

## 1、关于java.lang.UnsupportedOperationException异常

是指请求的方法不被支持的异常

从`Arrays.asList()`转化过来的List的不支持`add()`和`remove()`方法，这是由于从`Arrays.asList()`返回的是返回`java.util.Arrays$ArrayList`，而不是`ArrayList`。`Arrays$ArrayList`和`ArrayList`都是继承`AbstractList`，`add()`和`remove()`等方法在`AbstractList`中默认`throw UnsupportedOperationException`而不做任何操作。`ArrayList`重写这些方法对List进行操作，而`Arrays$ArrayList`却没有重写`add()`和`remove()`等方法，所以对从`Arrays.asList()`转化过来的List进行`add()`和`remove()`会出现`UnsupportedOperationException`异常。没有对集合的增删改查的方法，如果调用就会抛`UnsupportedOperationException`异常

异常原因：

```
private String[] F = {"一", "二", "三"};

protected List<String> setFirstData() {
    List<String> B = Arrays.asList(F);    集合工具类将数组转换为集合对象的
    return B;
}
```

其引用对象不可以直接赋值为另外的集合

错误做法：`ArrayList<T> datas = new ArrayList<>()``datas = setFirstData();`  
`datas.addAll(setFirstData());`;

正确做法：

只能是：`ArrayList<T> list = new ArrayList<>(Arrays.asList(F));`;

或者是`ArrayList<T> list = new ArrayList<>(setFirstData())`

`datas = list;`

## 2、对ImageView等图片的资源的操作 加载欢迎界面的图片时 bug: OutOfMemoryError. oom的错误

android加载大量图片内存溢出bitmap size exceeds VM budget的解决办法。

```
ImageView imageView = (ImageView) findViewById(R.id.img);
Bitmap bitmap = BitmapFactory.decodeResource(getResources(), R.drawable.asasa);
imageView.setImageBitmap(bitmap);

//手动回收 界面销毁时

bitmap.recycle();
```

尽量不要使用`setImageBitmap`或`setImageResource`或`BitmapFactory.decodeResource`来设置一张大图，

因为这些函数在完成`decode`后，最终都是通过java层的`createBitmap`来完成的，需要消耗更多内存。

然而，可以改用先通过`BitmapFactory.decodeStream`方法，创建出一个`bitmap`，再将其设为`ImageView`的 `source`。

`decodeStream`最大的不同之处在于其直接调用JNI>>`nativeDecodeAsset()`来完成`decode`，无需再使用java层的`createBitmap`，从而节省了java层的空间。

但是，`decodeStream`有这么一个缺点：

`decodeStream`直接拿的图片来读取字节码了，不会根据机器的各种分辨率来自动适应，使用了`decodeStream`之后，需要在`hdpi`和`mdpi`、`ldpi`中配置相应的图片资源，

否则在不同分辨率机器上都是同样大小（像素点数量），显示出来的大小就不对了。

**Bitmap，占用内存的算法如下：**

图片的`width*height*Config`。如果`Config`设置为`ARGB_8888`，那么上面的`Config`就是4。一张480\*320的图片占用的内存就是480\*320\*4 byte。

压缩图片

使用`BitmapFactory.Options`设置`inSampleSize`就可以缩小图片。属性值`inSampleSize`表示缩略图大小为原始图片大小的几分之一。如果知道图片的像素过大，就可以对其进行缩小。

## 那么如何才知道图片过大呢？

方法是：使用BitmapFactory.Options设置inJustDecodeBounds为true后，再使用decodeFile()等方法，并不会真正的分配空间，即解码出来的Bitmap为null，但是可计算出原始图片的宽度和高度，即options.outWidth和 options.outHeight。通过这两个值，就可以知道图片是否过大了。在实际项目中，先获取图片真实的宽度和高度，然后判断是否需要跑缩小。如果不需要缩小，设置inSampleSize的值为 1。如果需要缩小，则动态计算并设置inSampleSize的值，对图片进行缩小。需要注意的是，在下次使用BitmapFactory的 decodeFile()等方法实例化Bitmap对象前，别忘记将opts.inJustDecodeBound设置回false。否则获取的 bitmap对象还是null。

## 使用actionProvider时出现的问题

bug复现：

```
E 25291 java.lang.ClassCastException: com.ShanFuBao.SmartCall.utils.new_menus_action_provider_preview_options cannot be cast to android.view.ActionProvider
E 25291 at android.view.MenuInflater$MenuState.readItem(MenuInflater.java:374)
E 25291 at android.view.MenuInflater.parseMenu(MenuInflater.java:160)
E 25291 at android.view.MenuInflater.inflate(MenuInflater.java:110)
E 25291 at com.ShanFuBao.SmartCall.clientMenu.newFromHistoryMenu.onCreateOptionsMenu(newFromHistoryMenu.java:295)
E 25291 at android.app.Activity.onCreatePanelMenu(Activity.java:2538)
```

解决方案：

```
1 //import android.support.v4.view.ActionProvider;
2 import android.view.ActionProvider;
```

换一种import的方式即可。tmd，这就是一个坑。

## 1、将Eclipse项目导入到Android studio 中 很多点9图出现问题的解决方法

在build.gradle里添加以下两句：

```
1 aaptOptions.cruncherEnabled = false aaptOptions.useNewCruncher = false
2 1
3 2
4 1
5 2
6
```

用来关闭Android Studio的PNG合法性检查的，直接不让他检查。

## 2、Android Studio 错误: 非法字符: '\uffeff'

原因：Eclipse可以智能的把UTF-8+BOM文件转为普通的UTF-8文件，Android Studio还没有这个功能，所以使用Android Studio编译UTF-8+BOM编码的文件时会出现“非法字符: '\uffeff' ”之类的错误

手动将UTF-8+BOM编码的文件转为普通的UTF-8文件。

用EditPlus打开.Java文件依次：文档 -> 文本编辑 -> 转换文本编码 -> 选择UTF-8编码即可

二、上下文为空 经常出现在fragment界面，打开fragment界面切到后台，然后快速打开其他应用程序，手机上能打开的都打开。再回到这个fragment界面，就有可能出现上下文空指针（经分析，当内存紧张的时候系统回收了。解决方案：判断上下文为空就重新初始化一边就行）

三、数据库不存在 当执行查询的时候报表不存在。（我们这边一直无法复现，小米测试力度比较大我们这边一般就几台机器测试，小米会有很多台，经分析就是在压力

测试的时候，正在建表的时候内存不足没有建立成功。用事务来解决。之前还用过早先把整个数据库删除再建，变相解决了 但是还会删除已经存入的信息。）

Execution failed for task ':app:buildInfoDebugLoader'. 的解决办法

今天在使用Androidstudio运行项目的时候，发现总是报错：

Error:Execution failed for task ':app:buildInfoDebugLoader'.

```
> Exception while doing past iteration backup : Source /home/ruiqin/AndroidStudioProjects/DouTuGaoShou/app/build
/intermediates/builds/debug/5818437215551/classes.dex and destination /home/ruiqin/AndroidStudioProjects/DouTuGao
Shou/app/build/intermediates/builds/debug/5818437215551/classes.dex must be different
```

解决方法：点击菜单栏Build--->RebuildProject重新加载一下再运行，就行了！

## PopupWindow mPopupWindow的问题

设置外面点击是否可消失 小bug 必须要设置一个背景颜色才会有效

```
mPopupWindow.setOutsideTouchable(true);
```

设置一个背景颜色才会有效

```
ColorDrawable colorDrawable = new ColorDrawable(Color.parseColor("#ffffff"));
mPopupWindow.setBackgroundDrawable(colorDrawable);
```

## Android蓝牙开发中遇到的问题

写蓝牙服务端的时候BluetoothServerSocket的accept方法时。老是出现空指针异常，我明明已经在构造方法里面初始化

BluetoothServerSocket 的 serverSocket对象了呀，怎么还会出现为空呢，后来才发现是没有打开蓝牙的缘故，我的确调用了蓝牙适配器的enable()方法，为什么会出现这样的情况呢，后来查看的api，自习阅读了

enable ( ) 方法：

意思是说这是一个异步的方法，他会立即返回结果，但是蓝牙打开有一个过程，要通过接受广播才能判断是否是真正的打开了蓝牙，所以要注册一个action为ACTION\_STATE\_CHANGED的广播，接收到广播之后再去开启服务端的线程，这时再去调用accept方法就不会出现空指针异常了

- 1、已经初始化了BluetoothServerSocket 的 serverSocket对象
- 2、调用accept方法，出现空指针
- 3、原因是没有打开蓝牙，但是调用了蓝牙适配器的enable方法
- 4、解决方法：注册广播来判断是否开启了蓝牙，在接收到广播后再调用accept方法就可以了