编码规范

**一、命名规范：**

1. **包名**

包名全部小写，连续单词只是简单的连接起来，不使用下划线。

1. **类名**

类名均以UpperCamelCase风格编写，大驼峰命名法，如WelcomeActivity。尽量避免缩写，除非该缩写是众所周知的，比如HTML，URL，如果类名称中包含单词缩写，则每个单词都应该要大写。

测试类的命名以它要测试的类的名称开始，以Test结束，如HashTest.

接口：命名规则与类一样采用大驼峰命名法，多以able或ible结尾。

1. **方法名**

以lowerCamelCase风格编写，如initView();

1. **常量名**

常量名命名模式为CONSTANT\_CASE，全部字母大写，用下划线分割单词。

1. **非常量字段名**

以lowerCamelCase风格的基础改造为如下风格：基本结构为scopeVariableNameType.

**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，共有静态字段(全局变量)命名以g开头。如：int mPackagePrivate。

1. **参数名**

参数名以lowerCamelCase风格编写。参数名应该避免用单个字符命名。

1. **局部变量名**

局部变量名以lowerCamelCase风格编写，尽量避免使用单字符进行命名，除了临时变量和循环变量。

1. **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一般用于字符型。

1. **类型变量名**

命名方式①、单个的大写字母，后面可以跟一个数字（如：E,T,X,T2）；②、以类命名方式，后面加个大写的T（RequestT）。

二**、资源文件规范**

1. **资源布局文件（XML文件（layout布局文件））**

全部小写，采用下划线命名法

1. **contentview命名**

必须以全部单词小写，单词间以下划线分割，使用名词或使用名词词组。所有Activity或Fragment的contentView必须与其类名对应，对应规则为：

将所有字母都转为小写，将类型和功能调换（也就是后缀变前缀）。如：activity\_main.xml

1. **Dialog命名**

dialog\_描述.xml,如：dialog\_hint.xml

1. **PopupWindow 命名**

ppw\_描述.xml ,如：ppw\_info\_xml

1. **列表项命名**

item\_描述.xml,如：item\_city.xml

1. **包含项命名**

模块\_(位置)描述.xml，如：activity\_main\_head.xml

**2、资源文件（图片drawable文件夹下）**

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，逻辑名称可由多个单词加下划线组成，采用以下规则：

用途\_模块名\_逻辑名称  
用途\_模块名\_颜色  
用途\_逻辑名称  
用途\_颜色

注：用途也指控件类型

1. **动画文件（anim文件夹下）**

全部小写，采用下划线命名法，加前缀区分。具体动画采用以下规则：模块名\_逻辑名称。如：fade\_in ----淡入

1. **values中name命名**

##### **1) strings.xml**

strings的name命名使用下划线命名法，采用以下规则：模块名+逻辑名称

#### **2)layout中的id命名**

命名模式为：view缩写\_模块名\_逻辑名，比如btn\_main\_search  
使用 AndroidStudio 的插件 ButterKnife Zelezny，生成注解非常方便，或者也可以使用Android Code Generator插件。

**三、注释规范**

#### **1. 类注释**

每个类完成后应该有作者姓名和联系方式的注释，对自己的代码负责。

#### **2. 方法注释**

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，

#### **3. 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。