**#laravel auth验证模块 by wooght 2017**

php artisan make:auth #开启auth功能

[puwenfeng@localhost laravel]$ php artisan make:auth

Authentication scaffolding generated successfully.

#成功后 resources/views得到auth,layouts文件夹

#成功后路由添加如下内容

Auth::routes();

Route::get('/home', 'HomeController@index')->name('home');

至此,用户验证功能开启并可以正常使用

#开启多用户认证功能

打开配置文件auth.php

添加guard用户,及需要认证的用户

//用户类型

'guards' => [

'web' => [

'driver' => 'session',

'provider' => 'users',

],

//添加后台用户类型

'admin'=>[

'driver' => 'session',

'provider' => 'admins',

],

'api' => [

'driver' => 'token',

'provider' => 'users',

],

],

定义数据模型和验证类型

'providers' => [

'users' => [

'driver' => 'eloquent',

'model' => App\User::class,

],

//添加后台用户验证及数据模型

'admins' => [

'driver' => 'eloquent',

'model' => App\Admin::class,

],

添加数据模型,并通过工厂模式填充数据,然后迁移数据.这里略.

修改Admin数据模型

namespace App;

use Illuminate\Notifications\Notifiable;

use Illuminate\Foundation\Auth\User as Authenticatable;

//启用auth功能依赖上面两个

use Illuminate\Database\Eloquent\Model;

class Vip\_users extends Authenticatable

{

use Notifiable;//启用auth功能依赖

//自动填充

protected $fillable = [

'name', 'email', 'password',

];

/\*\*

\* The attributes that should be hidden for arrays.

\*

\* @var array

\*/

//黑名单

protected $hidden = [

'password', 'remember\_token',

];

}

注意和普通数据模型不一样,需要引入两个依赖包

这里引入依赖包和定义了自动填充以及黑名单属性.

创建控制器,最基本的两个login和home

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;

use Illuminate\Foundation\Auth\AuthenticatesUsers;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

class LoginController extends Controller

{

use AuthenticatesUsers;

protected $redirectTo = '/admin/home';//登陆后位置

protected $redirectAfterLogout='/';

protected $username;

public function \_\_construct()

{

//定义此控制器的中间件

//guest:admin 是因为RegisterController文件指定为guest

$this->middleware('guest:admin', ['except' => 'logout']);//except 除了,及除了退出以外都会用到此中间件

$this->username = config('admin.global.username');

}

//定义视图

public function showLoginForm()

{

return view('admin.login.index');

}

//自定义认证驱动,及指定auth配置文件下guard的哪一个

protected function guard()

{

//名词解释 auth 认证,guard 警卫,护卫,guest 客人,except 除外,除了

return auth()->guard('admin');

}

public function logout(){

//Auth::check()获取失败,必须指定guard admin 默认guard是web

if(Auth('admin')->check()){

Auth('admin')->logout();

}

return redirect('/admin/login');

}

}

guard方法为定义认证驱动, 及指定到配置文件中添加的guard

构造器中指定中间件middleware,其中不包括logout

可以重写logout方法

namespace App\Http\Controllers\Admin;

use Illuminate\Http\Request;

use App\Http\Controllers\Controller;

class HomeController extends Controller

{

/\*\*

\* Create a new controller instance.

\*

\* @return void

\*/

public function \_\_construct()

{

$this->middleware('auth.admin:admin');

}

/\*\*

\* Show the application dashboard.

\*

\* @return \Illuminate\Http\Response

\*/

public function index()

{

return view('admin.home');

}

public function haha(){

dd('后台首页，当前用户名：'.auth('admin')->user()->name);

}

}

由于现在除了默认user外还有admin在使用auth,这里要修改登录后跳转的位置

app\Http\Middleware\RedirectIfAuthenticated.php下:

public function handle($request, Closure $next, $guard = null)

{

if (Auth::guard($guard)->check()) {

//不同的guard调转到不同的页面

$url = $guard ? 'admin/home':'/home';

return redirect($url);

}

return $next($request);

}

Home中使用到中间件,目的是维持登录判断,定义中间件,

php artisan make:middleware AdminAuthMiddleware

然后编辑AdminAuthMiddleware

namespace App\Http\Middleware;

use Closure;

use Illuminate\Support\Facades\Auth;

class AdminAuthMiddleware

{

/\*\*

\* Handle an incoming request.

\*

\* @param \Illuminate\Http\Request $request

\* @param \Closure $next

\* @return mixed

\*/

public function handle($request, Closure $next, $guard=null)

{

if (Auth::guard($guard)->guest()) {

if ($request->ajax() || $request->wantsJson()) {

return response('Unauthorized.', 401);

} else {

return redirect()->guest('admin/login');

}

}

return $next($request);

}

}

注册middleware:

//认证中间件定义

protected $routeMiddleware = [

'auth' => \Illuminate\Auth\Middleware\Authenticate::class,

'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,

'bindings' => \Illuminate\Routing\Middleware\SubstituteBindings::class,

'can' => \Illuminate\Auth\Middleware\Authorize::class,

'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,

'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,

//注册中间件

'auth.admin' => \App\Http\Middleware\AdminAuthMiddleware::class,

];

然后配置路由:

Route::group(['prefix' => 'admin','namespace' => 'Admin'],function ($router){

$router->get('/','HomeController@haha');

$router->get('login', 'LoginController@showLoginForm');

$router->post('login', 'LoginController@login');

$router->post('logout', 'LoginController@logout');

Route::get('home', 'HomeController@index');

});

最后是编辑视图文件即可访问.

