**TAC组Android代码规范**

Java通用编码规范：

1. **包：**

采用反域名命名规则，本组采用com.tac.<所属功能区域>.<功能模组>.<内部子功能>

命名

例如com.tac.UI.GiveJobs

**2. 类：**

名词，采用大驼峰命名法，且不允许自定义缩写，除非是HTML、URL等已经约定的缩写，而如果包含了单词缩写，则该缩写所有字母大写。

例如：UnvalidJobContent

**3. 接口：**

同样与类一样采用大驼峰命名法，但是尽量使用able或者ible结尾

例如：Runnable

**4. 方法：**

采用小驼峰命名法，即第一个单词首字母小写，而其余所有单词首字母大写。

例如：getValidJobContent

**5. 变量：**

采用小驼峰命名法，特别的，如果该变量对应布局.中的某个控件，则其名字必须与该控件的id一致。

局部变量名、静态成员变量名：只能包含字母，大驼峰命名法。

常量：全部大写，并且在不同单词间加入\_

前端独有编码规范：

**1． 资源文件：**

全部小写，在不同单词间加入\_，加入前缀区：控件名\_逻辑名称

**2. 资源布局文件：**

包括layout布局文件

全部小写，下划线命名法

例如：app\_main.xml

**3. 资源ID命名：**

包括string.xml,style.xml等

activity名称\_功能模块名称\_逻辑名称

activity名称\_逻辑名称 common\_逻辑名称

**4. layout中的id命名法：**

view缩写\_模块名称\_view的逻辑名称

例如：tv\_job\_detail、btn\_accpet\_worker

**TAC组编码原则**

1. **网络请求优化**

http用gzip压缩，设置连接超时时间和响应超时时间

http请求按照业务需求，分为是否可以缓存和不可缓存，那么在无网络的环境中，仍然通过缓存的httpresponse浏览部分数据，实现离线阅读。

2. **listview 性能优化**

1).复用convertView

在getItemView中，判断convertView是否为空，如果不为空，可复用。如果couvertview中的view需要添加listerner，代码一定要在if(convertView==null){}之外。

2).异步加载图片

item中如果包含有webimage，那么最好异步加载

3).快速滑动时不显示图片

当快速滑动列表时（SCROLL\_STATE\_FLING），item中的图片或获取需要消耗资源的view，可以不显示出来；而处于其他两种状态（SCROLL\_STATE\_IDLE 和SCROLL\_STATE\_TOUCH\_SCROLL），则将那些view显示出来

3. **避免阻塞主线程**

使用线程池，分为核心线程池和普通线程池，下载图片等耗时任务放置在普通线程池，避免耗时任务阻塞线程池后，导致所有异步任务都必须等待

4. **多线程注意事项**

异步任务，分为核心任务和普通任务，只有核心任务中出现的系统级错误才会报错，异步任务的UI操作需要判断原activity是否处于激活状态

**5. 资源耗用控制**

尽量避免static成员变量引用资源耗费过多的实例,比如Context

**8. UI布局优化**

(1) layout组件化，尽量使用merge及include复用

(2) 使用styles，复用样式定义

(3) 复杂布局使用RelativeLayout

(4) 自适应屏幕，使用dp替代pix

(5) 使用android:layout\_weight或者TableLayout制作等分布局