# 数据解析技术选型

1. Json

Json(JavaScript Object Notation)是一种轻量级的数据格式, 目前大多数的语言平台都支持Json的解析, 而且Json采用了完全独立于语言的文本格式，因此使用Json数据解析可以在不同的平台进行数据交换。

1. 具有良好的可读性；
2. 易于机器快速解析和生成;
3. 便于开发人员快速编写;
4. 拥有丰富的解析手段
5. 相比于其它数据解析方式(比如XML解析)，数据的体积更小

Json数据包括两种组织结构形式：

1. 以键值对(key/value)形式存在的无序的jsonObject对象，一个对象以“{”（左花括号）开始，“}”（右花括号）结束。每个“名称”后跟一个“:”（冒号）；“‘名称/值’ 对”之间使用“,”（逗号）分隔，对于该种格式，其中key的值必须是String类型，String类型的值是使用" "号包含的，它可以包括任意长度的字符的集合，使用反斜线转义。一个字符（character）即一个单独的字符串（character string)，而对于value，则可以是String、Object、number和Array等等。
2. Json另一种数据组织就是有序的value的集合，这种形式被称为是jsonArray，数组是值（value）的有序集合。一个数组以“[”（左中括号）开始，“]”（右中括号）结束。值之间使用“,”（逗号）分隔同样的，value也可以是String、Object、number和Array等等。例如["AndroidLeaf","Steve.P.Jobs","Bill Gates"]就是一个数组集合，其组织形式是[String]。

目前Android在使用的Json解析框架主要有Jackson、Gson、FastJson、org.json，支持Json解析的第三方Json解析库还有很多，较为常用的有jsonp、json-lib和Json-smart等，感兴趣的同学可以自行研究学习。

### 1.1 Android原生Api

JSONObject：JSONObject是一个名称/值对的集合。它主要就是为了操作我们上面介绍的Json两种数据组织结构形式中的第一种，以键值对(key/value)形式存在的无序的jsonObject对象。该类的作用主要包括：可以用JSONObject.put(String key,Object object)方法将Java对象转换成Json字符串；还有就是可以用JSONObject.get(String key)方法将Json字符串转换成对应的Java对象。实际上一句话总结就是JSONObject可以实现Java对象和Json字符串之间的相互转换。

JSONArray: JSONArray是一个有序序列值的集合。它主要就是为了操作我们上介绍的第二种数据组织结构形式。该类的主要作用也是可以用JSONArray.put(Object object)方法和JSONArray.get...(Object object)方法实现Java对象集合和Json字符串之间的相互转换。

JSONStringer:JSONStringer是一个JSON文本构建类 ，根据官方的解释，这个类可以帮助快速和便捷的创建JSON text。其最大的优点在于可以减少由于 格式的错误导致程序异常，引用这个类可以自动严格按照JSON语法规则（syntax rules）创建JSON text。每个JSONStringer实体只能对应创建一个JSON text。

JSONTokener:JSONTokener是一个Json字符串的解析类，它主要的作用就是读取一个Json字符串里面的字符，比如读取Json字符串中的指定位置的字符、读取下一个字符、读取上一个字符、跳转到指定位置的字符等等。

JSONException：JSON异常处理类

### 1.2 Gson数据解析

Gson解析技术使用简单、方便的特点，目前被很多公司和个人应用。在使用Gson解析时，使用最频繁的是toJson()和fromJson()两个装换方法。其中toJson()方法主要作用是将Java对象转换成Json字符串，而fromJson()方法主要作用是将Json字符串转换成对应的Java对象或集合。

Github:<https://github.com/google/gson>

**优势**

* 快速，高效
* 代码量少
* 面向对象
* 数据传输解析方便
* 可按需解析

**注意事项**

1、内部嵌套的类必须是static的，要不然解析会出错；   
2、类里面的属性名必须跟Json字段里面的Key是一模一样的；   
3、内部嵌套的用[]括起来的部分是一个List，所以定义为 public List< B> b，而只用{}嵌套的就定义为 public C c，是不是很简单，而且现在Android Studio上可以直接用插件生成实体类，那更加方便了。

### 1.3 Jackson数据解析

Jackson可以轻松的将Java对象转换成JSON对象和xml文档，同样也可以将JSON、xml转换成Java对象。基于事件驱动，与GSON相同，先创建一个对应于JSON数据的JavaBean类就可以通过简单的操作解析出所需JSON数据。但和GSON解析不同的是，GSON可按需解析，即创建的JavaBean类不一定完全涵盖所要解析的JSON数据，按需创建属性，但Jackson解析对应的JavaBean必须把JSON数据里面的所有key都有所对应，即必须把JSON内的数据所有解析出来，无法按需解析。但Jackson的解析速度和效率都要比GSON高。

Github:<https://github.com/FasterXML/jackson>

**优势**

* 解析效率最高
* 在数据量大的情况优势尤为明显、占存少

**缺点**

* 必须完全解析文档，如果要按需解析的话可以拆分JSON来读取，操作和解析方法复杂；

**推荐使用情景**

* 适用于需要处理超大型JSON文档、不需要对JSON文档进行按需解析、、性能要求较高的场合。

### 1.4 FastJson数据解析

FastJson是阿里巴巴公司提供的一个用Java语言编写的高性能功能完善的JSON库，该库涉及的最基本功能就是序列化和反序列化。Fastjson支持Java bean的直接序列化,同时也支持集合、Map、日期、Enum和泛型等的序列化。你可以使用com.alibaba.fastjson.JSON这个类进行序列化和反序列化，常用的序列化操作都可以在JSON类上的静态方法直接完成。Fastjson采用独创的算法，将parse的速度提升到极致，号称超过所有Json库。而且，使用Fastjson解析时，除了需要使用Fastjson所提供的jar包外，再不需要额外的jar包，就能够直接跑在JDK上。

Github:<https://github.com/alibaba/fastjson>

**优势**

* 快速FAST（比任何一款都快）
* 面向对象
* 功能强大(支持普通JDK类任意java bean Class,Collection,Map,Date或者 enum)
* 零依赖（只需要有JDK即可）
* 支持注解，全类型序列化

### 1.5 JSON框架分析

以上介绍了Jackson、Gson、FastJson的基本内容，细心的同学可能发现介绍Gson与FastJson时我并未提及他们的缺点，这里说一下，并不是说这两个框架没有缺点，只是相对与其他的框架他们的缺点并不突出。特别介绍了Jackson的缺点，以及推荐了它使用的场景，那么其他场景下还是不建议大家使用，这点对于开发者来说其实也是需要考虑的一个问题，如果该库的API使用难度大，或者很复杂，那么不建议使用，毕竟JSON解析的性能差异并不大。

|  |  |  |  |
| --- | --- | --- | --- |
| 三者性能对比 | | | |
|  | 序列化性能(ns) | 反序列化性能(ns) | jar大小(kb) |
| org.json | 5468 | 6904 | / |
| Gson | 4667 | 4403 | 227 |
| Jackson | 1164 | 1866 | jackson-annotations-2.8.6.jar(55)  jackson-core-2.8.1.jar(274)  jackson-databind-2.8.5.jar(1207) |
| FastJson | 1058 | 1241 | 205 |
| 数据来源：<https://github.com/eishay/jvm-serializers/wiki> | | | |

注: 该结论的对错与否暂先不讨论, 阿里Github网也引用了这个数据。

**选型如下：**

1. Jackson在性能上确实要优于Gson，但是依赖太多了，不易用，jar太大。
2. 比较来说, Gson比fastjson 考虑更全面, 对用 URL , UUID, BIT\_SET, CALENDAR 等等,都有特定的输出规则。
3. Gson 有对各种类型的属性支持, 如果有特殊类型json化需求或复杂结构时可以选择 Gson,并自定义扩充。
4. 小数量的调用 Gson 比 fastjson 快一点. (几十毫秒,可以毫不在意.猜测是因为 javassist 生成新的 Wrapper 类导致,因为还要编译的.)。
5. 大数量的调用 fastjson 比 Gson 快. (千万级别的.还不太确定为什么会变快, 猜测是 gson 的反射调用,毕竟比不上 fastjson Wrapper 类的真实调用.)
6. 代码可阅读性: fastjson 比 Gson 好很多很多
7. Fastjson是阿里开发的
8. Xml

//待续。