1. 目的 2

2. 适用范围 2

3. 编码规范 2

3.1. 文件: 2

3.2. 注释 3

3.3. 编码排版格式 3

3.4. 命名规范 5

4. 功能类的定义 6

4.1. Service类 6

4.2. ConfigData类 6

4.3. DBOPerata类 6

4.4. GlobalConfig类 7

4.5. SecondLevelView，SecondLevelViewController 类 7

WARNING：

* 如果没有特殊情况，做页面布局的时候请不要使用Storyboard和Xib，请使用代码进行布局。
* 目前暂不鼓励使用Swift语言来进行项目的开发。

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | Auther | 日期 | 修订内容 |
| 1.0 | 朱檑 | 2013．09.08 | 编码规范编写 |
| 1.1 | 朱檑 | 2015.06.02 | 添加功能类的定义 |

# 目的

统一规范Xcode开发环境下Objective-C的编码风格和标准

# 适用范围

Objective-C语言开发的项目

# 编码规范

## 文件:

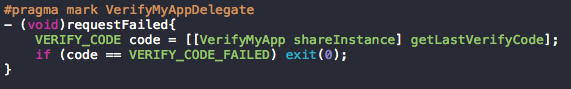
1. 项目命名以实际开发的项目名称来命名.类名不用加前缀.不要使用中文或者汉语拼音.
2. 公共的文件统一叫GlobalConfig.h .项目中的公用常量都在此类中定义 .
3. 所有相同等级的页面.都需要继承一个基础类.相同的代码都需要在基础类中完成.
4. 图片统一都放在Supporting Files目录下.需要在该文件夹下新建一个image文件夹(这个文件夹需要在外部建立,然后脱进项目中来image文件夹真实存在,)
5. 所有的第三方类库都是用3rdLibs文件夹,然后第三方类库直接放在3rdLibs文件夹下.(所有文件夹要真是存在,不能只是虚拟目录,方便文件源码文件归类).

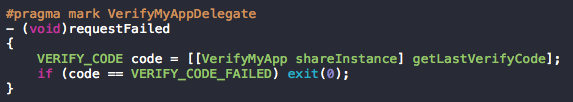
## 注释

1. 新建的一个类文件.需要在该类的头文件中注释下,介绍类的功能作用.方便其他开发人员快读了解该类.
2. 在头文件中对于暴露给外部调用的函数,需要加上注释说明.
3. 对于一些类的代理方法实现 . 一定要在代理函数上面加上 #pragma mark XXXXXDelegate 或者 #pragma mark XXXXXDatasource 方便其他编程人员通过索引快度查找.

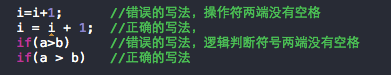
## 编码排版格式

1. 编码排版遵循美观大方得体的要求,在xcode中多使用control + I 来对所选代码进行自动排版.
2. 类中的函数括弧的每一个括弧都需要在单独占一行.

不要使用这种风格,建议使用下面风格



1. 空格的使用
   1. 关键词与其后的表达式之间要有空格,
   2. 操作符号后也需要有空格隔开



* 1. 函数前面的-与(void)或者(returnType)之间需要有个一个空格(如上图函数)

1. 每行建议每行只要一个语句,不要连续一行申明N个变量.

不建议使用:

NSUInteger objectIndex, stuffCount;

建议使用:

NSUInteger objectIndex;

NSUInteger stuffCount;

## 命名规范

1. 一切变量,函数,类的命名一定要已可读性为主.
2. 不要使用Ojective-c语言保留的字或者关键词.特别需要注意的是id类型.
3. 命名带上类型.比如NSArrar \* xxxArr . NSDictionary \* xxxDic ,UIView \* xxxView, UIViewController \* xxxViewCtontroller.
4. 函数名第一个字母小写，后面单词首字母大写；如果有参数，第一个参数要从函数名称上携带出来，myFunctionWithXxx；第二个参数的首字母小写；完整示例如：- (void)myFunctionWithSizeA:(CGSize)sizeA sizeB:(CGSize)sizeB{}
5. 变量名第一个字母小写，后面单词首字母大写，如：myVariable常量名全大写，如：MYCONSTANT
6. 定义全局变量不要使用下划线开头.例如NSString \* \_userName.不建议使用.建议使用NSString \* userName.

# 功能类的定义

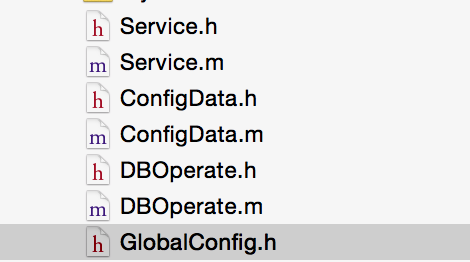


图1

## Service类

网络请求类。网络的请求创建，统一在该类中处理，在头文件中暴露接口供Contoller类调用，使用block方式返回请求后数据，建议网络库使用AFNetworking类。

## ConfigData类

App运行周期内，用来存数数据的类，比如登录成功后Token，Uid等信息，一些关于核心数据的操作函数亦可以在该类中完成

## DBOPerata类

SQLite数据库的操作类，关于数据的增删改查，在此类中提供接口供外部的Controller调用，为了代码的可读性以及结构要求，请不要在Controller类中直接操作数据库。

## GlobalConfig类

App的配置参数类，对于一些常量参数，请在此类中直接定义，比如屏幕的高度，宽度，接口服务器地址等信息



图2

## SecondLevelView，SecondLevelViewController 类

基类，创建所有的ViewController的时候需要继承SecondLevelViewController，该类中提供一些基础的成员变量以及各种函数，具体请参见代码。