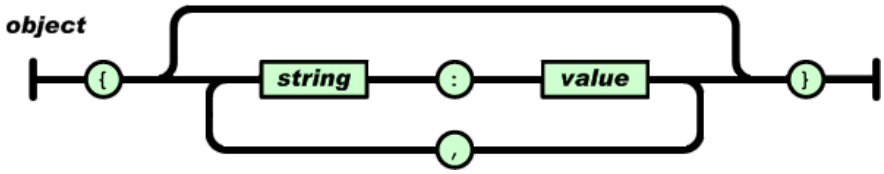
```
1 String jsonString = getJsonString();

2 jsonString = jsonString.substring(jsonString.indexOf("{"),jsonString.lastIndexOf("}")+1);
```

2.JSON必知

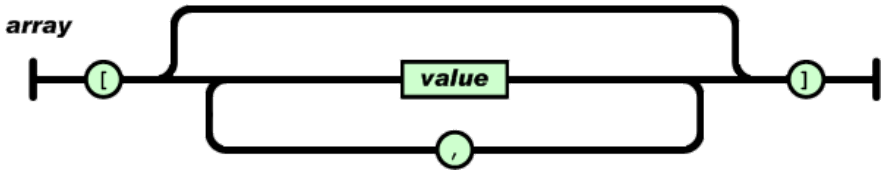
(1).JSON是一种轻量级的数据交换格式

(2).JSON基于两种**数据结构**：Object和Array。其中Object是“名称/值”对的集合。

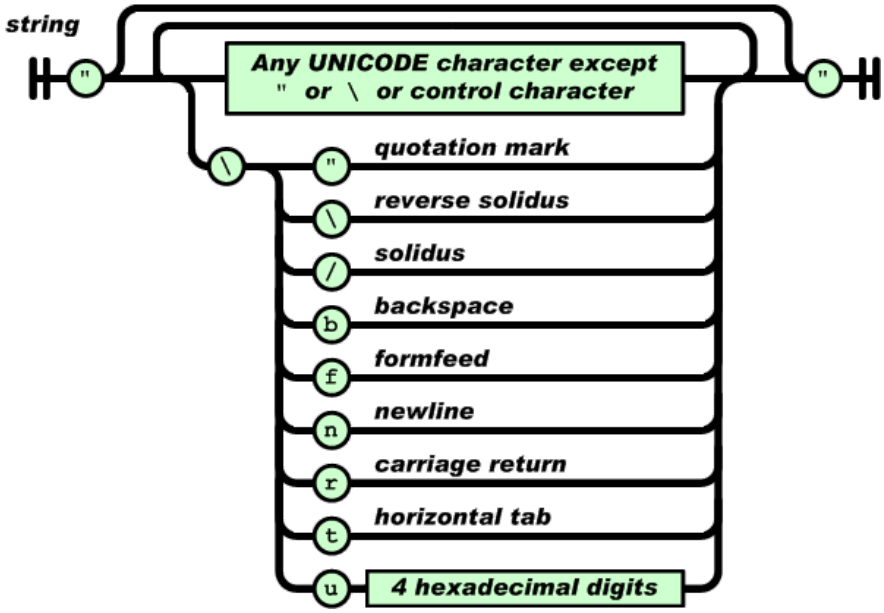
(3).对象：大括号，每一组string-value结合以","分隔，string和value以冒号分隔。



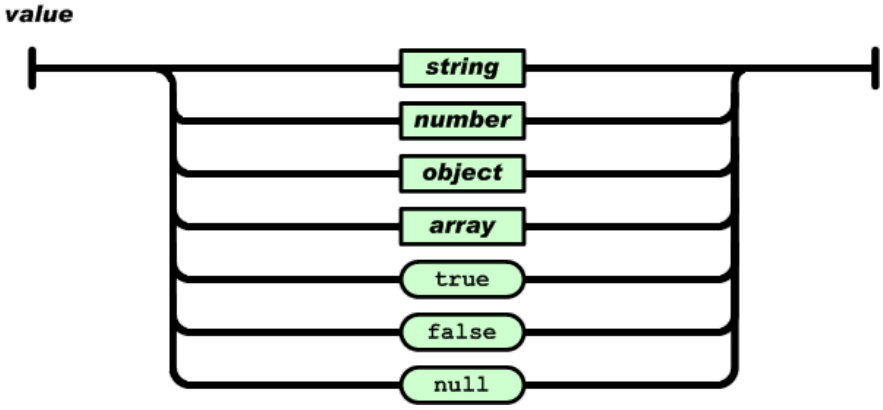
(4).数组：



(5).string由双引号包围的任意数量Unicode字符的集合，使用反斜线转义。



(6).value可以是双引号括起来的字符串（*string*）、数值(number)、true、false、 null、对象（object）或者数组（array）。这些结构可以嵌套。



(7).空白可以加入到任何符号之间,包括空格，tab，回车，换行等。

(8).举例：

a.Object实例:

view sourceprint?

```
{
        "Image": {
            "Width":  800,
            "Height": 600,
            "Title":  "View from 15th Floor",
            "Thumbnail": {
                "Url":       "http://www.example.com/image/481989943",
                "Height": 125,
                "Width":  "100"
            },
            "IDs": [116, 943, 234, 38793]
        }
}
```

b.Array实例:

view sourceprint?

```
[
    {
        "precision": "zip",
        "Latitude":  37.7668,
        "Longitude": -122.3959,
        "Address":     "",
        "City":          "SAN FRANCISCO",
        "State":         "CA",
        "Zip":           "94107",
        "Country":     "US"
```

```
11      },
12      {
13          "precision": "zip",
14          "Latitude":  37.371991,
15          "Longitude": -122.026020,
16          "Address":    "",
17          "City":          "SUNNYVALE",
18          "State":        "CA",
19          "Zip":            "94085",
20          "Country":    "US"
21      }
22 ]
```

3.小结

很简单 ，很基础，积水方能成江，累砖才可筑楼。

接下来举两个实际的例子：

一种是普通的，一种是带有数组形式的；

普通形式的：

服务器端返回的json数据格式如下：

```
{"userbean":{"Uid":"100196","Showname":"\疯\狂\的\猴\子","Avtar":null,"State":1}}
```

分析代码如下：

```
// TODO 状态处理 500 200
            int res = 0;
            res = httpClient.execute(httpPost).getStatusLine().getStatusCode();
            if (res == 200) {
               /*
                * 当返回码为200时，做处理
                * 得到服务器端返回json数据，并做处理
                * */
               HttpResponse httpResponse = httpClient.execute(httpPost);
               StringBuilder builder = new StringBuilder();
               B?redReader b?redReader2 = new B?redReader(
                     new InputStreamReader(httpResponse.getEntity().getContent()));
               String str2 = "";
               for (String s = b?redReader2.readLine(); s != null; s = b?redReader2
                     .readLine()) {
```

```
                builder.append(s);
            }
            Log.i("cat", ">>>>>>" + builder.toString());

JSONObject jsonObject = new JSONObject(builder.toString())
                .getJSONObject("userbean");

        String Uid;
        String Showname;
        String Avtar;
        String State;

        Uid = jsonObject.getString("Uid");
        Showname = jsonObject.getString("Showname");
        Avtar = jsonObject.getString("Avtar");
        State = jsonObject.getString("State");
```

带数组形式的：

服务器端返回的数据格式为：

```
{"calendar":
   {"calendarlist":
        [
        {"calendar_id":"1705","title":"(\亲\子)ddssd","category_name":"\默\认\分
\类","showtime":"1288927800","endshowtime":"1288931400","allDay":false},
        {"calendar_id":"1706","title":"(\旅\行)","category_name":"\默\认\分
\类","showtime":"1288933200","endshowtime":"1288936800","allDay":false}
        ]
   }
}
```

分析代码如下：

```
// TODO 状态处理 500 200
        int res = 0;
        res = httpClient.execute(httpPost).getStatusLine().getStatusCode();
        if (res == 200) {
            /*
             * 当返回码为200时，做处理
             * 得到服务器端返回json数据，并做处理
             * */
```

```
HttpResponse httpResponse = httpClient.execute(httpPost);
StringBuilder builder = new StringBuilder();
B?redReader b?redReader2 = new B?redReader(
      new InputStreamReader(httpResponse.getEntity().getContent()));
String str2 = "";
for (String s = b?redReader2.readLine(); s != null; s = b?redReader2
      .readLine()) {
   builder.append(s);
}
Log.i("cat", ">>>>>>" + builder.toString());
/**
 * 这里需要分析服务器回传的json格式数据，
 */
JSONObject jsonObject = new JSONObject(builder.toString())
      .getJSONObject("calendar");
JSONArray jsonArray = jsonObject.getJSONArray("calendarlist");
for(int i=0;i<jsonArray.length();i++){
   JSONObject jsonObject2 = (JSONObject)jsonArray.opt(i);
   CalendarInfo calendarInfo = new CalendarInfo();
   calendarInfo.setCalendar_id(jsonObject2.getString("calendar_id"));
   calendarInfo.setTitle(jsonObject2.getString("title"));
   calendarInfo.setCategory_name(jsonObject2.getString("category_name"));
   calendarInfo.setShowtime(jsonObject2.getString("showtime"));
   calendarInfo.setEndtime(jsonObject2.getString("endshowtime"));
   calendarInfo.setAllDay(jsonObject2.getBoolean("allDay"));
   calendarInfos.add(calendarInfo);
}
```

总结，普通形式的只需用JSONObject ，带数组形式的需要使用JSONArray 将其变成一个list。

转：http://blog.163.com/tuchengju@126/blog/static/3807116520116225462596 1/

- 上一篇 struts2: config-browser-plugin 与 convention-plugin 学习
- 下一篇 南阳ACM 题目8：一种排序 Java版