[编程规范 2](#_Toc53941314)

[注释 2](#_Toc53941315)

[类 2](#_Toc53941316)

[实例成员 2](#_Toc53941317)

[函数 3](#_Toc53941318)

[语句 4](#_Toc53941319)

[命名规范 5](#_Toc53941320)

[缩写 5](#_Toc53941321)

[另一些事情 6](#_Toc53941322)

[界面原型 7](#_Toc53941323)

[主界面布局 7](#_Toc53941324)

[Fragment 首页 8](#_Toc53941325)

[Fragment 服务 9](#_Toc53941326)

[Fragment 客服 10](#_Toc53941327)

[Fragment 视频 11](#_Toc53941328)

[Fragment 我的 12](#_Toc53941329)

[Activity 我要缴费 13](#_Toc53941330)

[Activity 账单详情 14](#_Toc53941331)

[Activity 远传检测 15](#_Toc53941332)

[Activity 我的水号 16](#_Toc53941333)

[Activity 编辑信息 17](#_Toc53941334)

[模块划分 18](#_Toc53941335)

[com.dev.water\_query 18](#_Toc53941336)

[类说明 20](#_Toc53941337)

# 编程规范

## 注释

### 类

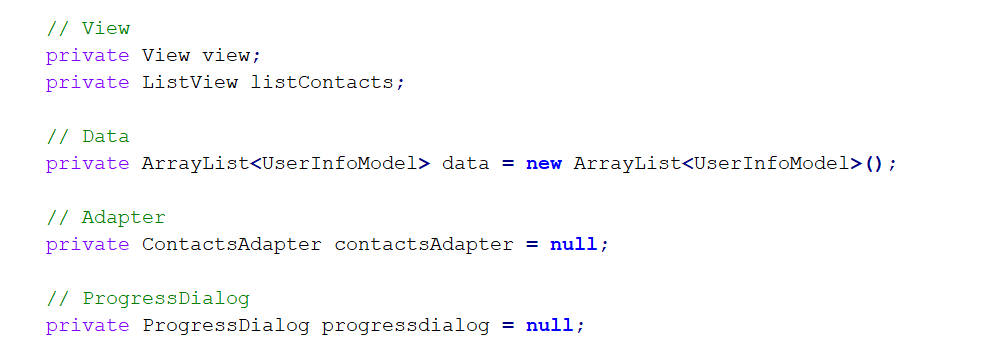
使用注释模板自动生成注释

|  |
| --- |
| 1. /\*\* 2. \* @Title: 标题 3. \* @Package 包名 4. \* @Description: 简述该类的作用 5. \* @author: 上次修改的作者 6. \* @date: 上次修改的时间和日期 7. \* @version 版本 8. \*/ 9. /\*\* 10. \* @Title: NetUtils 11. \* @Package com.qlnu.waterquery.net 12. \* @Description: 网络工具类，封装了网络请求 13. \* @author: Juston 14. \* @date: 2020-10-18 9:29 15. \* @version v1.2 16. \*/ |

### 实例成员

按照不同的功能区分开来 不同功能的实例成员中间用空行隔开并且用注释表明功能

实例成员应当放在类的前面 常量也应该放在前面



### 函数

使用注释模板自动生成注释 构造函数也是如此

|  |
| --- |
| 1. /\*\* 2. \* @Title: 标题 3. \* @Description: 简述该函数的作用 4. \* @param: 参数 5. \* @return: 返回值 6. \* @date: 上次修改的时间和日期 7. \* @throws 可能会抛出的异常 8. \*/ 9. /\*\* 10. \* @Title: getUserInfoById 11. \* @Description: 通过id获取用户信息 12. \* @param: id 唯一标识 13. \* @return: UserInfoModel 用户信息实体类 14. \* @date: 2020-10-1 12:32 15. \* @throws UserNotFoundException   \*/ |

### 语句



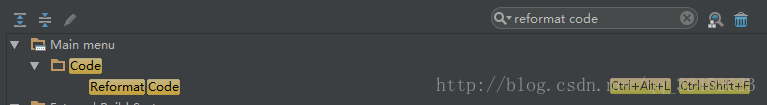
为了完成同一操作的代码应该放到一起 不同操作的代码之间应该用空行隔开

除非意图很明显的操作否则都应该加上一个简单的注释

如果有比较复杂的操作，应该详细解释代码是如何运行的

还有一点 就是缩进

使用系统自带的代码格式整理快捷键整理



## 命名规范

**命名尽可能的使用描述性的，少用意义不明的缩写**

|  |  |  |
| --- | --- | --- |
| **用处** | **用法** | **示例** |
| **类** | 大驼峰 | UpperCameCase |
| **常量** | 全部大写，下划线分开 | MAX\_SPEED |
| **函数** | 小驼峰 | getUserInfoById |
| **参数** | 小驼峰 | dateTime |
| **变量** | 小驼峰 私有变量前用m 静态变量前用s | mBtnLogin sPersonInfo |
| **资源id和资源文件** | 模块+功能名称 单词间下划线分开 都采用小写 | btn\_login |

mImgLogin 实例成员图片控件登录 要这样

缩写

|  |  |  |
| --- | --- | --- |
| 缩写 | 通用 | 意义 |
| btn | Button | 按钮 |
| img | Image | 图片 |
| str | String | 字符串 |
| init |  | 初始化 |
| bmp |  | 图片 |
| buf | Buffer | 缓冲区 |

待补充

## 另一些事情

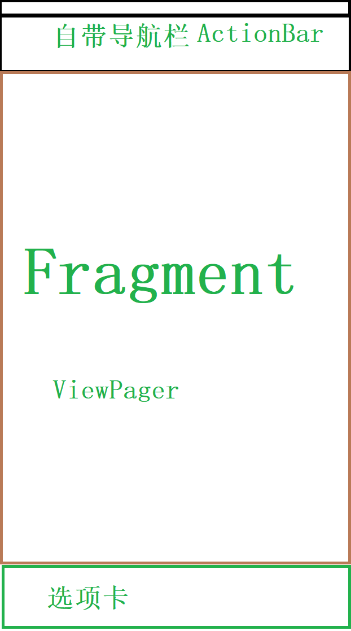
不能直接在活动里面new线程做耗时操作（读写文件数据库、网络请求），而是应该交给service去做。

OnCreate只做一些简单的事情，不做耗时操作。

初始化应该交给onStart，然后在onStop释放。

# 界面原型

## 主界面布局



## Fragment 首页

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 效果图 | 界面说明 | 功能说明 |
| 效果 |  | 整体由三个部分组成。服务列表、轮播图、通知 |  |
| 服务列表 |  | 由GridView实现。  每项格式如下 | 点击项可以转到相应的活动 |
| 轮播图 |  | 由ViewPager嵌套Fragment实现 | 可以左右滑动。点击相应图片可以转到网页或活动 |
| 通知 |  | 顶部为选项卡。  为一个Fragment。其中上方为listview，“查看更多”为textView | 左右滑动或者点击“选项卡”可以切换不同类别的通知。  点击通知进入相应页面。  点击查看更多进入新页面显示所有通知 |

## Fragment 服务

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 效果图 | 界面说明 | 功能说明 |
| 效果 |  | 是一个ListView。具体每项组成请参考下面说明。 | 可以上下滑动。 |
| ListView |  | Listview每一项由两部分组成。上方的文字以及图标、下方的GridView  GridView格式如下 | 点击项可以转到相应的页面或活动 |

## Fragment 客服

暂无

## Fragment 视频

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 效果图 | 界面说明 | 功能说明 |
| 效果 |  | 由上方的选项卡和下方的Fragment组成。 | 点击选项卡，转到对应的Fragment。每个Fragment显示不同类别视频。 |
| ListView的项 |  | 由标题、图片以及下方时间组成。 | 点击项转到对应页面或网页 |

## Fragment 我的

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 效果图 | 界面说明 | 功能说明 |
| 效果 |  | 由互相独立的控件组成。  分为有头像选项和文字选项 | 显示常用的功能。 |
| 头像 |  | 左边显示头像，中间文字，最右边显示小三角 | 点击后转入编辑信息界面 |
| 文字 |  | 左边显示竖杠，中间文字，最右边显示小三角 | 点击后调用相应功能 |

## Activity 我要缴费

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 效果图 | 界面说明 | 功能说明 |
| 效果 |  | 这个就认为是一个单独的，只有一个给水号 | 进入该页面后（onResume）  弹出请求对话框，告诉用户正在请求信息。请求成功把信息显示在对应位置。 |
|  |  |  | 如果金额为负数，那么显示为红色  并且下方显示已欠费xxx元 |
|  |  |  |  |

## Activity 账单详情

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 效果图 | 界面说明 | 功能说明 |
| 效果 |  |  |  |
|  |  |  |  |
|  |  | 用网络上的chart包 |  |
|  |  |  |  |

## Activity 远传检测

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 效果图 | 界面说明 | 功能说明 |
| 效果 |  |  |  |
|  |  |  |  |
|  |  |  | 这个表格还不太确定  做不出那个效果来 |

## Activity 我的水号

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 效果图 | 界面说明 | 功能说明 |
| 效果 |  |  |  |
|  |  |  | 从网络上请求信息并显示。点击按钮转到相应页面 |
|  |  | 这个我也很纳闷 难道不久应该只绑定一个嘛？ |  |

## Activity 编辑信息

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 效果图 | 界面说明 | 功能说明 |
| 效果 |  |  |  |
|  |  |  | 点击后从下方弹出对话框  “拍照”、“相册”、“取消” |
|  |  |  | 点击后弹出输入文字的对话框 |
|  |  |  | 如果没有绑定那么按钮文字显示为绑定，昵称控件什么都不显示 |
|  |  |  | 点击修改信息后，用户修改过的手机号和昵称信息才会被上传到服务器进行更改操作 |

# 模块划分

|  |  |  |
| --- | --- | --- |
| 模块 | 功能 |  |
| activity | 存放活动类 | WelcomeActivity.java  登录  BaseActivity.java  EditUserInfoActivity.java  WaterAccountActivity.java  WaterUseSituationActivity.java  PayFeesActivity.java  RemoteDetectionActivity.java  BillDetailsActivity.java |
| fragment | 存放Fragment类 | HomeFragment.java  ServiceFragment.java  OnlineCustomerServiceFragment.java  VideosFragment.java  PersonalFragment.java |
| entity | 存放实体类 | HalfYearStatisticsEntitiy.java  PayInfoEntity.java  RecordDataEntity.java |
| network | 存放网络相关类 | 与网络相关操作全都封装在这里。 活动只负责调用传参和接受数据 |
| utils | 存放工具类 |  |
| adapters | 存放适配器类 | 用的的适配器都放在这里 |
| services | 存放服务类 | WaterServices.java 注意：网络应该把请求操作提交给服务去执行！ |
| events | 存放事件（回调函数）类 | 待定 回调类应该放在对应模块中 比如网络的请求完毕模块 |

# com.dev.water\_query

模块划分情况

# 类说明