## Laravel笔记

laravel是一个简介优雅，功能强大的全栈框架

#注释必须是在行开头才有效，而且DocumentRoot "C:\xampp\htdocs\laravel-master"后面不能有\

利用composer来安装laravel的方法有两种

1.先安装laravel安装器，composer global require "laravel/installer"，安装过程需要联网，装完后，添加

~/.composer/vendor/bin 到系统路径中，进入到laravel项目目录，直接在命令行通过laravel new pro\_name创建项目

速度较快

2.直接composer create-project --prefer-dist laravel/laravel xxx 来安装，速度较慢

以上两种安装方法都是要联网的

这是什么意思？

Generating autoload files

> php artisan clear-compiled

> php artisan optimize

Generating optimized class loader

Compiling common classes

> php -r "copy('.env.example', '.env');"

> php artisan key:generate

Application key [7HrRdCiXFDNyGLEDAWPJ0jRuIttslyL9] set successfully.

new [--dev] [--5.2] [--] [<name>]

无法使用laravel new来安装的原因是，PHP的版本太低，要把高版本的PHP设置在环境变量中才能使用

使用composer下载以前版本的laravel

composer create-project laravel/laravel=4.1 xxx目标文件夹

其他盘上的虚拟目录无法访问：

修改Apache配置文件，如下，即可

<Directory />

AllowOverride all

Require all granted

</Directory>

或者在虚拟目录下设置 require all granted 也可以

如果不能正常访问phpinfo或者homepage，很可能是常规端口设置错了，改为与配置文件相同的端口即可

静态资源都是放在source目录下

谷歌插件网：http://chromecj.com/utilities/2015-04/423.html

离线安装postman：http://chromecj.com/utilities/2015-04/423.html

就是将下载的crx文件改名为.zip后缀，解压，在谷歌浏览器设置中添加扩展包，提示错误时，修改\_metedata为metedata，重试即可安装成功

无法发送出get的其他请求？

Route::get('post/{d}/get/{i}', function ($d,$i) {//这里的参数是依次与前面参数相对应，?表示可选

return $d;

reture $i;

});

如何翻墙呢？

访问网站流程：路由->控制器->视图

可以使用artisan 命令快速创建控制器

php artisan make:controller UserController

在Controler下新建一个admin文件夹，用artisan创建一个控制器，拖到admin，修改这个控制器的namespace为当前路径，并且还要引入Controller类，也就是：use App\Http\Controllers\Controller ;

最后在路由router.php中设置 Route::get('admin',

'admin\AdminController@index')，就是要找到这个控制器

高级路由

路由的命名：

Route::get(

'user',['as'=>profile,function(){

return '命名路由';

}]

);

Route::get('user/profile', function () {

//

})->name('profile');

Route::get('user/profile', 'UserController@showProfile')->name('profile');

可以通过辅助函数 route 为该命名路由生成 URL：

$url = route('profile');

这里的返回值就是路由的地址，也就是浏览器上的访问地址

如果命名路由定义了参数，可以将该参数作为第二个参数传递给 route 函数。给定的路由参数将会自动插入到 URL 中：

Route::get('user/{id}/profile', ['as' => 'profile', function ($id) {

//

}]);

route('profile', ['id' => 1,'name'=>22]);

后面数组是路由参数数组，这样生成的路由就会自动加上相应的参数

路由的前面部分是URL中需要输入的部分，后面部分可以是function()，可以是控制器@方法，表示指向那个控制器

路由的分组：就是把具有相同属性的路由放在一个group中

数组存放的就是各个路由的共同特性

Route::group(['prefix'=>'admin'],function(){

route::get(xxx,xxx);

route::get(xxx,xxx);

});

中间件

就是在路由群组中加一个middleware=>web的键值对

session 必须放在middelware中才能被访问

在laravel中创建session的方法是：session(['key'=>'value']),

直接通过session('key')来访问，只有访问过相应的页面才能有session值存在

中间件实在kernel.php中定义的，有middlewaregroup，可以在这个文件中依样创建自己的中间件

还可以用php artisan make:middleware name命令创建中间件,php artisan 直接查看artisan的全部命令

中间件也是按数组来存放的'middleware'=>['web','xxx']

中间件就是在路由和界面之间挡了一层

中间件用法总结：首先在kernel.php文件中创建自己的中间件，然后利用php artisan make:middleware middle\_name 创建中间键的文件，最后在路由中通过middleware绑定中间件到路由。中间件文件中可以做很多操作来达到过滤路由的作用

清空session的方法是将他的值设置成null

访问带中间件路由的顺序：路由->中间件->页面，可以用于控制登录验证

资源控制器

使用artisan来查看所有路由

php artisan route:list

为什么显示资源控制器不存在?

后面的参数必须写完整路径，否则会找不到：Route::resource('article', 'admin\ArticleController');

为什么namespace和prefix不能同时存在？

答：如果是访问带有prefix的路由，在地址栏必须输入prefix的值，这样才能正常访问!

好的，问题解决

学习的方法是，不懂？没关系，继续看下去，继续学下去，到最后总是会解决的

在laravel中大量运用了数组，只要是有多个值可能的，都会用到数组，比如session，路由组的共同属性，多参数路由中，通过route('name',['pram1'=>'val1','pram2'=>'val2'])

如果访问其他文件夹的控制器，一般要在路由中设置命名空间，否则就要写出完整的xxx文件夹/xxx控制器

视图部分

return view('view\_name')

在控制器中把参数传到视图

$name='朱明浩';

view('view\_name')->with('name',$name);这里的with支持连缀

也可以直接在with中传一个数组 return view('my\_view')->with(['name'=>$name,'age'=>$age]);

，还可以把要传过去的信息定义在外面一个数组中，然后with('data',$data)

在视图模板中收参

可以直接用php原生方法echo输出，但更方便的是使用模板输出:{{$variable}}

最佳方法：

控制器部分

$data=array(

'name'=>'白彩虹',

'age'=>22

);

return view('my\_view',$data);

视图部分

<h1>这是<?php echo $name?>(年龄：<?php echo $age?>岁)的第一个视图</h1>

如果要一次穿多个变量进去，可以用compact()函数

$data=array(

'name'=>'白彩虹',

'age'=>22

);

$tittle='我是朱明浩';

view('my\_view',compact('data','tittle'))//这里直接传变量的名字，$也不需要

视图中

$data['name'] $tittle

blade 模板引擎：

作用：使得代码简介

{{$tittle}}@{{$tittle}}：@后面的内容原样输出

在模板中可以用or设置值为null变量的默认值

{{$name or '朱明浩'}}

如果访问的变量本身就不存在，该如何处理？

{{isset($name)?$name:$name='zmh'}}

{{ isset($data['name'])?$data['name']:$data['name']='朱明浩'}}

可以把一段js代码当做字符串传到模板视图中

{!!$javascript!!}

默认情况下，blade会将代码片段（也就是带有<>）部分，自动转换成实体，原样输出

因为js代码只能在浏览器中执行，所以不能用postman测试生成的js代码！！！

blade模板的流程控制

@if($data['score']>=60)

及格

@else

不及格

@endif

unless语句 （除了）

@unless(条件部分)

循环：@for()

xxx

@endfor

@for($i=0;$i<$data['num'];$i++)

{{$i}}<br>

@endfor

对于变量的输出，必须要用{{}}

foreach函数

@foreach(数组 as $value)

{{$value}}<br> 这样就可以将数组内容全部输出

@endforeach

@forelse($data['article'] as $value)

{{$value}}<br>

@empty

如果没有信息，则显示这里的信息

@endforelse

forelse的用法和foreach用法很相似，只是多了个@empty如何处理的部分

循环还可以嵌套使用，，比如foreach中嵌套if

@foreach($array as $key => $value)

@if($key>2)

{{$key.'--'.$value}}<br>

@endif

@endforeach

模板中的变量同样可以用字符串隔开

blade模板子视图

分析：对于一个网站的所有网页，一般头部尾部的内容格式是一样的，变化的只是中间部分，可以将头部和尾部独立放在一个xxx.blade.php文件中，在需要的页面中使用@include('xxx.xxx')进行引入

include(xxx1,xxx2)他的第二个参数是一个参数数组

['page'=>'首页']

在include部分传参，在公共部分接受参数

哪一部分传了参数就在哪一部分接受参数，比如在header传参，则是在header.blade.php 中接受参数

制作一个网页模板，具有相同的头部和尾部

模板部分：

<div class="header">我是头部</div>

@yield('content')

{{--这里的作用是标记一个空间存放可变内容，也可以用@section('content') @show--}}

<div class="bottom">我是头部</div>

</body>

页面部分：

{{--先继承home.blade.php,然后用section往content部分填写内容--}}

@extends('layouts.home')

@section('content')

<div class="middle">我是layouts生成的内容</div>

@endsection

{{--这里是子视图，如果要调用主模板视图中的内容，可用@parent--}}

@extends('layouts.home')

@section('content')

@parent

@endsection

连接数据库

首先要在.env 中配置数据库类型，名字，用户，密码

在 database.php 中进一步设置

在控制器中引包use Illuminate\Support\Facades\DB;

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=laravel

DB\_USERNAME=root

DB\_PASSWORD=

tips：可以通过env()函数读取.env文件中的配置项，可以通过config()函数读取configuration文件夹下各个配置文件的相关值

echo env('DB\_CONNECTION');

echo '<br>';

echo config('app.timezone');

连接数据库：

$user=DB::table('user')->where('user\_id','>',1)->get();

dd($user);//相当于print\_r() exit;的组合

数据库模型：不能暴露在app外面，所以要把他移动到app/httpd/Model

ORM 模式：就是数据的对象关系映射，将数据表的一条记录当做是一个对象，记录的字段就是该对象的属性

$user=User::find(1);//返回主键为1的记录

$user->user\_name='朱明浩';//修改记录的值，先要在模型中禁用时间戳才能更新 public $timestamps=false;

$user->update();

dd($user->toarray());//返回的是一个数组