Android开发规范手册V1.2.0

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **编写者** | **日期** | **内容** |
| V1.1.0 | HaiyuKing |  | 创建文档 |
| V1.2.0 | HaiyuKing |  | 完善文档 |

# Android Studio规范

* 尽量使用最新版的IDE进行开发
* 编码格式统一为UTF-8
* 自动删除多余的import，减少警告出现，可利用AS的Optimize Imports(Settings → Keymap → Optimize Imports)快捷键

# 项目样式规范

## 项目Style样式

为了兼容低版本的ADT。便于移植到其他开发环境中。新建项目时不使用Theme并且修改styles.xml文件中的APPBaseTheme为android:Theme.Light.NoTitleBar

# 命名规范

## 项目名称命名规范

[公司简称]\_[地区简称]\_[项目名称简称]\_[APP/Pad/IOS]

根据公司实际情况灵活处理。

*备注：*

* *如果项目有所属地区，则写地区简称，比如BJ（北京），HB（河北）；如果没有则不写*
* *APP——手机应用；Pad——平板应用。如果不确定机型或者两者都有，则不写*

## 项目主包名命名规范

com.[公司简称].[地区简称].[项目名称简称]

根据公司实际情况灵活处理。

*备注：*

* *如果项目有所属地区，则写地区简称，比如BJ（北京），HB（河北）；如果没有则不写*

## 项目子包名命名规范【按功能分包】

[项目主包名].[组件类型]

例如：

com.why.project.demo.activity

com.why.project.demo.fragment

com.why.project.demo.dialog

com.why.project.demo.adapter

com.why.project.demo.model【存放实体类】

com.why.project.demo.utils【工具类】

com.why.project.demo.utils.fileutil【工具类——file工具类】

参考Google I/O 2015的代码结构，按功能分包具体可以这样做：

|  |
| --- |
| src  └─com  └─why  └─demo  │ AppApplication.java 定义Application类  │ Config.java 定义配置数据（常量）  │  ├─framework  │ 定义interface以及相关基类  │  ├─io  │ 数据定义（model）、数据操作（比如json解析，但不包括db操作）  │  ├─model  │ 定义model（数据结构以及getter/setter、compareTo、equals等等，不含复杂操作）  │ 以及modelHelper（提供便于操作model的api）  │  ├─provider  │ 实现ContentProvider，及其依赖的db操作  │  ├─receiver  │ 实现Receiver  │  ├─service  │ 实现Service（比如IntentService），用于在独立线程中异步do stuff  │  ├─ui  │ 实现BaseActivity，以及自定义view和widget，相关的Adapter也放这里  │  ├─util  │ 实现工具类，提供静态方法  │  ├─feature1  │ Item.java 定义model  │ ItemHelper.java 实现modelHelper  │ feature1Activity.java 定义UI  │ feature1DAO.java 私有db操作  │ feature1Utils.java 私有工具函数  │ ...其它私有class  │  ├─...其它feature |

## 类名命名规范

[功能简称][组件类型]

类名使用 UpperCamelCase 风格，必须遵从驼峰形式

例如：

LoginActivtiy【登录界面】

SetupModifyPwdDialog【设置密码对话框界面】

|  |  |  |
| --- | --- | --- |
| **类** | **描述** | **例如** |
| Activity 类 | Activity为后缀标识 | 欢迎页面类WelcomeActivity |
| Adapter类 | Adapter 为后缀标识 | 新闻详情适配器 NewDetailAdapter |
| 解析类 | Parser为后缀标识 | 首页解析类HomePosterParser |
| 工具方法类 | Utils或Manager为后缀标识（与系统或第三方的Utils区分）或功能+Utils | 线程池管理类：ThreadPoolManager日志工具类：LogUtils（Logger也可）打印工具类：PrinterUtils |
| 数据库类 | 以DBHelper后缀标识 | 新闻数据库：NewDBHelper |
| Service类 | 以Service为后缀标识 | 时间服务TimeService |
| BroadcastReceiver类 | 以Receiver为后缀标识 | 推送接收JPushReceiver |
| ContentProvider类 | 以Provider为后缀标识 | ShareProvider |
| 自定义的共享基础类 | 以Base开头 | BaseActivity , BaseFragment |

## 变量命名规范

m\_[功能简称][控件简称]

尽量使用望文知义的单词

方法名、参数名、成员变量、局部变量都统一使用 lowerCamelCase 风格，必须遵从

驼峰形式。

例如：

private ImageView mBackImg;【返回图标】

private EditText mUserNameEdt;【用户名输入框】

*备注：控件简称可以省略，不过尽量不要省略。*

**scope：范围**

非公有，非静态字段命名以m开头。

静态字段命名以s开头。

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。

|  |
| --- |
| public class MyClass {  int mPackagePrivate;  private int mPrivate;  protected int mProtected;  private static MyClass sSingleton;  public int pField;  public static int gField;  } |

## 方法名命名规范

尽量使用望文知义的单词

方法名、参数名、成员变量、局部变量都统一使用 lowerCamelCase 风格，必须遵从

驼峰形式。

方法名通常是动词或动词短语。

例如：

initViews()【初始化控件】

initDatas()【初始化数据】

initEvents()【初始化监听事件】

|  |  |
| --- | --- |
| **方法** | **说明** |
| initXX() | 初始化相关方法,使用init为前缀标识，如初始化布局initView() |
| isXX() checkXX() | 方法返回值为boolean型的请使用is或check为前缀标识 |
| getXX() | 返回某个值的方法，使用get为前缀标识 |
| setXX() | 设置某个属性值 |
| handleXX()/processXX() | 对数据进行处理的方法 |
| displayXX()/showXX() | 弹出提示框和提示信息，使用display/show为前缀标识 |
| updateXX() | 更新数据 |
| saveXX() | 保存数据 |
| resetXX() | 重置数据 |
| clearXX() | 清除数据 |
| removeXX() | 移除数据或者视图等，如removeView(); |
| drawXX() | 绘制数据或效果相关的，使用draw前缀标识 |

## 常量命名规范

常量命名全部大写，单词间用下划线隔开，力求语义表达完整清楚，不要嫌名字长。

例如：

public static final String EXTRA\_TITLE = "webTitle";

private static final String TAG = "LoginActivity";

# 资源文件命名规范

## 资源文件(XML文件、layout布局文件)命名规范

[组件类型]\_[功能简称]]

全部小写，采用下划线命名法，使用名词或名词词组

例如：

activity\_main.xml

adapter\_query\_list\_item.xml

fragment\_home\_database.xml

navigationbar\_home.xml

## 布局文件中控件的ID命名规范

[控件缩写]\_[模块简称]\_[功能简称]

全部小写，单词间用下划线隔开，力求语义表达完整清楚，不要嫌名字长。

见[《UI控件缩写表》](#_UI控件缩写表)

例如：

@+id/layout\_home

@+id/edt\_username

@+id/tv\_submit

@+id/list\_undo

@+id/web\_setupPwd

## 资源图片命名规范

[功能简称]\_[模块名]

或者

[逻辑名称]\_[功能简称]

全部小写，采用下划线命名法，加前缀区分

命名模式：可加后缀 \_small 表示小图， \_big 表示大图，模块名、逻辑名称可由多个单词加下划线组成。

功能简称也可以是UI控件缩写，见[《UI控件缩写表》](#_UI控件缩写表)

例如：

pwd\_img.png【密码图标】

setup\_logout.png【设置界面安全退出的图标】

如果有多种形态，参考下表进行命名：

|  |  |
| --- | --- |
| **名称** | **说明** |
| sel\_btn\_xx | 按钮图片使用btn\_整体效果（selector） |
| btn\_xx\_normal | 按钮图片使用btn\_正常情况效果 |
| btn\_xx\_pressed | 按钮图片使用btn\_点击时候效果 |
| btn\_xx\_focused | state\_focused聚焦效果 |
| btn\_xx\_disabled | state\_enabled (false)不可用效果 |
| btn\_xx\_checked | state\_checked选中效果 |
| btn\_xx\_selected | state\_selected选中效果 |
| btn\_xx\_hovered | state\_hovered悬停效果 |
| btn\_xx\_checkable | state\_checkable可选效果 |
| btn\_xx\_activated | state\_activated激活的 |
| btn\_xx\_windowfocused | state\_window\_focused |

## 动画文件（anim文件夹下）

[模块名称]\_[功能缩写]

全部小写，采用下划线命名法

例如：

activity\_alpha\_in.xml【界面跳转淡入动画】

普通的tween动画采用如下表格中的命名方式：[动画类型]\_[方向]

|  |  |
| --- | --- |
| **名称** | **说明** |
| fade\_in | 淡入 |
| fade\_out | 淡出 |
| push\_down\_in | 从下方推入 |
| push\_down\_out | 从下方推出 |
| push\_left | 推向左方 |
| slide\_in\_from\_top | 从头部滑动进入 |
| zoom\_enter | 变形进入 |
| slide\_in | 滑动进入 |
| shrink\_to\_middle | 中间缩小 |

## values中name命名（values文件夹下）

colors.xml、strings.xml、dimens.xml、styles.xml文件

[模块名称]\_[功能简称]

全部小写，采用下划线命名法

例如：

|  |
| --- |
| <!-- \*\*\*\*\*\*\*登录界面\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->  <!—登录按钮的文字颜色 -->  <color name="login\_btn\_textColor">#B8B8B8</color> |

|  |
| --- |
| <!-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*首页界面\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* -->  <!-- 标题的文字 -->  <string name="home\_title\_text">这是个标题</string> |

# 注释规范

* 本着的原则是：尽量添加注释，且完整、简洁。
* 代码修改的同时，注释也要进行相应的修改，尤其是参数、返回值、异常、核心逻辑

等的修改。

**注意：不同版本的Android Studio的规范不同，比如类文件注释：**

低版本上下面的写法是正常的：

|  |
| --- |
| /\*\*  \* **@Created** HaiyuKing  \* **@Used** 登录界面  \*/ |

但是在高版本上需要写成下面的样式：

|  |
| --- |
| /\*\*  \* Created by HaiyuKing  \* Used 登录界面  \*/ |

## 类文件注释规范

类、类属性、类方法的注释必须使用 Javadoc 规范，使用/\*\*内容\*/格式，不得使用

// xxx 方式。

|  |
| --- |
| /\*\*  \* **@Created** HaiyuKing  \* **@Used** 登录界面  \*/  **public** **class** LoginActivity **extends** BaseActivity{} |

|  |
| --- |
| /\*\*  \* **@Created** HaiyuKing  \* **@Used** 首页列表的适配器  \*/  **public** **class** HomeListViewAdapter **extends** RecyclerView.Adapter<ViewHolder>{} |

|  |
| --- |
| /\*\*  \* **@Created** HaiyuKing  \* **@Used** 首页界面查询碎片界面  \*/  **public** **class** HomeQueryFragment **extends** BaseFragment{} |

*备注：所有的类都必须添加创建者和用途。*

具体可以在AS中自己配制，Settings → Editor → File and Code Templates → Includes → File Header

## 变量注释规范

|  |
| --- |
| //用户名输入框  **private** EditText mUserNameEdt; |

## 方法注释规范

每一个成员方法（包括自定义成员方法、覆盖方法、属性方法）的方法头都必须做方法头注释。

|  |
| --- |
| /\*\*  \* 初始化控件  \*/  **private** **void** initView() {} |

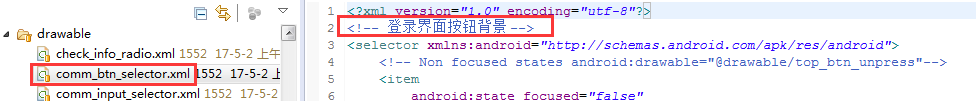
|  |
| --- |
| /\*\*  \* 系统登录  \* **@param** context 上下文  \* **@param** userName 用户名  \* **@param** password 密码  \*/  **public** String Login(Context context, String userName, String password,StringCallback callback)  **throws** Exception {} |

## 块注释

块注释与其周围的代码在同一缩进级别。它们可以是/\* ... \*/风格，也可以是// ...风格(**//后最好带一个空格**）。对于多行的/\* ... \*/注释，后续行必须从\*开始， 并且与前一行的\*对齐。

## Layout、drawable布局文件注释规范

使用这个<!-- 🡪进行注释。



# 废弃代码规范

代码被注释掉有两种可能性：

1 ） 后续会恢复此段代码逻辑。

2 ） 永久不用。前者如果没有备注信息，难以知晓注释动机。后者建议直接删掉 （ 因为代码仓库保存了历史代码 ） 。

## 废弃类、布局文件注释规范

在类文件注释、布局文件注释区域标注“已废弃”或者“暂时废弃”字样。并简要说明废弃原因。

## 废弃代码注释规范

废弃代码需要注释掉，并且标注“已废弃”或者“暂时废弃”字样。并简要说明废弃原因。

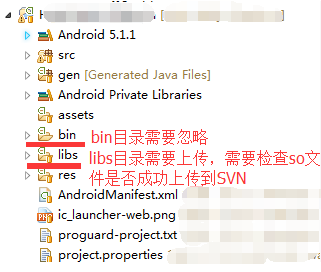
## 废弃资源及时清理

当项目发布一个版本的时候，及时清理废弃的类文件、图片资源等。

# SVN规范

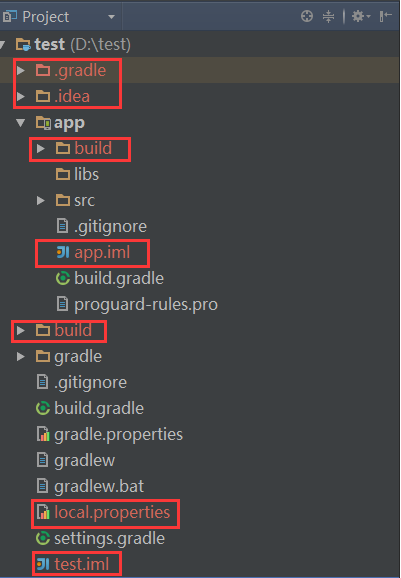
## 上传项目到SVN忽略文件规范

### Eclipse



### Android Studio

现在红色代表忽略的文件，白色代表版本控制的文件。



## 代码提交的规范

* 按照功能点作为节点进行提交，一天可以提交多次。
* 不可以多个独立的功能点一起提交。
* 提交注释一定要写明修改、完善的功能（一定要准确，可以写的文字多点）

# 代码混淆规范

发布前必须进行代码混淆

# 引用第三方库规范

优先使用本公司内部的第三方库

# 其他规范

## Android开发规范

<https://github.com/Blankj/AndroidStandardDevelop>

## 阿里巴巴 Android 开发手册

链接：https://pan.baidu.com/s/1m1NkHiJPt6k-9DbkO7pstg 密码：ba5e

# UI控件缩写表

|  |  |
| --- | --- |
| **名称** | **缩写** |
| TextView | tv |
| EditText | edt |
| ImageButton | ib |
| Button | btn |
| ImageView | img |
| ListView | list |
| GridView | grif |
| ProgressBar | progress |
| SeekBar | seekbar |
| RadioButtion | rb |
| CheckBox | cb |
| ScrollView | scroll |
| LinearLayout | layout |
| FrameLayout | flayout |
| RelativeLayout | rlayout |
| RecyclerView | rv |
| WebView | web |
| VideoView | video |
| Spinner | spinner |
| ToggleButton | tb |

# 常见的英文单词缩写表

|  |  |
| --- | --- |
| **名称** | **缩写** |
| icon | ic （主要用在app的图标） |
| color | cl（主要用于颜色值） |
| average | avg |
| background | bg（主要用于布局和子布局的背景） |
| selector | sel主要用于某一view多种状态，不仅包括Listview中的selector，还包括按钮的selector） |
| buffer | buf |
| control | ctrl |
| default | def |
| delete | del |
| document | doc |
| error | err |
| escape | esc |
| increment | inc |
| infomation | info |
| initial | init |
| image | img |
| Internationalization | I18N |
| length | len |
| library | lib |
| message | msg |
| password | pwd |
| position | pos |
| server | srv |
| string | str |
| temp | tmp |
| window | wnd(win) |