### 1.PHP知识总结

## 1.1 PHP/Laravel常用命令

（1）启动Laravel命令：php -S localhost:8888 -t public（进入Laravel根目录后） PhpStorm 激活方式:http://www.0-php.com:1017

（2）注意：Mysql 的 update 语句格式update tablename set column\_name=’’ where ?;

（3）Laravel 创建一个Controller

--php artisan make:controller DemoController

--php artisan admin:make UserController --model=App\User

（4）在Laravel 框架中 {{}}可用来输出传递给视图的参数

（5）在Laravel blad中 可以通过@yield()来定义一个公共模板

@extends()来继承哪个公共域,还可以用@if,@foreach来控制流程

（6）Controller 向视图传递参数用 compat();

（7）php artisan migrate 执行创建数据库的命令

（8）在使用$table->dropColumn()方法时需要用composer引入

--composer require doctrine/dbal;

--Carbon\Carbon::now();用来获取当前时间

-- php artisan tinker;命令行工具

-- php artisan make:model name;创建数据映射Bean

-- te\_users\_table --create=users

（9）php artisan passport:install解决缺少key文件

## **1.2** 数据库**操作**

（1） -- $d->save();保存数据

-- $d=App\User::where('age','>',10)->get();//where查询

-- $d=App\User::find(1);//通过ID查询

-- $d=App\User::where('age','>',10)->first();

-- $d=App\User::create([]);//新建插入数据

-- $d->update([]);//更新查询出来的数据

-- $d->delete;//删除某条查询出来的数据

（2）可以通过DB::select（）;写原生的sql语句

（3）如何从数据库中筛选出重复的字段：

SELECT id,tel FROM table GROUP BY tel HAVING COUNT(tel)>1;

(4)将某个表中的字段移动到对应表中相应的字段中

UPDATE users a,

(SELECT b.user\_id,b.tel,b.tel\_activate FROM userdatas b) b

set a.tel=b.tel,

a.tel\_verified=b.tel\_activate

WHERE b.user\_id=a.id;

(5)查询时可用:whereIn(),whereNotIn()

(6)解除数据库死锁:show full processlist;

select \* from information\_schema.innodb\_trx;

## **1.3 HTML的常识**

（1）a标签的使用href=”{{action(‘’,[])}}”;

（2）表单Laravel 5.1以后 composer require laravelcollective/html;

（3）css引用 href=”/css/...”//laravel默认是从public目录开始寻找

laravel

（1）新建一个Laravel项目

composer create-project --prefer-dist laravel/laravel zhihoo

Php namespace

作用:用来对不同类型的代码进行分层（domin/service/repository）

·explode("|",$str);//字符串分割函数

## **1.4 PHP数组的操作**

(1)bool ( mixed needle, array array [, bool strict] )

表 4-1

|  |  |
| --- | --- |
| needle | 需要在数组中搜索的值，如果是字符串，则区分大小写 |
| array | 需要检索的数组 |
| strict | 可选，如果设置为 TRUE ，则还会对 needle 与 array 中的值类型进行检查 |

# substr(string,start,length),函数返回字符串的一部分。

# array\_keys():将关联数组中的所有键值返回成一个数组

# explode()函数以指定分割=符分割数组

1. array\_slice() 函数在数组中根据条件取出一段值，并返回。该函数可用来处理当数组索引不对的时候进行重置索引。

表 4-2

|  |  |
| --- | --- |
| **参数** | **描述** |
| array | 必需。规定数组。 |
| start | 必需。数值。规定取出元素的开始位置。 0 = 第一个元素。  如果该值设置为正数，则从前往后开始取。  如果该值设置为负数，则从后向前取 start 绝对值。 -2 意味着从数组的倒数第二个元素开始。 |
| length | 可选。数值。规定被返回数组的长度。  如果该值设置为整数，则返回该数量的元素。  如果该值设置为负数，则函数将在举例数组末端这么远的地方终止取出。  如果该值未设置，则返回从 start 参数设置的位置开始直到数组末端的所有元素。 |
| preserve | 可选。规定函数是保留键名还是重置键名。可能的值：  true - 保留键名  false - 默认。重置键名 |

# 如果要从数组中删除一个元素,请使用unset()函数,array\_shift()和array\_pop()函数分别从数组的开头和末尾删除一个元素。

1. array\_flip(),将关联数组中的key与值反转并返回新数组
2. array\_merge(),将两个数组合并,若有相同的key时将会保留第二个数组中的key对应的值
3. array\_map()函数给每个数组元素应用一个回调。

(10)shuffle()将数组进行随机排序

(11)in\_array(a,b)判断值a是否存在于数组b中，array\_key\_exists()判断数组中是否存在某个key

(12)array\_search(a,b)搜索数组b中是否存在值a,存在则返回第一个值为a的key,否则返回false

(13)$arrayObject=new ArrayObject();//将数组看作一个对象

## **1.5 PHP常用时间函数**

(1)strtotime()将时间格式转化为时间戳

(2)date(format,timestamp),将时间戳转为指定格式

Y ：年（四位数）大写   
 m : 月（两位数，首位不足补0） 小写   
 d ：日（两位数，首位不足补0） 小写   
 H：小时 带有首位零的 24 小时小时格式   
 h ：小时 带有首位零的 12 小时小时格式   
 i ：带有首位零的分钟   
 s ：带有首位零的秒（00 -59）   
 a：小写的午前和午后（am 或 pm）

(3)date("Y-m-d 23:59:59", strtotime(-date('d').'day'));拿到上月末的时间戳

(4)date('Y-m-01 00:00:00',strtotime('-1 month'));拿到上月初开始的时间戳

(5)date("Y-m-d");拿到当天开始的时间戳

(6)date\_default\_timezone\_get();获取当前时区格式

date\_default\_timezone\_set('PRC')；临时时区格式设置

## **1.6 PHP Json数据处理**

(1)中文不转码json\_encode($request->get('data'),JSON\_UNESCAPED\_UNICODE),

## **1.7 获取浏览器地址信息**

(1)$\_SERVER全局变量的使用

$\_SERVER['SERVER\_NAME'],

$\_SERVER['REQUEST\_URI'],

$\_SERVER['HTTP\_HOST'],

$\_SERVER[‘REMOTE\_ADDR’];拿到浏览当前页面用户的ip地址

(2)file\_get\_contents();可以用来请求url拿到返回的数据

## **1.8哈希加密与解密（用户密码校验）**

bcrypt()-哈希加密函数

Hash::check(a,b)-a.所要比对的密码,b.老密码

## **1.9 PHP中字符串匹配模式**

preg\_match()：接收三个参数,1.正则表达式,2.匹配的字符串,3.将符合正则表达式中的字符放到参数三中（数组）

## **1.10动态调用函数**

call\_user\_func（[$publice,’demo’],$parm1,$parm2）

**1.11 PHP字符串处理**

(1)ucwords()首字母转为大写,lcfirst()首字母转为小写

(2)单引号只解析\\，\’,双引号所有转义字符都解析,单引号不解析变量,双引号解析变量,单引号比双引号效率高

(3){}可用来对字符串进行增删该查，{}可以定义变量整体

(4)false，null转化为空字符串,true转化为1,数组转化为字符串为Array

**1.12 文件资源处理**

(1)fopen();打开某资源文件

**1.13 PHP基础总结**

(1)类型转化gettype(),settype();

(2)PHP中可用is\_bool,is\_arrary等函数进行数据类型检测

(3)xor;逻辑异或运算符

(4)@;错误控制运算符,可以屏蔽出现错误的表达式

(5)PHP的赋值操作的顺序是从右到左的

(6)PHP在使用默认参数时,默认参数必须放在非默认参数的右边

(7)unset();取消变量名和变量之间的绑定

(8)双引号可以解析变量;

(9)通过$\_FILES预定义变量可拿到上传的文件数组

(10)getdate();可拿到指定时间戳的关联数组信息

(11)找ttf文件时路径前加一个.

(12)dirname(\_\_FILE\_\_);拿到当前目录

**1.14.PHP类操作**

(1)反射:根据类名反向实例化类

$refect=new \ReflectionClass($classname);

$company=$refect->newInstance(['name'=>'ddddd']);

**1.15.PHP匿名函数使用技巧**

(1)可以使用use 向匿名函数传递外部变量

(2)

**1.16.PHP回调函数**

# 正则表达式知识

## ^匹配开始,$匹配结束，\*:0次或多次,+:1次或多次,?:0次或1次.

(1)preg\_replace\_array();函数使用数组顺序替换字符串中的给定模式

# PHP Laravel框架知识

## **3.1.collection集合常用方法**

(1)$collection->flatten(1);取出集合中指定深度的数据

(2)$collection->filter();接受一个闭包函数返回集合中符合条件的元素,如果没有提供回调函数，集合中所有返回 false 的元素都会被移除.

(3)forget()来移除集合中对应的内容

(4)$collection->groupBy('account\_id');根据集合中的指定key进行分组

(5)$collection->has('product');判断集合中是否存在指定的key

(6)toArray()；集合转为array

## **3.2.PHP辅助函数**

### 3.2.1 array常用辅助函数

(1)array\_collapse();将多个数组合并成一个数组

(2)array\_divide();将数组的key-value分成两个数组

(3)array\_dot();将多维数组平铺成一维数组

(4)array\_except();删除数组中指定的键-值对

(5)array\_get();从指定数组中取出指定key,可接收默认值

(6)array\_has($array,‘people.people1.name’);检查数组中是否存在指定的项目,可用.语法

(7)array\_only();返回指定数组中的指定键-值对(可以为数组)

(8)array\_random();返回数组中的随机一个值

(9)array\_pull();返回并从数组中删除键-值对,可以有默认值

(10)array\_prepend($array, 'Desk', 'name');在数组头部压入一个键-值

(11)array\_set($array, 'products.desk.price', 200);在深度嵌套的数组中设置一个值

(12)array\_where();接受一个闭包过滤数组

(13)array\_wrap();将给定的值包装成一个数组,若给定的值为null则返回一个空数组[]

(14)data\_fill();填充数组的指定深度

(15)head();返回数组中第一个元素last()

(16)mt\_rand(0,100);返回指定区间的随机数

### 3.2.2 字符串常用辅助函数

(1)camel\_case('foo\_bar');函数将给定的值符传转换为「驼峰命名」

(2)preg\_replace\_array();函数使用数组顺序替换字符串中的给定模式

str\_replace\_array('?',[],$str);替换数组中指定字符 str\_replace\_first();

str\_replace\_last();

(3)snake\_case();将给定的字符串转换为「蛇形命名」

(4)starts\_with(),end\_with();

(5)str\_contains();判断给定的字符串是否包含给定的值

(6)str\_finish();将字符串以给定字符结尾并返回

str\_start();

(7)str\_limit(20,(...));将字符串从给定长度截断并将后边显示为指定样式

(8)str\_plural();将字符串转为复数形式,第二个参数可以为数量

(9)str\_random();生成一个指定长度的随机字符串

(10)title\_case();将字符串首字母变大写

(11)ends\_with();判断字符串已指定字符结尾

(12)strlen();获取字符串长度

(13)strcmp(),strcasrcmp();比较两字符串是否相等,后者不区分大小写

(14)strstr();从指定字符串中检索指定字符,并截取

(15)substr\_count();检索子字符串出现的次数

(16) str\_ireplace(a,b,c,d);替换字符串,参数a：需要查找的字符串,b:替换的字符串,查找的范围,执行的次数(可选)

(17)explode(),implode();按指定字符分割字符串,将数组组已指定字符,组装为字符串

(18)

### 3.2.3 其他常用辅助函数

(1)abort\_if();动态抛出指定错误响应abort\_unless();boolean值为false时抛出

abort(403);抛出指定的异常类型

(2)back();生成一个重定向 HTTP 响应到用户之前的位置

(3)blank();判断给定的值是否为空,可为字符串,数组,对象,集合，相反的函数filled()

(4)config();可以读取指定的配置信息,并可以在运行时设置值

(5)csrf\_field();生成包含 CSRF 令牌值的 HTMLhidden 表单字段csrf\_token();获取当前令牌值

(6)dump();断点调试不打断程序

(7)dispatch();将给定的任务推送到队列中

broadcast()；广播给指定的监听器

(8)encrypt(),decrypt(),Laravel自带的加密解密函数

(9)event();将给定的事件分派给它的监听器

(10)info();将一些有用的信息写入日志,logger();将debug级的消息写入日志

(11)report();该方法抛出异常

(12)request();用来拿到请求中的数据

(13)response();获取响应实例

### 3.2.4数学运算函数

(1)pow(x,y);返回 x的 y次方的幂

(2)sqrt();返回平方根

## **3.3.Laravel文件操作**

(1)asset('storage/数据库密码.txt');生成文件路径

(2)Storage::disk('local')->put('/public/file/file.txt','77777');存放文件

(3)Storage::get('/public/file/file.txt');读取指定的文件内容

(4)Storage::prepend('/public/file/file.txt', 'Prepended Text');在文件前,写入指定的数据

Storage::append('/public/file/file.txt', 'Appended Text');在文件后写入指定的数据

(5)Storage::copy('file.txt','/public/demo/file.txt');将文件复制到指定目录下

(6)Storage::move('old/file.jpg', 'new/file.jpg');

(7)在Controller中可以用file(‘key’);获取前端传递的文件

(8)Storage::delete('/public/file/file.txt');删除指定的文件

(9)Storage::allFiles('/');取得指定目录下包含子目录的所有文件

Storage::files($directory);取得指定目录下的所有文件

(10)Storage::directories(/);Storage::allDirectories(/);

(11)Storage::makeDirectory('/666/777/888');递归创建目录

Storage::deleteDirectory('/666');删除指定目录下的所有东西

## **3.4.Laravel队列queue**

(1)dispatch(new Job()->onQueue('per\_timeout'))；指定某个job所在的任务队列名称

(2)php artisan queue:work --queue=name;指定监听某个任务队列,若无则默认监听default队列

## **3.5.Laravel-admin 框架知识**

(1)在数据表对应得Controller中$form->saving();可以在数据持久话之前进行数据处理.

## **3.6.Laravel框架一些便捷的方法(5.7)**

(1)$request->exceptsJson();判断预期返回值是否为json

## **3.7.Auth验证**

(1)Auth::attempt();方法可以指定用户的登陆字段。

## **3.8.模型Model**

(1)getAttribute(keys);--可以动态获取模型属性

# 4.Linux知识点

(1)crontab -e；在linux系统中设定cron定时任务命令.

\* \* \* \* \* php /home/vagrant/PHPwork/branch/artisan schedule:run >> /dev/null 2>&1

(2)重置管理员密码:sudo passwd root

(3)更换登陆用户为管理员身份:su root

# 5.Redis基本知识

## **5.1 redis基本数据类型**

# (1)string数据类型:

set name value;设置一个类型

incr name;将某个值自增

(2)list数据类型:(不要求里边元素唯一，有序)

lpush name value;压入一个值

lpop name;弹出list中第一个元素

len name;拿到list的长度

1. set类型：(无序)

sadd set1 value;向set1中加入一个值

sismember set1 value;判断某个值是否在set1中

srem set1 value ;将值从set1中移除

(4)hash类型

hset hash1 key value;向hash集合中插入一个值

hash hash1;获取指定集合的长度

hget hash1;获取指定集合中的key值

hmget hash1;获取指定集合中的key值

(5)sort set类型

zset zset1 scores value;//存放元素

zrange zset1 0 10 withscores;//按rank值返回数据

## **5.2 redis常用命令**

# (1)redis-cli.--进入redis客户端

flushall.--清空redis中所有缓存

# 设计模式

# (1)单例模式:

1.非public的构造方法

2.拥有一个保存类的实例的静态成员变量$\_instance;

3.拥有一个访问这个实例的公共的静态方法

(1)配置文件中的api配置信息必须与accept中的请求头信息一致,才能实现版本控制.