

Android开发规范

一、Android开发编码规范

1、java代码中不出现中文，最多注释中可以出现中文

2、局部变量命名、静态成员变量命名

只能包含字母，单词首字母出第一个外，都为大写，其他字母都为小写

3、常量命名

只能包含字母和_，字母全部大写，单词之间用_隔开

4、layout中的id命名

命名模式为：view缩写_模块名称_view的逻辑名称

view的缩写详情如下

LayoutView: lv RelativeLayout:rv TextView:tv ImageView:iv ImageButton:im Button:btn

layout的缩写详情如下

FrameLayout: fraLay RelativeLayout:relLay TextView:tv ImageView:iv ImageButton:im
Button:btn

5、activity中的view变量命名

命名模式为：逻辑名称+view缩写

建议：如果layout文件很复杂，建议将layout分成多个模块，每个模块定义一个moduleViewHolder，其成员变量包含所属view

6、strings.xml中的id命名

命名模式：activity名称_功能模块名称_逻辑名称/activity名称_逻辑名称/common_逻辑名称

strings.xml中，使用activity名称注释，将文件内容区分开来

drawable中的图片命名

命名模式：activity名称_逻辑名称/common_逻辑名称

7、styles.xml：将layout中不断重现的style提炼出通用的style通用组件，放到styles.xml中；

8、使用layer-list和selector

- 9、图片尽量分拆成多个可重用的图片
- 10、服务端可以实现的，就不要放在客户端
- 11、引用第三方库要慎重，避免应用大容量的第三方库，导致客户端包非常大
- 12、处理应用全局异常和错误，将错误以邮件的形式发送给服务端
- 13、图片的处理
- 14、使用静态变量方式实现界面间共享要慎重
- 15、Log(系统名称模块名称接口名称，详细描述)
- 16、单元测试（逻辑测试、界面测试）
- 17、不要重用父类的handler，对应一个类的handler也不应该让其子类用到，否则会导致message.what冲突
- 18、activity中在一个View.OnClickListener中处理所有的逻辑
- 19、strings.xml中使用%1\$s实现字符串的通配
- 20、如果多个Activity中包含共同的UI处理，那么可以提炼一个CommonActivity，把通用部分叫由它来处理，其他activity只要继承它即可
- 21.使用button+activitygroup实现tab效果时，使用Button.setSelected(true)，确保按钮处于选择状态，并使activitygroup的当前activity与该button对应
- 22、如果所开发的为通用组件，为避免冲突，将drawable/layout/menu/values目录下的文件名增加前缀
- 23、数据一定要效验，例如
字符型转数字型，如果转换失败一定要有缺省值；
Android开发服务端响应数据是否有效判断；

二、Android开发性能优化

1、http用gzip压缩，设置连接超时时间和响应超时时间

http请求按照业务需求，分为是否可以缓存和不可缓存，那么在无网络的环境中，仍然通过缓存的httpresponse浏览部分数据，实现离线阅读。

2、listview性能优化

(1) 复用convertView

在getItemView中，判断convertView是否为空，如果不为空，可复用。如果convertView中的view需要添加listener，代码一定要在if(convertView==null){}之外。

(2) 异步加载图片

item中如果包含有webimage，那么最好异步加载

(3) 快速滑动时不显示图片

当快速滑动列表时（SCROLL_STATE_FLING），item中的图片或获取需要消耗资源的view，可以不显示出来；而处于其他两种状态（SCROLL_STATE_IDLE和SCROLL_STATE_TOUCH_SCROLL），则将那些view显示出来

3、使用线程池，分为核心线程池和普通线程池，下载图片等耗时任务放置在普通线程池，避免耗时任务阻塞线程池后，导致所有异步任务都必须等待

4、异步任务，分为核心任务和普通任务，只有核心任务中出现的系统级错误才会报错，异步任务的ui操作需要判断原activity是否处于激活状态

5、尽量避免static成员变量引用资源耗费过多的实例,比如Context

6、使用WeakReference代替强引用，弱引用可以保持对对象的引用，同时允许GC在必要时释放对象，回收内存。对于那些创建便宜但耗费大量内存的对象，即希望保持该对象，又要在应用程序需要时使用，同时希望GC必要时回收时，可以考虑使用弱引用。

7、超级大胖子Bitmap

及时的销毁(Activity的onDestroy时将bitmap回收，在被UI组件使用后马上进行回收会抛RuntimeException:Canvas:trying to use a recycled bitmap android.graphics.Bitmap)设置一定的采样率(有开发者提供的图片无需进行采样，对于有用户上传或第三方的大小不可控图片，可进行采样减少图片所占的内存)，从服务端返回图片，建议同时反馈图片的size巧妙的运用软引用drawable对应resid的资源，bitmap对应其他资源任何类型的图片，如果获取不到（例如文件不存在，或者读取文件时跑OutOfMemory异常），应该有对应的默认图片（默认图片放在在apk中，通过resid获取）；

8、保证Cursor占用的内存被及时的释放掉，而不是等待GC来处理。并且Android明显是倾向于编程者手动的将Cursorclose掉

9、线程也是造成内存泄露的一个重要的源头。线程产生内存泄露的主要原因在于线程生命周期的不可控

10、如果ImageView的图片是来自网络，进行异步加载

11、应用开发中自定义View的时候，交互部分，千万不要写成线程不断刷新界面显示，而是根据TouchListener事件主动触发界面的更新

12、Drawable

ui组件需要用到的图片是apk包自带的，那么一律用setImageResource或者setBackgroundResource，而不要根据resourceid

注意：get(getResources(),R.drawable.btn_achievement_normal)该方法通过resid转换为drawable，需要考虑回收的问题，如果drawable是对象私有对象，在对象销毁前是肯定不会释放内存的。

三、Android开发UI 优化

1、layout组件化，尽量使用merge及include复用

2、使用styles，复用样式定义

3、软键盘的弹出控制，不要让其覆盖输入框

4、数字、字母和汉字混排占位问题：将数字和字母全角化。由于现在大多数情况下我们的输入都是半角，所以字母和数字的占位无法确定，但是一旦全角化之后，数字、字母的占位就和一个汉字的占位相同了，这样就可以避免由于占位导致的排版问题。

5、英文文档排版：textview自动换行时要保持单词的完整性，解决方案是计算字符串长度，然后手动设定每一行显示多少个字母并加上‘\n’

6、复杂布局使用RelativeLayout

7、自适应屏幕，使用dp替代pix

8、使用android:layout_weight或者TableLayout制作等分布局

9、使用animation-list制作动画效果

代码元素变量命名：

元素	全称	缩写
布局	RelativeLayout	xxxRelLay
布局	FrameLayout	xxxFrmLay
布局	GridLayout	xxxGriLay
布局	TableLayout	xxxTabLay
布局	LinearLayout	xxxLinLay
元素	ListView	xxxLv
元素	ScrolView	xxxSv
元素	TextView	xxxTv
元素	ImageView	xxxIv
元素	TextEdit	xxxEdit

元素	CheckBox	xxxBox
元素	Spinner	xxxSpinner
元素	Radio	xxxRadio
元素	Button	xxxButt
元素	AlertDialog	xxxDialog
元素	Numberpicker	xxxNumPicker
元素	Datepicker	xxxDatePicker
元素	Progress	xxxProgress