#### **解决方案**

状态异常。

java.lang.IllegalStateException异常产生的原因及解决办法

错误类型大致为以下几种：

java.lang.IllegalStateException：Cannot forward a response that is already committed

IllegalStateException：response already commited

IllegalStateException：getOutputStream() has already been called for this request

…………

=========================================================

IllegalStateException: Can not perform this action after onSaveInstanceState：

#解决办法：onSaveInstanceState方法是在该Activity即将被销毁前调用，来保存Activity数据的，如果在保存玩状态后

再给它添加Fragment就会出错。解决办法就是把commit（）方法替换成 commitAllowingStateLoss()

=========================================================

错误原因：

该异常表示，当前对客户端的响应已经结束，不能在响应已经结束（或说消亡）后再向客户端（实际上是缓冲区）输出任何内容。

Object is no longer valid to operate on. Was it deleted by another thread?

该异常表示，realmObject对象在其他线程已被删除，在这个线程中使用的时候抛出的异常。

具体分析：

首先解释下flush()，我们知道在使用读写流的时候数据先被读入内存这个缓冲区中， 然后再写入文件，但是当数据读完时不代表数据已经写入文件完毕，因为可能还有一部分仍未写入文件而留在内存中，这时调用flush()方法就会把缓冲区的数据强行清空输出，因此flush()的作用就是保证缓存清空输出。response是服务端对客户端请求的一个响应，其中封装了响应头、状态码、内容等，服务端在把response提交到客户端之前，会向缓冲区内写入响应头和状态码，然后将所有内容flush。这就标志着该次响应已经committed(提交)。对于当前页面中已经committed(提交)的response，就不能再使用这个response向缓冲区写任何东西（注：同一个页面中的response.XXX()是同一个response的不同方法，只要其中一个已经导致了committed，那么其它类似方式的调用都会导致 IllegalStateException异常）。

参考：http://my.oschina.net/guhai2004/blog/187041，https://github.com/realm/realm-java/issues/1206

补充另一种异常情况：

我这里的异常是：

java.lang.IllegalStateException

Can't change tag of fragment d{e183845 #0 d{e183845}}: was d{e183845} now d{e183845 #0 d{e183845}}

经查，我在显示fragment的代码中使用了：

fragment.show(getSupportFragmentManager, fragment.toString());

而这里是因为两次toString()结果不同，导致不同的tag指向的是同一个fragment。

获取fragment的tag的正确方法应该是使用其提供的fragment.getTag()方法。

=====================================================

补充异常:

java.lang.IllegalStateException

Expected BEGIN\_ARRAY but was BEGIN\_OBJECT at line 1 column 37 path $.data

错误原因：该异常是由于服务器错误返回的JSON字符串和服务器正常下时返回的JSON字符串结构不同,导致利用Gson解析的时候报了一个异常:本该去解析集合却强制去解析对象所致.

解决办法:在使用Gson解析JSON时try cash一下,不报错按照正常逻辑继续解析,报异常则处理为请求失败逻辑即可.

======================================================