使用composer安装Laravel应用

composer create-project --prefer-dist laravel/laravel blog 5.5.\*

配置虚拟主机,配置数据库

Route::get('/', 'StaticPagesController@home');

Route::get('/help', 'StaticPagesController@help');

Route::get('/about', 'StaticPagesController@about');

创建控制器

php artisan make:controller StaticPagesController

布局文件

<!DOCTYPE html>

<html>

<head>

<title>Sample App</title>

</head>

<body>

@yield('content')

</body>

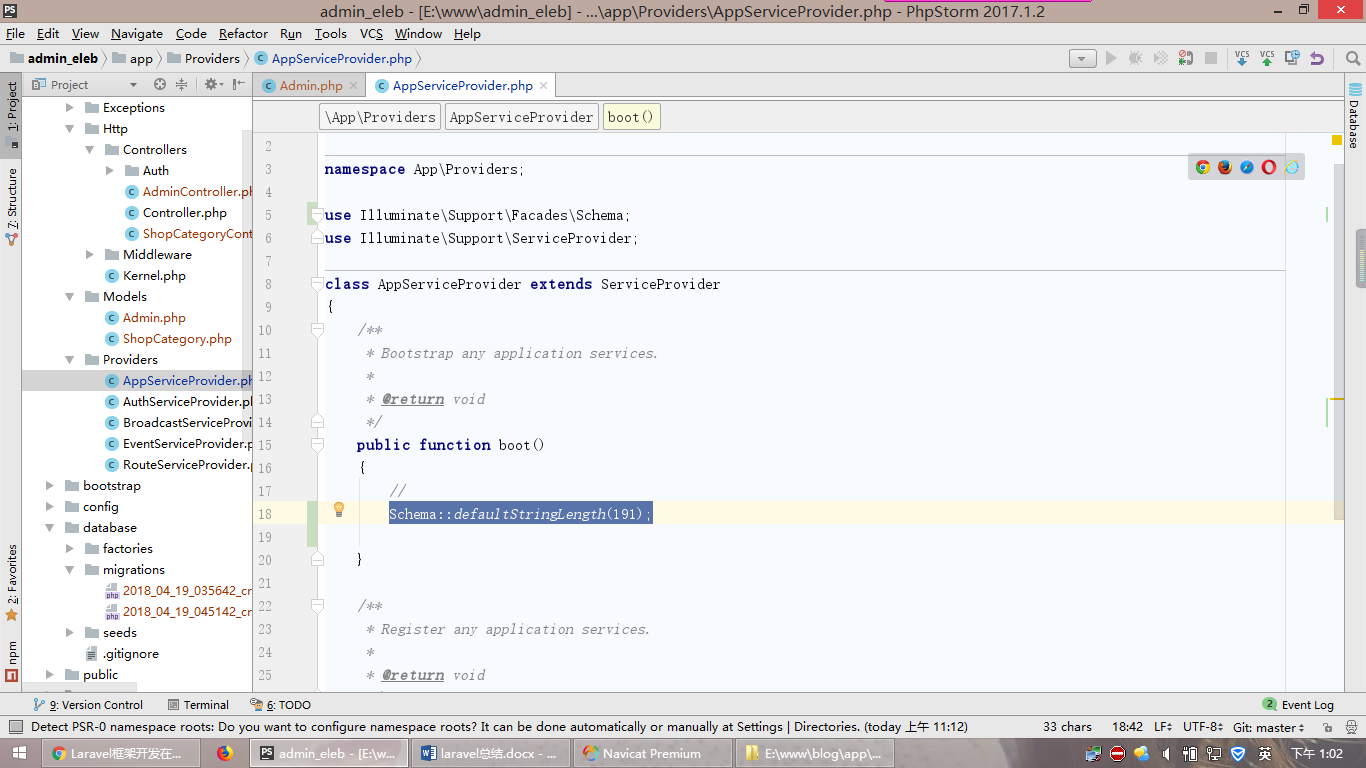
数据库迁移:

php artisan make:model 模型 -m

php artisan migrate

设置字段添加的长度:

Schema::*defaultStringLength*(191);



设置主从模型

public function art\_cate(){

return $this->belongsTo(ArtCategory::class);

}

设置路由

Route::resource('users', 'UsersController');

数据库充值

php artisan migrate:refresh

CSRF（跨站请求伪造）

{{ csrf\_field() }}

错误信息

@if (count($errors) > 0)

<div class="alert alert-danger">

<ul>

@foreach($errors->all() as $error)

<li>{{ $error }}</li>

@endforeach

</ul>

</div>

@endif

消息提示

session()->flash('success', '欢迎，您将在这里开启一段新的旅程~');

提示

@foreach (['danger', 'warning', 'success', 'info'] as $msg)

@if(session()->has($msg))

<div class="flash-message">

<p class="alert alert-{{ $msg }}">

{{ session()->get($msg) }}

</p>

</div>

@endif

@endforeach

自增值

DB::table('users')->increment('votes');

分页

$users = App\User::paginate(15);

页面

{{ $users->links() }}

验证码’

composer require mews/captcha

修改config/app.php,在providers添加

Mews\Captcha\CaptchaServiceProvider::class,

'providers' => [

// ...

Mews\Captcha\CaptchaServiceProvider::class,

]

在aliases添加

'Captcha' => Mews\Captcha\Facades\Captcha::class,

'aliases' => [

// ...

'Captcha' => Mews\Captcha\Facades\Captcha::class,

]

运行以下命令生成配置文件 config/captcha.php：

php artisan vendor:publish

然后选择7

前端

<input id="captcha" class="form-control" name="captcha" >

<img class="thumbnail captcha" src="{{ captcha\_src('flat') }}" onclick="this.src='/captcha/flat?'+Math.random()" title="点击图片重新获取验证码">

文件上传

app/Handlers/ImageUploadHandler.php

// 只允许以下后缀名的图片文件上传

protected $allowed\_ext = ["png", "jpg", "gif", 'jpeg'];

public function save($file, $folder, $file\_prefix)

{

// 构建存储的文件夹规则，值如：uploads/images/avatars/201709/21/

// 文件夹切割能让查找效率更高。

$folder\_name = "uploads/images/$folder/" . date("Ym/d/", time());

// 文件具体存储的物理路径，`public\_path()` 获取的是 `public` 文件夹的物理路径。

// 值如：/home/vagrant/Code/larabbs/public/uploads/images/avatars/201709/21/

$upload\_path = public\_path() . '/' . $folder\_name;

// 获取文件的后缀名，因图片从剪贴板里黏贴时后缀名为空，所以此处确保后缀一直存在

$extension = strtolower($file->getClientOriginalExtension()) ?: 'png';

// 拼接文件名，加前缀是为了增加辨析度，前缀可以是相关数据模型的 ID

// 值如：1\_1493521050\_7BVc9v9ujP.png

$filename = $file\_prefix . '\_' . time() . '\_' . str\_random(10) . '.' . $extension;

// 如果上传的不是图片将终止操作

if ( ! in\_array($extension, $this->allowed\_ext)) {

return false;

}

// 将图片移动到我们的目标存储路径中

$file->move($upload\_path, $filename);

return [

'path' => config('app.url') . "/$folder\_name$filename"

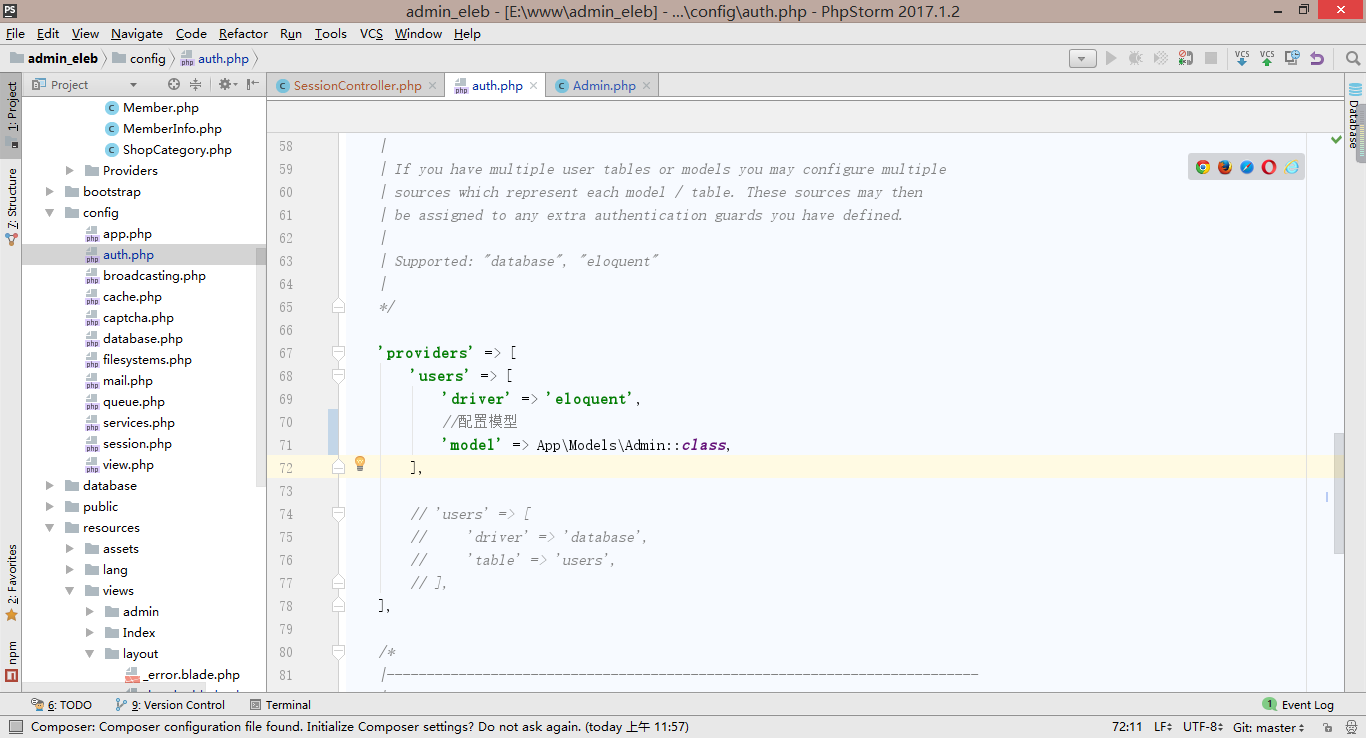
];

}

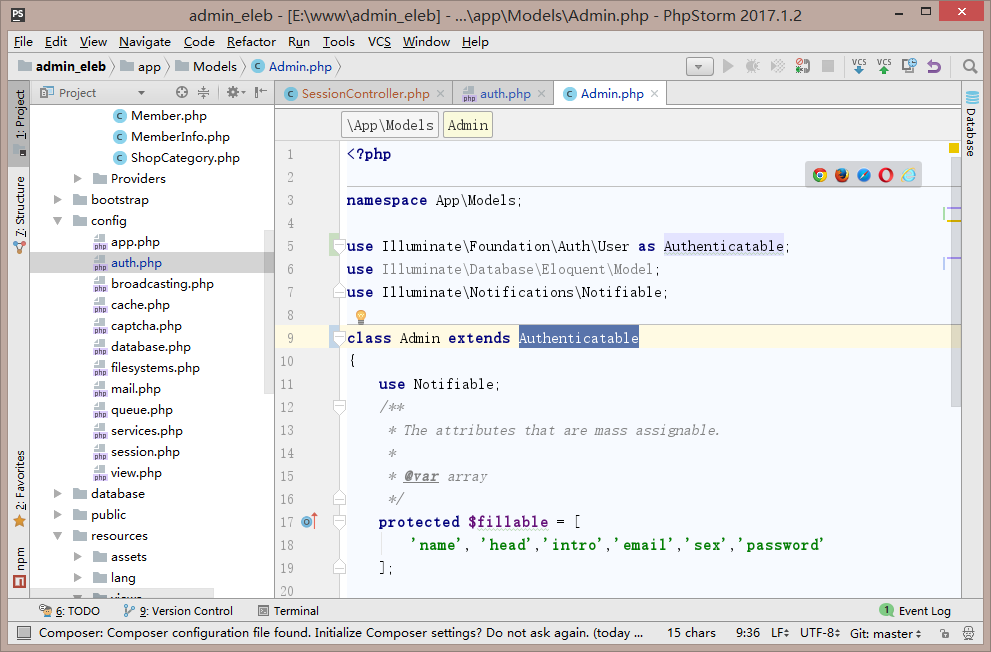
’

使用别的表登录

1.修改config 下面的auth.php配置文件



2.将要登录的模型修改继承



**use** Illuminate\Foundation\Auth\User **as** Authenticatable;

标识选中的字