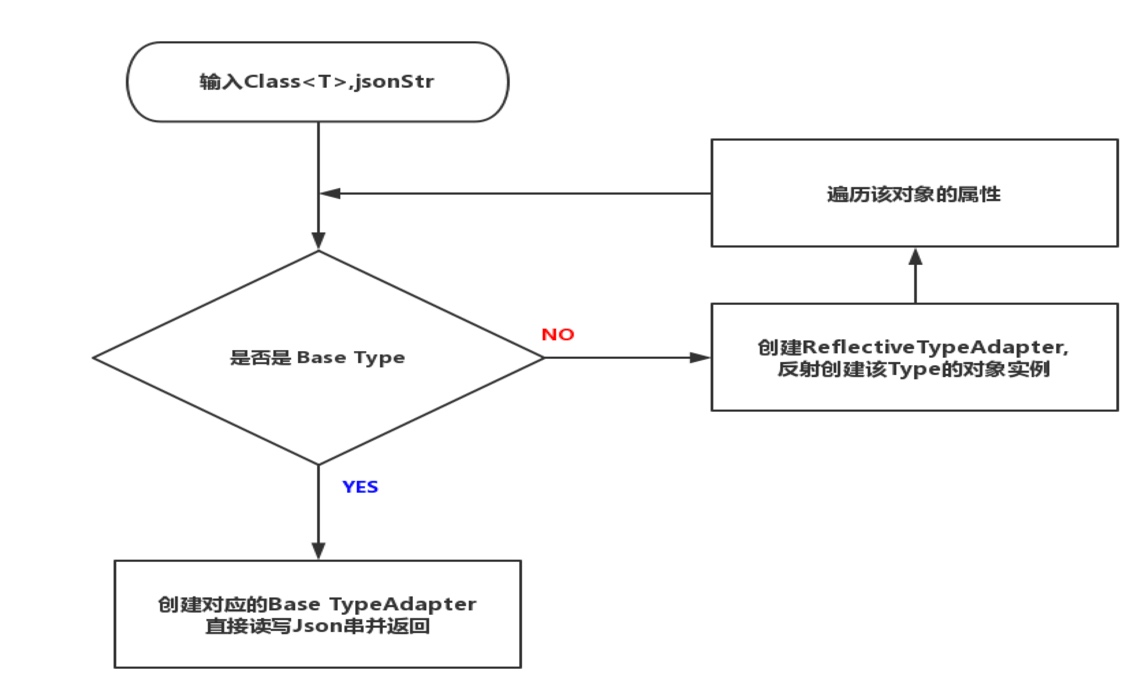
**android JSON解析**

参考：<https://www.jianshu.com/p/b87fee2f7a23>

1. JSON：JavaScript Object Notation，轻量级的文本数据交换格式，独立于语言和平台，读写速度快，解析简单。
2. JSON值：键值对、数组、对象。Android解析JSON数据的方法主要有两种：基于事件驱动（Gson和Jackson）和基于文档驱动解析方式。
3. Gson解析：基于事件驱动，关键是根据json数据里面的结构写出一个对应的javaBean。
4. JSON的大括号对应一个对象，对象里面有key和value(值)。在JavaBean里面的类属性要和key同名；
5. JSON的方括号对应一个数组，所以在JavaBeanBean里面对应的也是数组，数据里面可以有值或者对象。如果数组里面只有值没有key，就说明它只是一个纯数组，如果里面有值有key，则说明是对象数组。纯数组对应JavaBean里面的数组类型，对象数组要在Bean里面建立一个内部类，内部类的类属性就是对应的对象里面的key，建立了之后要创建一个这个内部类的对象，名字对应数组名
6. 对象里面嵌套对象时候，也要建立一个内部类，和对象数组一样，这个内部类对象的名字就是父对象的key。
7. Gson解析流程：

参考：<https://www.jianshu.com/p/aef252db9869>



1. Gson类型分为基本类型和复合类型（非基本类型）。每种基本类型都与唯一一个TypeAdapter相对应(适配器模式)，复合类型由ReflectiveTypeAdapter统一处理。通过getAdapter(type)可以找到与type对应的TypeAdapter。
2. getAdapter(type) 首先尝试从缓存typeTokenCache中获取是否有对应的TypeAdapter，然后再从ThreadLocal中map缓存找是否FutureTypeAdapter，有的话直接返回FutureTypeAdapter（代理模式，防止类中属性为本类时死循环），否则new FutureTypeAdapter并将属性对应的TypeAdapter保存进去，最后将FutureTypeAdapter缓存到ThreadLocal的map中。
3. ReflectiveTypeAdapter需要创建该类型javabean，在解析时首先创建该类型的对象，然后遍历该对象内部的所有属性保存到map中,接着把json传的读去委托给了各个属性。若为基本类型则直接读取，否则递归ReflectiveTypeAdapter创建流程。
4. ReflectiveTypeAdapter解析时，JsonRead每读取到一个name就从map找到对应的属性并赋值。
5. Jackson解析：基于事件驱动。与Gson解析一样，需要创建一个对应的javaBean。
6. Gson解析与Jackson解析的不同：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内的所有数据解析出来，无法按需解析。但Jackson的解析速度和效率都要比GSON高。
7. Android API自带org.json解析，基于文档驱动。类似于XML的DOM解析方法，先把全部文件读入到内存中，然后遍历所有数据，然后根据需要检索想要的数据。