

# 一个优秀的Android应用从建项目开始

筒 by Jude95, jianshu.com

---

## 1.项目结构

现在的MVP模式越来越流行。就默认采用了。

如果项目比较小的话：

- app——Application Activity Fragment Presenter等的顶级父类
- config——API,常量表等
- model——数据层
  - bean——数据模型
- presenter——MVP的P
- view——MVP的V
- utils——工具类集合
- widget——各个可复用View集合

如果项目比较大，上面的方式一定会造成presenter和view里近百个文件。看瞎眼系列。推荐下列方式：

- app
- config
- model
  - bean
- module——将界面层以功能模块分配包。
  - launch
  - main
  - account
  - news
  - music
  - .....
- utils

- widget

## 2.配置主题

对于不遵守Material Design的项目无视这一步。

### 1.先在color.xml中写好需要的颜色:

```
<resources>
    <color name="Orange">#ff5722</color>
    <color name="DeepPurple">#673AB7</color>
    <color name="DeepPurple900">#311B92</color>
    <color name="White">#fff</color>
    <color name="Gray">#888888</color>
    <color name="Gray100">#dddddd</color>
    <color name="Gray600">#999999</color>
</resources>
```

注意color.xml是配色表。应该是描述颜色而不是对字体颜色，背景颜色等的定义。这样能防止相近的颜色重复定义。而导致界面颜色不统一。

### 2.在style.xml里定义主题:

```
<style name="AppTheme.Base" parent="Theme.AppCompat.Light.NoActionBar">
    <!-- Customize your theme here. -->
    <item name="colorPrimary">@color/DeepPurple</item>
    <item name="colorPrimaryDark">@color/DeepPurple900</item>
    <item name="colorAccent">@color/Orange</item>
</style>

<style name="AppTheme" parent="AppTheme.Base"></style>
```

在res目录下，创建一个values-v21目录，再创建一个style.xml:

```
<style name="AppTheme" parent="AppTheme.Base">
    <item name="android:windowDrawsSystemBarBackgrounds">true</item>
    <item name="android:statusBarColor">?colorPrimaryDark</item>
</style>
```

然后在AndroidManifest.xml文件中修改application的theme属性为上面定义的AppTheme.即可实现沉浸式状态栏。

然后关于Theme与Toolbar的详细设置参考我另两篇博客：

<http://www.cnblogs.com/Jude95/p/4369816.html>

<http://www.cnblogs.com/Jude95/p/4370176.html>

### 3.依赖库与SDK

必选的库：

gradle-retrolambda——Android的lambda表达式插件

fresco——Android最屌图片加载库

material-dialogs ——Material Dialog向下兼容库

material-ripple——Ripple向下兼容库

fastjson——最快JSON解析

butterknife——View注解库和配套插件android-butterknife-zelezny

ActiveAndroid——数据库注解库。

compile 'com.android.support:design:22.2.0'——谷歌Material Design控件库

下面安利几个自己写的库，如果有什么建议欢迎交流：

Utils——Android各种小功能集合

RollViewPager——自动轮播使用方便的ViewPager

EasyRecyclerView——支持下拉上拉刷新等功能全面的RecyclerView

RequestVolley——仅仅是让Volley方便一点

尝试了很多，这几个是现在常用的。

融云——即时通讯

友盟——数据统计,推送,意见反馈,自动更新,第三方分享及登录,社区

七牛——云存储

Mob——短信验证

Bmob——做后台不求人

依赖这一大堆库和SDK以后。建议在合适的时机初始化他们，而不是全堆在Application的onCreate()里面。这样会导致启动时间过长。启动后也会较卡。虽然是不会影响功能正常使用。

### 4.配置Gradle

某些SDK运行时需要检查签名是否正确。所以在debug模式时也必须用正式KEY签名。而把

签名放进版本控制不是明智的做法。所以推荐下面的做法：  
在app的gradle加入下面代码

```
Properties props = new Properties()
props.load(new FileInputStream(file("signing.properties")))
android {
    signingConfigs {
        release{
            keyAlias props['KEY_ALIAS']
            keyPassword props['KEY_PASSWORD']
            storeFile file(props['KEYSTORE_FILE'])
            storePassword props['KEYSTORE_PASSWORD']
        }
    }
    buildTypes {
        release {
            signingConfig signingConfigs.release
        }
        debug {
            signingConfig signingConfigs.release
        }
    }
}
```

在app的gradle文件同级目录新建signing.properties文件，里面填入你的key的相应信息

```
KEYSTORE_FILE = C:\\Users\\Mr.Jude\\Documents\\Android\\HelloWorld.jks
KEYSTORE_PASSWORD = xxxxxx
KEY_ALIAS = xxxxxx
KEY_PASSWORD = xxxxxx
```

将signing.properties添加进忽略目录。  
其他人pull下来代码后。自己新建signing.properties填入相应信息后即可编译成功。

## 5.制定开发规范

为了避免合作开发写的代码风格迥异。或做出了多套开发模式。下面是个例子。毕竟是为了高效开发而制定的。适合自己项目的才是最好。

所有Activity继承BaseActivity

所有Fragment继承BaseFragment

所有Presenter继承BasePresenter

这样利于生命周期管理。也可以方便的全局修改。

命名，例

AccountFragment

UserDetailActivity

layout命名，例

activity\_collection

fragment\_account

item\_person

include\_toolbar

view\_progress

不过对于庞大项目的开发。近百个activity开头的layout列表还是会眼瞎。所以那种情况会在前面加上模块名。

id命名，例

btn\_send

tv\_name

list\_persons

et\_password

然后用butterknife的插件生成变量会自动将下划线变成驼峰命名

变量命名:以m开头。例mAdapter使用时按一个m全都出来了

方法命名:与其写好名字不如写好注释。==。

**TextView**使用官方标准字体

Display 4	Regular 112sp
Display 3	Regular 56sp
Display 2	Regular 45sp
Display 1	Regular 34sp
Headline	Regular 24sp
Title	<b>Bold 21sp</b>
Subhead	Regular 17sp (Device), Regular 16sp (Desktop)
Body 2	<b>Bold 15sp (Device), Bold 14sp (Desktop)</b>
Body 1	Regular 15sp (Device), Bold 14sp(Desktop)
Caption	Regular 13sp
Button	<b>Bold 15sp</b>

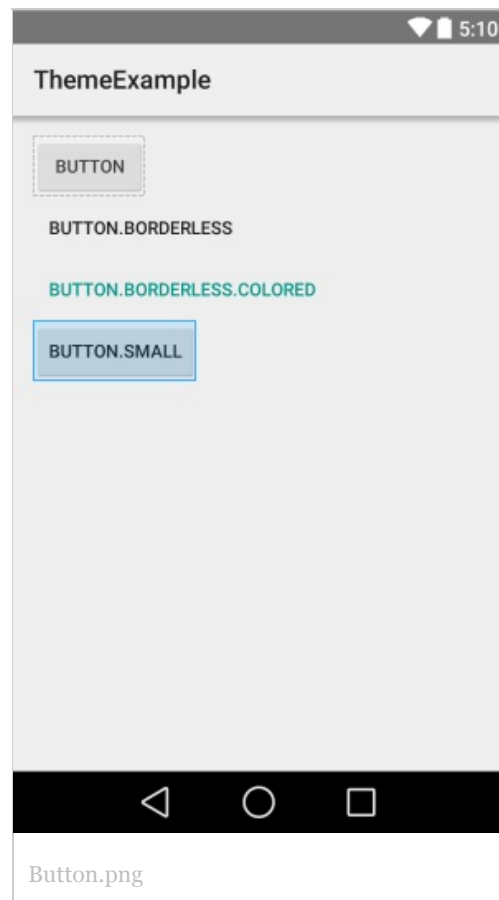
TextView.png

```

style="@style/TextAppearance.AppCompat.Display4"
style="@style/TextAppearance.AppCompat.Display3"
style="@style/TextAppearance.AppCompat.Display2"
style="@style/TextAppearance.AppCompat.Display1"
style="@style/TextAppearance.AppCompat.Headline"
style="@style/TextAppearance.AppCompat.Title"
style="@style/TextAppearance.AppCompat.Subhead"
style="@style/TextAppearance.AppCompat.Body2"
style="@style/TextAppearance.AppCompat.Body1"
style="@style/TextAppearance.AppCompat.Caption"
style="@style/TextAppearance.AppCompat.Button"

```

**Button**使用**Material Design**标准样式



```
style="@style/Widget.AppCompat.Button"  
style="@style/Widget.AppCompat.Button.Borderless"  
style="@style/Widget.AppCompat.Button.Borderless.Colored"  
style="@style/Widget.AppCompat.Button.Small"
```

定好网络请求写法。文件存储方式与位置。写好项目所使用的类库框架用法。

好了，下面就开始正式开发吧！如果有什么建议欢迎交流。本文也会即时修改。