<项目名称>

**Android开发建议及常见问题汇总1.0**

文件履历

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本 | 换版/修订记录 | 制/修订 | 审核 | 批准 | 生效日期 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[1 引言 1](#_Toc43886831)

[1.1 编写目的 1](#_Toc43886832)

[1.2 背景 1](#_Toc43886833)

[1.3 定义 1](#_Toc43886834)

[1.4 参考资料 1](#_Toc43886835)

[2 命名建议 2](#_Toc43886836)

[2.1 推荐资源文件 2](#_Toc43886837)

[2.2 layout命名 2](#_Toc43886838)

[2.3 drawable资源命名 2](#_Toc43886839)

[2.4 anim命名 2](#_Toc43886840)

[2.5 color命名 2](#_Toc43886841)

[2.6 dimen 2](#_Toc43886842)

[2.7 style命名 3](#_Toc43886843)

[2.8 string命名 3](#_Toc43886844)

[2.9 资源ID命名 3](#_Toc43886845)

[3 常见问题 4](#_Toc43886846)

[3.1 【系统相关】关于ArmAbi问题 4](#_Toc43886847)

[3.2 【性能相关】Activity间数据通信 4](#_Toc43886848)

[3.3 【性能相关】后台服务服务集中启动问题 4](#_Toc43886849)

[3.4 【性能相关】主线程中创建很多dialog 5](#_Toc43886850)

[3.5 【性能相关】BroadcastReceiver中的onReceive耗时操做 5](#_Toc43886851)

[3.6 【性能相关】线城池的使用Executors与ThreadPoolExecutor 5](#_Toc43886852)

[3.7 【界面相关】ScrollView与ListView/GridView/ExpandableListView嵌套问题 6](#_Toc43886853)

[3.8 【界面相关】过度绘制 6](#_Toc43886854)

[3.9 【界面相关】png图片大小 6](#_Toc43886855)

[3.10 【界面相关】建议：尽量不要使用AnimationDrawable 6](#_Toc43886856)

## 引言

### 编写目的

在实际开发过程中，会遇到很多常见问题，并且这些问题已经是存在解决方案，只要稍加注意便能避免问题的再次发生，因此形成该文档对问题进行总结并不断更新。

本文适合Android应用层开发人员。

### 背景

<说明待开发软件系统的名称；列出此项目的任务提出者、开发者、用户以及将运行该软件的平台。>

### 定义

<列出本文件中用到的专业术语的定义；列出外文首字母缩写词的完整单词及含义。>

### 参考资料

<列出相关参考文件，如：

A．阿里巴巴安卓手册；

B．阿里巴巴java开发手册；

C．Google官方SDK文档。

## 命名建议

### 推荐资源文件

带模块前缀进行命名

### layout命名

* Activity 的 layout 以 module\_activity 开头
* Fragment 的 layout 以 module\_fragment 开头
* Dialog 的 layout 以 module\_dialog 开头
* include 的 layout 以 module\_include 开头
* ListView 的行 layout 以 module\_list\_item 开头
* RecyclerView 的 item layout 以 module\_recycle\_item 开头
* GridView 的行 layout 以 module\_grid\_item 开头

### drawable资源命名

资源名称以小写单词+下划线的方式命名，根据分辨率不同存放 在不同的 drawable 目录下，建议只使用一套,例如 drawable-xhdpi。采用规则如下:

模块名\_业务功能描述\_控件描述\_控件状态限定词

如：module\_login\_btn\_pressed，module\_tabs\_icon\_home\_normal

### anim命名

anim 资源名称以小写单词+下划线的方式命名，采用以下规则

模块名\_逻辑名称\_[方向|序号]

tween 动画资源：尽可能以通用的动画名称命名，如 module\_fade\_in , module\_fade\_out , module\_push\_down\_in (动画+方向)；

frame 动画资源：尽可能以模 块+功能命名+序号。如：module\_loading\_grey\_001

### color命名

color 资源使用#AARRGGBB 格式，写入 module\_colors.xml 文件中，命 名格式采用以下规则：

模块名\_逻辑名称\_颜色

<color name="module\_btn\_bg\_color">#33b5e5e5</color>

### dimen

dimen 资源以小写单词+下划线方式命名，写入 module\_dimens.xml 文件中， 采用下规则：

模块名\_描述信息

如：

<dimen name="module\_horizontal\_line\_height">1dp</dimen>

### style命名

style 资源采用小写单词+下划线方式命名，写入 module\_styles.xml 文件中， 采用以下规则：

父 style 名称.当前 style 名称 如：

<style name="ParentTheme.ThisActivityTheme">

…

</style>

### string命名

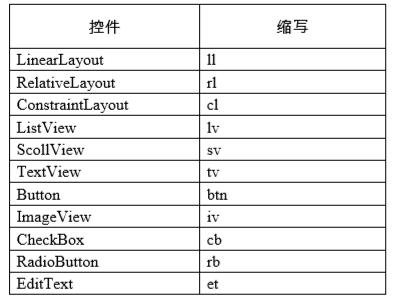
string资源文件或者文本用到字符需要全部写入module\_strings.xml文件中， 字符串以小写单词+下划线的方式命名，采用以下规则：

模块名\_逻辑名称

如：moudule\_login\_tips，module\_homepage\_notice\_des

### 资源ID命名

Id 资源原则上以驼峰法命名，View 组件的资源 id 需要以 View 的缩写作为 前缀。常用缩写表如下



其它控件的缩写推荐使用小写字母并用下划线进行分割，例如：

* ProgressBar 对应的缩写为 progress\_bar
* DatePicker 对应的缩写为 date\_picker

## 常见问题

### 【系统相关】关于ArmAbi问题

问题现象

Caused by: java.lang.ClassNotFoundException: Didn't find class "com.qorosauto.tsp.service.BaseApplication" on path: DexPathList[[zip file "/system/app/TspSdk\_Release/TspSdk\_Release.apk"],nativeLibraryDirectories=[/system/app/TspSdk\_Release/lib/arm, /system/app/TspSdk\_Release/TspSdk\_Release.apk!/lib/armeabi-v7a, /system/lib, /vendor/lib, /system/lib, /vendor/lib]]

问题原因

系统在确定应用的armabi时错误的将其设置成了32位，导致此错误设置的出现，是因为相同sharedUserId的某个应用使用了32位的so，系统根据自身的算法将没有配置armabi的应用默认的配置32位的abi。

造成影响

1. 应用程序崩溃；
2. 核心服务崩溃，导致系统进入拯救模式；

解决办法

在makefile中配置如下语句，促使系统强制将armabi设置成64位：

$(shell mkdir -p $(PRODUCT\_OUT)/system/app/\*\*\*/lib/arm64)

### 【性能相关】Activity间数据通信

问题现象

TransactionTooLargeException

问题原因

数据量过大时，使用intent + Parcelable方式传输数据。

解决办法

可以考虑使用EventBus等替代。

### 【性能相关】后台服务服务集中启动问题

问题现象

系统性能突然下降，出现大量的slow信息。

问题原因

很多服务集中的在某个时间段内启动。导致系统出现繁忙，影响了性能和启动时间；

解决办法：

1. 暂不使用的服务，可以在使用前触发启动；
2. 一定要启动的服务，可以策略性推迟每个服务的启动时间点，避免同时的去启动；
3. 自己的服务最好自己通过某种策略启动，而不是交给三方模块去启动（除非业务需要）；

### 【性能相关】主线程中创建很多dialog

问题现象

activity 启动慢；

问题原因

在onCreate时预先创建了未来可能要使用的dialog，导致onCreate花费很多时间；从而导致启动慢。

解决办法

1. 在onCreate主线程中不要预先创建dialog对象
2. 当业务触发时，再去创建dialog实例，且保证只new一次即可。多次创建会产生大量的内存碎片，导致GC从而影响新能；

### 【性能相关】BroadcastReceiver中的onReceive耗时操做

问题现象

onReceive一直不返回导致，UI不流畅甚至ANR.

问题原因

由于该方法是在主线程中执行，若果耗时操作会导致UI不流畅，甚至导致anr；

解决办法

可以使用IntentService、创建HandlerThread等方式，在其他worker线程中执行相关业务。

### 【性能相关】线程池的使用Executors与ThreadPoolExecutor

问题现象

OOM;

问题原因

ExecutorService cachedThreadPool = Executors.newCachedThreadPool();

1) FixedThreadPool 和 SingleThreadPool ： 允 许 的 请 求 队 列 长 度 为 Integer.MAX\_VALUE，可能会堆积大量的请求，从而导致 OOM；

2) CachedThreadPool 和 ScheduledThreadPool：允许的创建线程数量为 Integer.MAX\_VALUE，可能会创建大量的线程，从而导致 OOM

解决办法

使用ThreadPoolExecutor方式创建，如下：

**int** NUMBER\_OF\_CORES = Runtime.*getRuntime*().availableProcessors();  
**int** KEEP\_ALIVE\_TIME = 1;  
TimeUnit KEEP\_ALIVE\_TIME\_UNIT = TimeUnit.SECONDS;  
BlockingQueue<Runnable> taskQueue = **new** LinkedBlockingQueue<Runnable>();  
ExecutorService executorService = **new** ThreadPoolExecutor(NUMBER\_OF\_CORES,  
 NUMBER\_OF\_CORES\*2, KEEP\_ALIVE\_TIME, KEEP\_ALIVE\_TIME\_UNIT,  
 taskQueue, **new** BackgroundThreadFactory(), **new** DefaultRejectedExecutionHandler());

### 【界面相关】ScrollView与ListView/GridView/ExpandableListView嵌套问题

问题现象

ListView/GridView/ExpandableListView，加载额全部的item到内存中；消耗大量的内存和CPU资源去绘制界面

ScrollView中嵌套 ListView 或 GridView等的做法官方明确禁止。除了开发过程中遇到 的各种视觉和交互问题，这种做法对性能也有较大损耗。ListView 等 UI 组件自身有 垂直滚动功能，也没有必要在嵌套一层 ScrollView。目前为了较好的 UI 体验， 更贴 近 Material Design 的设计，推荐使用 NestedScrollView。

### 【界面相关】过度绘制

问题现象

过度绘制

解决办法

禁止在设计布局时多次设置子 view 和父 view 中为同样的背景造成页面过 度绘制，推荐将不需要显示的布局进行及时隐藏。

### 【界面相关】PNG图片大小

对于打的PNG可以使用Tinypng或者其他工具压缩处理，减少包的体积

<https://tinypng.com/> 可在线处理PNG图片。

### 【界面相关】建议：尽量不要使用AnimationDrawable

问题现象

内存消耗大，且容易报错

问题原因

AnimationDrawable会一次加载大量图片，导致一瞬间占用大量内存，导致OOM。

建议

当图片较少时可以使用，当图片较多时不建议使用，可以通过自定义动画等方式来控制内存的使用及动画策略，同时要优化图片的大小来减轻内存消耗。