通用库说明及使用

通用库是为了提高工作效率，统一风格，方便代码迁移而产生的公司宝贵的财富，同时你能学习到别人的优点，也能和别人切磋产生思维的碰撞，本文将介绍如何对使用通用库提高自己的工作效率，以及如何维护和新增模块。

通用库：包括网络框架、图片缓存、版本更新、工具类、常用基类、数据库管理、自定义控件、上拉下拉列表、下拉刷新、及一些实用方法（日期选择、地区选择、地图、照片选择等一些常用的方法）

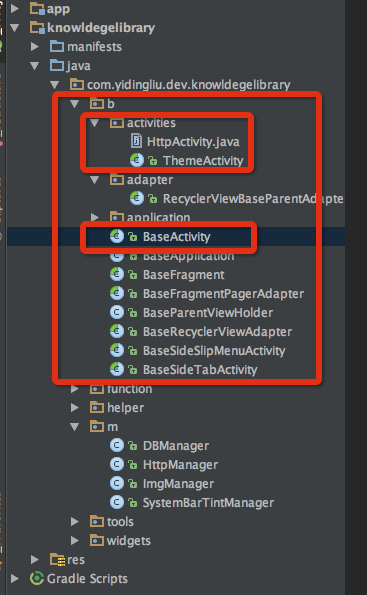
暴露的接口及传入参数应当越少越好，越简单且无Bug为上佳。

1. 规则：

所有代码布局、命名均以Android移动开发规范执行。

框架结构：功能模块应当按照分类存放(基类及通用的类与接口除外)；基类：应当在当前包名下存放直接使用的基类，与该基类相关的类应当在当前包下单独新建包存放

例子：网络模块，所有与网络相关的应该存放在一起



1. 说明及使用：
2. 网络框架
3. 说明：

http使用okhttp，其他使用Socket

1. 使用方法：
2. 图片缓存
3. 说明：

采用imageLoader加载图片

~~采用常用的三级缓存模式：本地->内存->网络~~

1. ~~存取规则~~

~~存：查看本地是否已经缓存，如未缓存执行存储~~

~~取：查看内存是否已经缓存，如未缓存，再查看本地是否已经缓存，如未缓存，再通过网络取图片。有则反之。~~

1. ~~本地缓存：~~

~~缓存文件储存于数据库、sharedpreferences及SD卡中。~~

~~说明：~~

~~（1）所有配置信息均放于sharedpreferences中。~~

~~（2）图片均缓存在本地SD卡中。~~

~~（3）除了图片与配置信息都存放于数据库中~~

1. ~~内存缓存：~~

~~使用lrucache，最大可使用内存的8分之1做为缓存，~~

1. ~~网 络：~~

~~服务器正常取图片。~~

1. 使用方法：
2. 版本更新
3. 说明

使用蒲公英更新

~~异步更新，检测服务器是否有新的版本，如有，进行弹框提醒，或在WIFI网络下自动下载，下载完成之后，提醒用户安装更新。~~

1. 使用方法：

~~此功能为Service服务，只需传入地址、logo、标题即可完成更新。~~

1. 工具类
2. 说明：

包含所有的通用工具，此类全部为公有静态方法

1. 使用方法：
2. 基类
3. 说明：

基类应当以暴露抽象方法少，提供方法多实用性强为标准。与子类有关联的变量应当以public或者protected修饰，若抽象方法返回的值与子类相关联，应当使用泛型增加其便易性。

1. 数据库管理

说明：使用第三方数据库，郭霖的开源数据库LitePal,该数据库框架能够实现对象的存储，使用比较方便。

1. 下拉刷新，下拉加载列表
2. 下拉刷新控件
3. 自定义控件
4. 实用方法