**PHP及数据库标准**

**PHP部分**

一：开发框架：larevel

二：基本原则遵照larevel标准

三：系统根目录：\public\

四：MVC：

C：\app\Http\Controllers\

M：\app\Http\Model\

V：\resources\views\

五：命名规则：

基本格式遵照PSR标准：

1：文件名：大写开头的驼峰命名规范

2：类名：大写开头的驼峰命名规范

3：方法名：小写开头的驼峰命名规范

4：变量名：【驼峰法则/小写？？】

六：注释：图：注释.png

1：每个类前必须注释清楚开发者姓名、建类时间、类的功能

2：每个方法前必须注释清楚方法的参数（类型等传参事宜）、功能

3：方法中调用其他方法必须注释清楚调用方法的功能

4：方法中的其他函数或处理或别人不易看懂的地方请注释清楚

七：格式：

基本格式遵照PSR标准：

1. PHP代码文件必须以 <?php 或 <?= 标签开始；
2. PHP代码文件必须以 不带BOM的 UTF-8 编码；
3. PHP代码中应该只定义类、函数、常量等声明，或其他会产生 从属效应 的操作（如：生成文件输出以及修改.ini配置文件等），二者只能选其一；
4. 命名空间以及类必须符合 PSR 的自动加载规范：[PSR-0](https://github.com/PizzaLiu/PHP-FIG/blob/master/PSR-0-cn.md) 或 [PSR-4](http://segmentfault.com/blog/gtg/1190000002521658) 中的一个；
5. 类的命名必须遵循 StudlyCaps 大写开头的驼峰命名规范；
6. 类中的常量所有字母都必须大写，单词间用下划线分隔；
7. 方法名称必须符合 camelCase 式的小写开头驼峰命名规范。
8. 类的常量中所有字母都必须大写，词间以下划线分隔。
9. 每个namespace命名空间声明语句和use声明语句块后面，必须插入一个空白行。
10. 类的开始花括号({)必须写在函数声明后自成一行，结束花括号(})也必须写在函数主体后自成一行。
11. 方法的开始花括号({)必须写在函数声明后自成一行，结束花括号(})也必须写在函数主体后自成一行。
12. 类的属性和方法必须添加访问修饰符（private、protected以及public），abstract以及final必须声明在访问修饰符之前，而static必须声明在访问修饰符之后。
13. 控制结构的关键字后必须要有一个空格符，而调用方法或函数时则一定不能有。控制结构的开始花括号({)必须写在声明的同一行，而结束花括号(})必须写在主体后自成一行控制结构的开始左括号后和结束右括号前，都一定不能有空格符。

八：PHP程序修改

1：理论上，自己修改自己的程序，他人不可随意修改别人的代码

2：在他人的类中添加方法时，必须注明清楚添加者姓名；并且以:“//以下是某某添加的方法”开始，以“//以上是某某添加的方法”结束。 如图：注释1.png

3：在他人的方法中修改代码时，应当重新复制此方法，以第二条的形式书写；

九：sql语句

1：sql语句应当简洁、明了；不应该有重复的计算；如图：优化.png

2：少使用not in 与 <>

查询语句中的ont in可使用left join代替；如sql优化1.png

3：不在索引列上使用函数；例应当将：

SELECT \* from `index` where UNIX\_TIMESTAMP(deleted\_at)<=1470598407;

改为：SELECT \* from `index` where deleted\_at<=' 2016-08-08 03:33:27';

4：where语句

将经常使用，选择性高的列放在where语句的前面

5：order by

1：Order by子句的顺序和where的顺序一样；

2：order by 子句应该是第一张表中的字段；

十：事务

应当尽可能的将事务中的逻辑转移到事务外；尽可能的减少事务的运行时间，从而减少mysql的执行时间与锁的时间；

**MYSQL部分**

一：数据库：mysql

二：引擎：InnoDB;

三：数据库由设计人员进行增删改，其他人员若需要增加修改等，必需由设计人员进行增改

四：命名规则：

1：全部以英文小写命名（不得以拼音命名），字母间以‘\_’间隔；

2：前缀采用‘abon’命名，以‘\_’间隔；

五：数据字典：

1：数据字典必须完整包括字段名，类型，参数,默认值，注释（注释必须完整）；

2：由设计人员设计出来后，发送给每一个程序员，若有增改后，再由设计人员发送给每一个程序员；

六：数据设计：

1：主键：必须是数字类型

2：索引：根据实际情况，具体而定；

3：联合索引：根据实际情况，具体而定；

4：外键：采用实际外键形式，而不采用形式外键；

5：enum类型：由于enum类型的采用索引取值，易造成程序员的书写错误，本OA中采用tinyint类型，而不采用enum类型；

6：时间类型：本OA中采用timestamp类型；

时间字符串比较（推荐使用方式）：

select id from `table` where time<=”2016-8-15 11:17:56”;

SELECT \* from `index` where deleted\_at<='2016-08-09';

时间戳比较（应当少使用）：

select id from `table` where UNIX-TIMESTAMP（time）=”时间戳”;

SELECT \* from `index` where UNIX\_TIMESTAMP(deleted\_at)<=1470598407;

七：sql语句（简洁，明了）

1：少使用not in 与 <>

查询语句中的ont in可使用left join代替；如sql优化1.png

2：不在索引列上使用函数；例应当将：

SELECT \* from `index` where UNIX\_TIMESTAMP(deleted\_at)<=1470598407;

改为：SELECT \* from `index` where deleted\_at<=' 2016-08-08 03:33:27';

3：where语句

将经常使用，选择性高的列放在where语句的前面

4：order by

1：Order by子句的顺序和where的顺序一样；

2：order by 子句应该是第一张表中的字段；