## 334.java.lang.RuntimeException

【出错堆栈】

http://bugly.qq.com/share/5ZpGZx

【出错代码】

public class MainActivity extends Activity {  
 ImageView mImg;  
 Bitmap drawable;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 mImg = (ImageView) findViewById(R.id.img);  
 drawable = BitmapFactory.decodeResource(getResources(), R.drawable.see);  
 ListView listView = (ListView) findViewById(R.id.list);  
 listView.setAdapter(new MAdpater());  
 }  
  
 public void onClick(View v) {  
 Log.i("a", "bug begin");  
 Toast.makeText(MainActivity.this, "向上滑动产生bug", Toast.LENGTH\_SHORT).show();  
 drawable.recycle();  
// drawable.setPremultiplied(false);  
 }  
  
 class MAdpater extends BaseAdapter {  
 @Override  
 public int getCount() {  
 return 20;  
 }  
 @Override  
 public Object getItem(int position) {  
 return position;  
 }  
 @Override  
 public long getItemId(int position) {  
 return position;  
 }  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent) {  
 View v = View.inflate(MainActivity.this, R.layout.item, null);  
 ImageView img = (ImageView) v.findViewById(R.id.img);  
 TextView textView = (TextView) v.findViewById(R.id.text);  
 textView.setText("position:" + position);  
 img.setDrawingCacheEnabled(false);  
 img.setImageBitmap(drawable);  
 return v;  
 }  
 }  
}

【出错原因】

先从源码中找到出现这个异常的地方，从imgview的ondraw方法找到

mDrawable.draw(canvas);

然后在BitmapDrawable中找到draw方法，发现调用了

canvas.drawBitmap(bitmap, null, mDstRect, state.mPaint);

具体代码如下：

public void drawBitmap(Bitmap bitmap, Rect src, Rect dst, Paint paint) {  
 if (dst == null) {  
 throw new NullPointerException();  
 }  
 throwIfCannotDraw(bitmap);  
 native\_drawBitmap(mNativeCanvas, bitmap.ni(), src, dst,  
 paint != null ? paint.mNativePaint : 0, mScreenDensity, bitmap.mDensity);  
}

发现源头在Canvas类中

protected static void throwIfCannotDraw(Bitmap bitmap) {  
 if (bitmap.isRecycled()) {  
 throw new RuntimeException("Canvas: trying to use a recycled bitmap " + bitmap);  
 }  
 if (!bitmap.isPremultiplied() && bitmap.getConfig() == Bitmap.Config.ARGB\_8888 &&  
 bitmap.hasAlpha()) {  
 throw new RuntimeException("Canvas: trying to use a non-premultiplied bitmap "  
 + bitmap);  
 }  
}

这个方法里面出现。可以看到是由于bitmap被recycle了以及设置了premultiplied之后

会抛出异常，也就是说 如果对ImageView展示的drawable执行了上面的两种操作会导致抛出此异常，与崩溃堆栈相吻合。

【解决方案】

由出错原因可知在ImageView绘制时，不要对其展示的图像 mDrawable（bitmap类型）执行这两种recycle（）和setPremultiplied（）方法。

有两种解决方案，

1. 在ImageView设置的时候判断下,且在imageView可能重绘，即执行ondraw的时候都要进行这个判断

if (!drawable.isRecycled()&&!drawable.isPremultiplied()){  
 img.setImageBitmap(drawable);  
}

或者在invalidate之前也要这样做

if (!drawable.isRecycled()&&!drawable.isPremultiplied()){  
 img.invalidate;  
}

2.重写ImageView的ondraw方法

@Override   
protected void onDraw(Canvas canvas) {

if (mDrawable !=null&&(mDrawable.isRecycled()||mDrawable.isPremultiplied())){  
 return;  
}

super.onDraw(canvas);

}

当发生这种情况的时候直接return。

当然，最好是控制好ImageView的mDrawable不要出现这两种情况。