# 微信小项目代码规范

说明：本想着按照官方的开发规范来实现，奈何时间紧迫，又能力有限，所以执行如下简化版项目规范，另附比较好的规范网址，希望以此为目标。

[Laravel项目规范](https://laravel-china.org/docs/laravel-specification/5.5/whats-the-use-of-standards/510) 作者：[Summer](https://laravel-china.org/users/1)

[简书简化版项目规范](https://www.jianshu.com/p/e464a35e5ed2) 作者：[knghlp508](https://www.jianshu.com/u/63ba399e5f61)

[Laravel 的十八个最佳实践](https://laravel-china.org/articles/12762/eighteen-best-practices-of-laravel) 作者：[JokerLinly](https://laravel-china.org/Lang_ya_ju)

[Laravel中命名规范](https://blog.csdn.net/da_guo_li/article/details/78680767) 作者：[da\_guo\_li](https://me.csdn.net/da_guo_li" \t "_blank)

## 关于GIT

Git提交必须要有备注，要详细说明做了什么

如：

Edit：修改了什么

Delete：删除了什么

Insert：增加了什么

Save：修改了什么

1、提交代码务必写上注释。

2、注释一般要求，简洁，能够准确的说明问题。修改了多个文件建议提交Git时候分行写出具体修改的问题。方便后期合并代码、以及回滚代码。

3、打分支或者tag的时候，需要在注释说明本版本主要修改功能。

## 数据库命名

* 使用负数作为表名，例如users
* 多对多关系中，中间表命名用单数，例如article\_image
* 数据库以及表名使用均使用小写英文字母，单词之间采用下划线为分割符，例如article\_user
* 不要使用关键字作为数据库或表的名字，例如null，select等等。

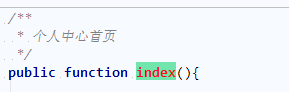
## 注释规范

### 类的注释

* 主要功能说明
* 作者 authr
* 时间



### **方法的注释**



### 代码的注释

I 代码内的注释建议使用 “//” 和代码层级保持一致

II 后期需要完善或者是可能出现问题的代码要 @todo 标识出来，方便后期排查代码



### 函数的注释

* 函数的主要用途
* 函数的参数说明
* 函数的返回值说明



## 公共模块说明

暂定

## 文件格式

### 文件编码

* 文件编码一般建议采用UTF-8无BOM

### PHP文件

* 对于只含有 php 代码的文件，我们将在文件结尾处忽略掉 "?>" 。防止多余的空格或者其它字符影响到代码。
* 不允许使用短标签。

### 代码缩进

* 代码缩进必须使用4个空格，不允许使用制表符。一方面保证代码的美观性，一方面保证多平台代码的可读性以及避免发生不可预见的问题。
* 类定义开始大括号建议和类命保持一致。
* IF分支开始大括号建议和if保持一致， IF语句必须有大括号，不建议省略大括号，以便提高代码可读性。
* 多个if语句建议空行。
* 较长IF判断建议写上开始、结束标记。
* SWITCH语句必须有default。
* 定义数组建议分行定义。 
* 一行代码不建议过长，一行代码过长时建议考虑分行。

## 命名规范

### 文件名命名规范

* 类文件名建议采用驼峰命名法，并且首字母大写 ( 例如：UserController.php，UserModel.php等 )。
* 配置文件建议以.config.php为后缀。配置项建议都使用大写。
* 方法文件不做要求。

### 类名命名规范

* 类名要求和文件名保持一致( 包括大小写一致 )。
* 建议采用驼峰命名法，并且首字母大写。

### 变量命名规范

* 变量命名要求简单明了，英文命名，严禁拼音以及无意义命名 ( 例如：$data2 , $a等 )。
* 长变量建议使用 \_ 隔开。( 例如： $user\_money用户金额， $user\_status用户状态 )
* Model 对象建议使用 类名 + \_ + model 命名。 ( 例如：$user\_model等 )。
* Controller 对象建议使用 类名 + model 命名。 ( 例如：$user\_controller等 )。

### 常量命名

* 常量定义由 大写字母 以及 ‘\_’组成。(例如：APP\_PATH，LOG\_PATH等 )。

### 属性命名

* 属性的命名使用驼峰法，并且首字母小写或者使用下划线“\_”，例如 tableName、\_instance，通常下划线开头的属性属于私有属性

### 方法命名规范

* 方法的命名使用驼峰法，并且首字母小写或者使用下划线“\_”，例如 getUserName，\_parseType，通常下划线开头的方法属于私有方法。
* 方法参数多个参数之间建议加上空格。
* 所有函数必须有访问控制符( public,protected,private)，必须存在控制符。 【 一般情况下，public表示全局都可以调用，protected方法只有本类、子类以及父类访问，private表示私有方法，只要本来可以调用】。
* static 表示静态方法，值能通过 类命 :: 方法名 调用，其中静态方法中值能调用静态方法以及静态属性。【建议接口之间通讯可以用静态类处理，调用方便，而且方便后期维护】

### 函数命名规范

* 函数的命名使用小写字母和下划线的方式，例如 get\_client\_ip
* 函数参数多个参数之间建议加上空格。

## 其他说明

1、尽可能使用单引号 '，而不是双引号''

2、php文件中绝不能出现html语句，html文件中尽可能避免出现php语句。

3、一个php文件只能包含一个类定义编码，以类名称作为文件名称。

4、使用完毕后的数组变量、对象变量、查询集合必须马上使用unset()、free\_result()释放资源。

5、把已经注释掉的代码删除，或者注明这些已经注释掉的代码仍然保留在源码中的特殊原因。

6、详细的注释，把主要功能注释出来，减少其他人阅读代码的用时，也方便以后排查问题。

7、当重复写一些代码时候，可以考虑封装为方法或者是类；

8、使用变量或者遍历数组时候，使用isset函数或是empty函数可以检查变量值是否为空。

9、程序中与其他服务端或者接口有交互时候，建议用注释表明每个字段的含义，方便排查问题。

10、必要时程序加入一些日志记录，方便排查问题。  
11、GIT提交需要加注释。 （GIT提交需要加注释，GIT提交需要加注释，GIT提交需要加注释，说三遍）

12、namespace声明后必须有一个空行，并且在声明块之后必须有一个空行use。

13、打开类的大括号必须在下一行，并且关闭大括号必须在主体后面的下一行。

14、打开方法的括号必须在下一行，并且关闭括号必须在主体后面的下一行。

15、代码重构使用PHPstor，快捷键ctrl+alt+L