## 303.java.lang.IllegalStateException

【出错堆栈】

http://bugly.qq.com/share/vgIvce

【出错代码】

Message msg = new Message();  
 msg.what = 1;  
// msg.recycle();  
 Handler handler = new Handler();  
 handler.sendMessage(msg);  
 handler.sendMessage(msg);

【出错原因】

先从源码中找到出现这个异常的地方， 发现源头在MessageQueue类的enqueueMessage方法中

boolean enqueueMessage(Message msg, long when) {  
 if (msg.target == null) {  
 throw new IllegalArgumentException("Message must have a target.");  
 }  
 if (msg.isInUse()) {  
 throw new IllegalStateException(msg + " This message is already in use.");  
 }  
  
 synchronized (this) {  
 if (mQuitting) {  
 IllegalStateException e = new IllegalStateException(  
 msg.target + " sending message to a Handler on a dead thread");  
 Log.w("MessageQueue", e.getMessage(), e);  
 msg.recycle();  
 return false;  
 }

有两个地方出现了IllegalStateException,在msg.isInUse为true的时候抛出，可以看到是由于Message在使用中造成的，进入该方法

boolean isInUse() {  
 return ((flags & FLAG\_IN\_USE) == FLAG\_IN\_USE);  
}

可知当Handler.sendMessage(msg)中的msg对象如果其flag==FLAG\_IN\_USE就会出错。由这个原理我们代码中尝试可知，同一个Message对象多次被sendMessage调用，或者sendMessage之前执行了msg.recycle方法都会抛出该异常。

从这里看到了抛出异常的地方，与崩溃堆栈相吻合。

【解决方案】

由出错原因可知Message对象不可以重复多次被Handler.sendMessage(msg)调用，以及在被调用前不可以被recycle掉，一个是确保代码中不要调用msg.recycle，还有确保每次sendMessage之前Message对象都不是inuse状态的，即sendMessage的Message对象都是最新的（用new Message和obtain都能确保Message是最新的）。如果都无法保证则使用try{}catch(){}捕获可能的异常。