**包的命名规范**

建议采用如下规则：【com】.【公司名/组织名】.【项目名称】.【模块名】

包名：模块+功能

如

order

send

resive

常见的包分层结构如下：

* com.xxx.xxx.view –> 自定义view 或者是View接口
* com.xxx.xxx.activities –> activity类
* com.xxx.xxx.fragments –> fragment类
* com.xxx.xxx.adapter –> 适配器相关
* com.xxx.xxx.utils –> 公共工具类
* com.xxx.xxx.bean –> 实体类
* com.xxx.xxx.service –> service服务
* com.xxx.xxx.broadcast –> 广播接收器
* com.xxx.xxx.db –> 数据库操作类
* com.xxx.xxx.persenter –> 中间对象
* com.xxx.xxx.model –> 数据处理类

**类的命名规范**

Android中类的命名与JAVA开发采用一致的规范即可。

大驼峰命名法，即所有单词首字母大写。

* Activity –> xxxActivity.java
* Application –> xxxApplication.java
* Fragment –> xxxFragment.java
* Service –> xxxService.java
* BroadcastReceiver –> xxxBroReceiver.java
* ContentProvider –> xxxProvider.java
* Adapter –> xxxAdapter.java
* Handler –> xxxHandler.java
* 接口 –> xxxInter.java
* 接口实现类 –> xxxImpl.java
* Persenter –> xxxPersenter.java
* 公共父类 –> BaseActivity.java、BaseFragment.java、- BaseAdapter.java等
* util类 –> LogUtil.java
* 数据库类 –> BaseSQLiteDBHelper.java

### 变量方法的命名规范

采用驼峰命名规则。

Java普通变量：

* resultString
* userBean
* loginPresenter

Android控件变量：

* btnLogin
* pwdEtInput
* tvShowName

有些人建议采用【控件缩写】+【控件逻辑名称】的方式，比如btnLogin。

常用控件的缩写

| **控件** | **布局文件中缩写** | **代码中缩写** |
| --- | --- | --- |
| LinearLayout | xxx\_layout | xxxLLayout |
| RelativeLayout | xxx\_layout | xxxRLayout |
| FrameLayout | xxx\_layout | xxxFLayout |
| TextView | xxx\_tv | xxxTv |
| EditText | xxx\_et | xxxEt |
| Button | xxx\_btn | xxxBtn |
| ImageView | xxx\_iv | xxxIv |
| CheckBox | xxx\_chk | xxxChk |
| RadioButton | xxx\_rbtn | xxxRbtn |
| ProgressBar | xxx\_pbar | xxxPbar |
| ListView | xxx\_lv | xxxLv |
| WebView | xxx\_wv | xxxWv |
| GridView | xxx\_gv | xxxGv |

常见单词的缩写：

| **单词** | **缩写** |
| --- | --- |
| icon | ic |
| background | bg |
| foreground | fg |
| initial | init |
| information | info |
| success | succ |
| failure | fail |
| error | err |
| image | img |
| library | lib |
| message | msg |
| password | pwd |
| length | len |
| buffer | buf |
| position | pos |

### **常量命名：** 全部单词采用大写，每个单词之间用“\_”分割。

例如：

public static final String API\_URL = "http://apis.baidu.com/heweather/weather/free";

### 方法的命名规范

与java开发类似，采用驼峰命名规则。首单词首字母小写，其余单词首字母大写。尽量不要使用下划线。

举例：

* setxxx()
* getxxx()
* loginxxx()
* onCreate()
* onDestory()
* isxxx() –> 返回值是boolean类型
* checkxxx()

### 布局的命名规范

全部采用小写，单词之间使用下划线分割。

布局文件：

* activity\_login.xml
* fragment\_first\_tab.xml
* item\_choose\_city.xml
* dialog\_choose\_city.xml
* common\_footer.xml
* popup\_xxx.xml

控件ID：

上面【常用控件的缩写】表格中基本列出了常用控件的ID写法。

* login\_btn
* input\_phone\_et
* input\_pwd\_et
* login\_pbar

### drawable目录下的命名规范

全部单词小写，单词之间采用下划线分割。

* 图标 – > ic\_xxx.png –> ic\_logo.png
* 背景图 –> bg\_xxx.jpg –> bg\_splash.jpg
* selector –> selector\_login\_btn.xml
* shape –> shape\_login\_btn.xml
* 图片状态 –> bg\_login\_btn\_pressed.jpg & - bg\_login\_btn\_unpressed.jpg

### anim目录下的命名规范

单词全部小写，单词之间采用下划线分割。

* fade\_in.xml
* fade\_out.xml
* slide\_in\_from\_left.xml
* slide\_in\_from\_top.xml
* slide\_out\_to\_right.xml
* slide\_out\_to\_bottom.xml

## 编码规范

* 代码中尽量不要出现中文。注释和除外。代码中通过strings.xml引用来显示中文。
* 控件声明放在activity级别，这样在activity其他地方可以使用。
* 在一个View.OnClickListener中处理所有的点击事件逻辑，这样看起来很集中和直观。
* strings.xml中使用%1ssd等实现字符串的通配。
* 布局文件中的字体大小，都定义在dimens.xml中。
* 有关margin和padding的值也都放在dimens.xml中。
* 界面之间传值尽量使用intent方式。少用全局变量。
* 不建议在布局文件中添加点击事件。
* 数据类型转换一定要校验。
* 使用常量代替枚举。
* 实体不要在不同模块间共享，但是可以在统一模块下的不同页面共享。
* 建议采用左括号与方法名称在同一行的代码格式来进行代码的编写和格式化。貌似左括号在下一行是C#的形式。
* 业务稍微复杂一些，都有可能提炼一个BaseActivity或BaseFragment出来做为公共父类。
* 类注释一定要写，管家的方法也要写方法注释。常量尽量写注释。