Gson 解析数据使用反射typeadapter解析。

bean对象中可以使用@SerialzedName("xxx")修饰字段，保证key为某一字段。

@Expose 修饰，标记是否需要序列化

@Since 修饰，版本控制。当前版本>=1.0时才会参与序列化/反序列化，否则忽略

@Until 修饰，版本控制。当前版本<=1.1时参加序列化/反序列化

@JsonAdapter 修饰，自定义adapter ，默认会通过反射typeadater解析。重写 read write 方法。

设置版本号

Gson gson = new GsonBuilder().setVersion("版本号").create();

TypeAdapter

JsonWriter 将javabean写字符串

JsonReader 将字符串读成javabean

Gson解析成JsonElement可以判断类型。

JsonDeserializer 可以自己实现 JsonDeserializer 接口，通过方法deserialize(JsonElement json, Type typeOfT, JsonDeserializationContext context)

获取到 JsonElement 字段判断字段类型。

面试题

如何解决Caused by ： java.lang.IllegalStateExcetion: Expected BEGIN\_ARRAY but was STRING at line 3 column 17 path ...？

1.自定义TypeAdapter

2.自定义JsonDeserializer

原理分析

编译原理

分割 语法 词义分析，语法分析。

JsonToken 类型 BEGIN\_ARRAY END\_ARRAY BEGIN\_OBJECT END\_OBJECT NAME STRING NUMBER BOOLEAN NULL END\_DOCUMENT

xml -> DOM 解析

一次性加载内存。 效率高

===================string[]

"{" " "key":value " "," " "arrays":[] " "}"

简单数据 类型

xml -> SAX , PULL 解析

事件驱动 需要的时候再加载到内存。效率低

===================基于事件驱动

"{" -> object

"key":value

"arrays":[]

存放到object中

"}"

Json此法分析

- BEGIN\_OBJECT（{）

- END\_OBJECT（}）

- BEGIN\_ARRAY（[）

- END\_ARRAY（]）

- NULL（null）

- NUMBER（数字）

- STRING（字符串）

- BOOLEAN（true/false）

- SEP\_COLON（:）

- SEP\_COMMA（,）