Android开发规范

## 一、命名规范

首先所有的命名应该是有意义的,不管是代码、drawable资源、string资源、color资源等，命名都按照 “模块功能\_属性”来命名，

比如: 车辆列表模块

CarListAct.java Activity命名

CarListFrag.java Fragment命名

CarListAdapter.java Adapter命名

CarListItemView.java 自定义View

car\_list\_frag.xml Fragment对应的布局

car\_list\_topbanner.png 图片命名

car\_list\_item.xml 列表item

btn\_car\_check\_selector.xml drawable下的selector文件

shape\_car\_list\_topbg.xml drawable下的shape文件

## 二、统一代码风格

为了统一code style ,请大家按如下步骤配置eclipse:

1.CodeTemplates.xml //去掉eclipse默认生成的//TODO 之类注释

Window->preferences->Java->CodeStyle->Code Templates->Import

2.CodeFormatter.xml //按我们的代码规范定义的格式化文档

Window->preferences->Java->CodeStyle->Formatter->Import

3.请常用ctrl+shift+f格式化代码

4.java->editor->save actions->保存时自动格式化，方便使用.

5.如有修改CodeFormatter.xml,Codetemplates.xml 请提交SVN,并周知大家更换。

SVN地址：技术\Android\文档\Eclipse配置

## 三、UI开发规范

### 1、默认布局

UI界面调整好后，尽量把UI控件预设内容置空,避免无数据时造成的显示错误。

### 2、资源管理

开发完毕后，删除无用的资源文件(图片资源、layout资源等)和代码，避免无用资源增加APK体积。

备注：针对版本升级功能开发，在删除老版本资源时，应当小心，看看是否还有其他模块也在引用该资源。

### 3、资源文件

共用资源应该写到对应资源文件中，比如string.xml，color.xml，dimen.xml等

### png图片压缩

对png图片进行无损压缩，一方面可以减少app内存占用，避免出现OOM，也可以减少APK大小。

所以请先将png图片进行无损压缩，然后放置到drawable对应目录下，推荐大家使用：<https://tinypng.com/> 网站进行png图片的压缩。

## 四、代码开发规范

### 1、首先熟悉业务逻辑

在开发前，首先熟悉开发内容，把存在疑惑的地方提出来，与需求提出方进行讨论，确保大家的理解是一致的，避免只按照自己的理解去做。

### 2、制定接口

首先和接口开发人员一起讨论制定接口，接口开发完毕后，客户端人员应该优先测试接口，如果接口需要调整或有Bug，尽快通知接口开发人员。

备注：千万不要等到UI做完后才去测试接口，这样会拉长整体开发时间。应该让我们的开发时间处于并行状态。

### 3、功能开发

设计相应的Java类，进行Java类的职责划分，谁负责UI渲染，谁负责加载数据，谁负责逻辑处理等.

温馨提示：写代码不要着急，只有将交互逻辑梳理通了，才能写的流畅。

### 4、模块优化

开发完毕后，首先自测一下，看是否有Bug，然后梳理一下逻辑，优化一下模块代码结构。优化方面大概有：

**Layout布局优化**

布局尽量简单，避免层级过深，层级过深除了导致加载xml缓慢外，还容易导致出现StackOverFlow 栈溢出。

**代码优化**

在移动设备上应当避免不必要的对象分配，将重复冗余的代码进行简单封装(一次编写，多处使用)，优化代码逻辑，代码逻辑应该简单清楚。

代码中应避免出现编译警告信息。

**页面加载速度优化**

如果页面的布局比较复杂或者层级结构嵌套较深，则View渲染速度比较缓慢，导致的一个直接现象就是用户进入该页面

的等待时间较长，所以应该采用延迟加载的方式，让用户先进入页面，然后再渲染耗时的View。

**数据库优化**

如果要处理批量数据，应该使用Sqlite提供的transaction机制。

**避免内存泄露**

如果项目中存在大量的内存泄露，则用户使用一段时间后，会出现卡顿现象，严重时会造成OOM。

内存泄露一般都是由于异步线程中引用短生命周期对象，当短生命周期对象被系统回收时，由于线程还持有该对象的引用，造成该对象内存无法回收，因此造成该对象的内存泄露，常见的由系统管理的生命周期对象：Activity、View等。如果是Activity的内存泄露，而且该Activity中持有大量图片引用，则后果会很严重。

### 代码审核

功能开发完毕后，负责人需要对模块功能代码进行审核。审核的目的是减少犯错的可能。

## 五、自动化测试

### 1、Monkey压力测试

Monkey模拟点击，在一定时间内完成模拟点击次数

常用monkey命令可百度搜索一下：

adb shell monkey -p 应用包名 -v 模拟点击次数

比如:

adb shell monkey -p ccom.bbchexian.android -v 1000

打开BB车险APP，模拟点击1000次

### 第三方自动化测试平台

常用的第三方自动化测试平台有：Testin云测试平台、腾讯优测

自动化测试平台主要针对安装、卸载、屏幕分辨率适配、各系统版本兼容适配、

各手机厂商系统兼容适配等。

### APM性能测试

目前APM性能测试这块做的比较好的有听云、博睿APM。

性能测试主要针对：用户使用过程中的崩溃、连接超时、交互卡顿等现象.

## 六、APK打包

### 1、打包前检查

**版本号检查**

AndroidManifest.xml文件中检查versionCode和versionName是否正确.

versionCode和versionName是有对应关系的。比如

|  |
| --- |
| android:versionCode=*"11000"*  android:versionName=*"1.1.0"* |

versionCode：升级时用于判断版本更新

versionName：显示给用户的版本号

对应关系举例：

|  |
| --- |
| versionCode=12000 则versonName=1.2.0  versionCode=12300 则versonName=1.23.0  versionCode=12304 则versonName=1.23.04  versionCode=12340 则versonName=1.23.40  versionCode=12345 则versonName=1.23.45 |

**服务器环境检查**

检查assets/config.properties配置文件中的debug模式是否已关闭，并且服务器环境为线上环境。

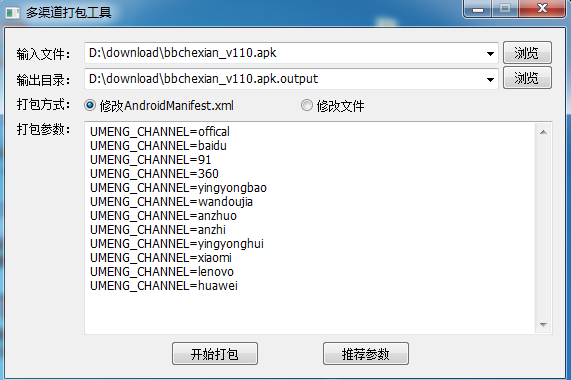


### 2、APK打包加固

为了防止别人反编译APK，目前采用的是腾讯的APK加固。

根据咱们的签名文件进行打包。然后利用腾讯提供的多渠道打包工具进行多渠道打包(apk加固)，打包完毕后，再利用提供的签名工具进行重签名。

加固步骤配置如下



1.、输入文件选择要要加固的apk文件。

2、打包方式选择“修改AndroidManifest.xml”。

3、打包参数

按照UMENG\_CHANNEL=渠道码 的格式添加渠道配置

## 七、APK发布

### 1、APK发布前，应该进行简单验证

比如服务器环境是否为线上环境、新增功能检验。

### 2、将发布的APK备份到SVN，作为存档。

SVN地址：技术\Android\线上包

### 3、代码版本管理。

版本发布后，代码备份到tags目录.

