Anroid开发规范文档

Text:宋体 size:8

1、android开发准备

1.1 开发环境使用android studio2.2

1.2 安装.ignore（配合git使用） markdown surport（md文件支持）这两款插件

1.3 采用markdown记录开发中的各种问题

1.4 务必在使用git前在项目根目录中添加.gitignore文件（用.ignore插件生成）

2.开发规定

2.1文件编码

源文件编码格式为 UTF-8

2.2命名规范

2.2.1 类名 采用大驼峰驼峰命名法，即功能名＋类型名的方式来命名类名及方法名

| **类** | **命名格式** | **示例** |
| --- | --- | --- |
| Activity | XXX功能+Activity | 如主界面HomeActivity,启动页LauncherActivity |
| Service | XXX功能+Service | 如消息推送的Service，PushService或PushMessageService |
| BroadcastReceiver | XXX功能+Receiver | 如在线的消息广播接受者，OnlineReceiver |
| ContentProvider | XXX功能+Provider | 如联系人的内容提供者，ContactsProvider |
| Fragment | XXX功能+Fragment | 如显示联系人的Fragment，ContactsFragment |
| Dialog | XXX功能+Dialog | 如普通的选择提示对话框，ChoiceDialog |
| Adapter | XXX功能+XX类型控件Adapter | 如联系人列表，ContactsListAdapter |
| 基础功能类 | Base+XX父类名 | 如BaseActivity，BaseFragment |
| 工具类 | XXX功能+Utils | 如处理字符串的工具类，StringUtils |
| 管理类 | XXX功能+Manager | 如管理联系人的类，ContactsManager |

2.2.2 接口命名 和类名基本一致。也可以在接口名前面再加一个大写的I，表明这是一个接口Interface。如： 可 以表明一个信息是否可以分享的接口，可以命名为Shareable，也可以是IShareable。

2.2.3 方法名 动词或动名词，采用小驼峰命名法

| **命名风格** | **含义** |
| --- | --- |
| initXX() | 初始化，如初始化所有控件initView() |
| isXX() | 是否满足某种要求，如是否为注册用户isRegister() |
| processXX() | 对数据做某些处理，可以以process作为前缀 |
| displayXX() | 显示提示信息，如displayXXDialog，displayToast，displayXXPopupWindow |
| saveXX() | 保存XX数据 |
| resetXX() | 重置XX数据 |
| addXX()/insertXX() | 添加XX数据 |
| deleteXX()/removeXX() | 删除XX数据 |
| updateXX() | 更新XX数据 |
| searchXX()/findXX()/queryXX() | 查找XX数据 |
| draw() | 控件里面使用居多，例如绘制文本drawText |

2.2.4 变量采用小驼峰命名法   
 非公有，非静态成员变量命名前面加m（member，表示成员变量之意），如，控件的宽高 mWidth，mHeight   
 静态类变量前面加s（static，表示静态变量之意），如，一个静态的单例 sSingleInstance   
 公有非静态字段命名以p开头   
 公有静态字段（全局变量）命名以g开头   
 局部变量名

2.2.5 局部变量名以LowerCamelCase风格编写，比起其它类型的名称，局部变量名可以有更为宽松的缩写。

2.2.6 虽然缩写更宽松，但还是要避免用单字符进行命名，除了临时变量和循环变量。

2.2.7 即使局部变量是final和不可改变的，也不应该把它示为常量，自然也不能用常量的规则去 命名它。

2.2.8 临时变量

临时变量通常被取名为i，j，k，m和n，它们一般用于整型；c，d，e，它们一般用于字型如： for (int i = 0; i < len ; i++)，并且它和第一个单词间没有空格。

2.2.9 布局文件名称的定义必须为小写字母，否者无法生成R类，尽量不要用缩写。以表达清楚该文件用途为本， 通常情况下用下划线连接各语义单词   
 例如dialog\_title\_icons.xml 或者list\_menu\_item\_checkbox.xml

2.2.10 包命名 采用反域名命名规则，全部使用小写字母 例如：cn.qhebusbar.bbcx.view.activity

一级包名为com;

二级包名为xx（可以是公司或则个人的随便）;

三级包名应用的英文名app\_name;

四级包名为模块名或层级名;

2.2.10 常量 全部大写,采用下划线命名法.如：MIN\_WIDTH,MAX\_SIZE

| **布局类型** | **命名风格** |
| --- | --- |
| Activity的xml布局 | activity\_+XX功能，如主页面activity\_home |
| Fragment的xml布局 | fragment\_+XX功能，如联系人模块fragment\_contacts |
| Dialog的xml布局 | dialog\_+XX功能，如选择日期dialog\_select\_date |
| 抽取出来复用的xml布局（include） | include\_+XX功能，如底部tab栏include\_bottom\_tabs |
| ListView或者RecyclerView的item xml布局 | XX功能+\_list\_item，如联系人的contact\_info\_list\_item |
| GridView的item xml布局 | XX功能+\_grid\_item，如相册的album\_grid\_item |

2.2.11 布局资源文件 全部小写，采用下划线命名法

2.2.12 动画资源文件(anim文件夹下) 全部小写，采用下划线命名法，加前缀区分-

| **动画效果** | **命名风格** |
| --- | --- |
| 淡入/淡出 f | fade\_in/fade\_out |
| 从某个方向淡入/淡出 | fade\_方向\_in(out),右边淡入淡出fade\_right\_in(out) |
| 从某个方向弹入/弹出 | push\_方向\_in(out),右边推入推出push\_right\_in(out) |
| 从某个方向滑入/滑出 | slide\_in(out)from方向,右边滑入滑出slide\_in(out)\_from\_right |

2.2.13 strings和colors资源文件小驼峰命名法,命名风格大致如下：

string命名格式：XX界面\_XX功能\_str,如 activity\_home\_welcome\_str

color命名格式：color\_16进制颜色值，如红色 color\_ff0000

像string通常建议把同一个界面的所有string都放到一起，方便查找。而color的命名则省去我们头疼 的想这个颜色怎么命名。

2.2.14 selecor、drawable、layer-list资源文件 小驼峰命名法。命名风格通常都是XX\_selector、XX\_drawable、 XX\_layer。

举两个比较常用的例子：

1. 按钮按压效果button\_selector，正常状态命名为button\_normal(XX\_normal)，按压状态命名为 button\_pressed(XX\_pressed)

2. 选择效果checkbox\_selector,未选中状态命名为checkbox\_unchecked(XX\_unchecked),选中状态为 checkbox\_checked(XX\_checked)

styles、dimens资源文件   
 1. style采用大驼峰命名法，主题可以命名为XXTheme,控件的风格可以命名为XXStyle   
 2. dimen采用小驼峰命名法，如所有Activity的titlebar的高度，activity\_title\_height\_dimen

2.2.15控件id命名

| **控件** | **前缀缩写** |
| --- | --- |
| RelativeLayout | rl |
| LinearLayout | ll |
| FrameLayout | fl |
| TextView | tv |
| Button | btn |
| ImageButton | imgBtn |
| ImageView | imgView或iv |
| CheckBox | chk |
| RadioButton | rdoBtn |
| analogClock | anaClk |
| DigtalClock | dgtClk |
| DatePicker | dtPk |
| EditText | edtTxt或et |
| TimePicker | tmPk |
| toggleButton | tglBtn |
| ProgressBar | proBar |
| SeekBar | skBar |
| AutoCompleteTextView | autoTxt |
| ZoomControl | zmCtl |
| VideoView | vdoVi |
| WebView | webVi |
| Spinner | spn |
| Chronometer | cmt |
| ScollView | sclVi |
| TextSwitch | txtSwt |
| ImageSwitch | imgSwt |
| ListView | lv |
| GridView | gv |
| ExpandableList | epdLt |
| MapView | mapVi |

2.3编码规范

2.3.1 import语句

import不要使用通配符

不要换行

顺序和间距   
 import语句可分为以下几组，按照这个顺序，每组由一个空行分隔：

所有的静态导入独立成组

com.google imports(仅当这个源文件是在com.google包下)

第三方的包。每个顶级包为一组，字典序。例如：android, com, junit, org, sun

java imports5.javax imports组内不空行，按字典序排列。

2.3.2 要分门别类存放各种类

2.3.3 Layout中的常量要在strings.xml文件中定义  
 以下使用方式错误：   
 <TextView **[Android](http://lib.csdn.net/base/android" \o "Android知识库" \t "http://blog.csdn.net/wangkuan219/article/details/_blank)**:text="评论">

2.3.4 Layout中所有空间字体大小，都定义在dimens.xml中

<resources>

<dimen name="bottom\_tab\_font\_size">12sp</dimen>

<dimen name="bottom\_tab\_padding\_up">5dp</dimen>

<dimen name="bottom\_tab\_padding\_drawable">8dp</dimen>

</resources>

2.3.5 Activity中，onCreate（）方法拆分为3个部分

initViews（）；加载layout布局文件，初始化控价

initDatas（）：调用MobileAPI；

initEvents（）：点击事件回调（看具体情况）

2.3.6 使用fastJSON自定义实体来作为MobileAPI的数据载体

2.3.7 页面之间传值，坚持使用Intent携带序列化实体数据的方式。禁止为了省事使用全局变量进行传值的方式

2.3.8 为空间添加事件，禁止在layout中直接声明事件的方法，以下代码不允许  
 <Button android:onclick="gotoLogin"..../>

2.3.9 图片的处理，请统一使用第三方组建ImageLoader或Fresco来进行异步加载

2.3.10 SharedPreferences仅用于简单的配置信息，例如设置页面的各种开关。对于复制的对象，还是要保存在 本地文件中

2.3.11 Activity中不要嵌套内部类，尽量独立出来

2.3.12 Javadoc

格式 Javadoc块的基本格式如下所示：

/\*\*

\* Multiple lines of Javadoc text are written here,

\* wrapped normally...

\*/

public int method(String p1) { ... }

或者是以下单行形式：

/\*\* An especially short bit of Javadoc. \*/

2.3.13 段落   
 空行(即，只包含最左侧星号的行)会出现在段落之间和Javadoc标记(@XXX)之前(如果有的话)。   
 除了第一个段落，每个段落第一个单词前都有标签，并且它和第一个单词间没有空格。

2.3.14 Javadoc标记   
 标准的Javadoc标记按以下顺序出现：@param, @return, @throws, @deprecated,   
 前面这4种标记如果出现，描述都不能为空。 当描述无法在一行中容纳，连续行需要至少再缩进4个空格。

2.3.15 摘要片段   
 每个类或成员的Javadoc以一个简短的摘要片段开始。这个片段是非常重要的，在某些情况下，它是唯一出 现的文本，比如在类和方法索引中。