Laravel 5.5 不同用户表登录认证(前后台分离)

Auth 认证原理简述

Laravel 的认证是使用 guard 与 provider 配合完成， guard 负责认证的业务逻辑，认证信息的服务端保存等； provider 负责提供认证信息的持久化数据提供。

请求提交给 guard， guard 从 provider 里取出数据（类似用户名、密码等），验证输入数据与服务器端存储的数据是否吻合。如果提交的数据正确，再做 session 等业务的处理（如有需要）。

认证脚手架

首先我们导入 Laravel 的自带的认证脚手架

```
php artisan make:auth
```

执行数据库迁移：

```
php artisan migrate
```

修改 Auth 认证的配置文件 config/auth.php

在 gurads 处，添加 admin guard 用于后台管理员认证

```
    'guards' => [
        'web' => [
            'driver' => 'session',
            'provider' => 'users',
        ],

        'admin' => [
            'driver' => 'session',
            'provider' => 'admins',
        ],

        'api' => [
            'driver' => 'token',
            'provider' => 'users',
        ],
    ],
```

在 providers 处添加 admins provider，使用 Admin 模型

```
    'providers' => [
        'users' => [
            'driver' => 'eloquent',
            'model' => App\User::class,
        ],
        'admins' => [
            'driver' => 'eloquent',
            'model' => App\Admin::class,
        ],
    ],
```

创建后台管理员模型

我们再创建一个 Admin 模型，用于后台管理员登录验证。

```
php artisan make:model Admin -m
```

-m 参数会同时生成数据库迁移文件 xxxx_create_admins_table

修改 app/Admin.php 模型文件

```
<?php

namespace App;

use Illuminate\Notifications\Notifiable;
use Illuminate\Foundation\Auth\User as Authenticatable;

class Admin extends Authenticatable
{
    use Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name', 'password',
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];
}
```

编辑 xxxx_create_admins_table 文件，后台管理员模型结构与前台用户差不多，去掉 email 字段，name 字段设为 unique

```php
<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateAdminsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('admins', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name')->unique();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('admins');
    }
}
```

管理员模型填充数据

定义一个数据模型工厂，在 database/factories/ModelFactory.php 中添加如下代码

```
$factory->define(App\Admin::class, function (Faker\Generator $faker) {
    static $password;

    return [
        'name' => $faker->firstName,
        'password' => $password ?: $password = bcrypt('secret'),
        'remember_token' => str_random(10),
    ];
});
```

使用 Faker 随机填充用户名

在 database/seeds 目录下生成 AdminsTableSeeder.php 文件。

```
php artisan make:seeder AdminsTableSeeder
```

编辑 database/seeds/AdminsTableSeeder.php 文件的 run 方法，添加3个管理员用户，密码为 123456

```
    public function run()
    {
        factory('App\Admin', 3)->create([
            'password' => bcrypt('123456')
            ]);
    }
```

在 database/seeds/DatabaseSeeder.php 的 run 方法里调用 AdminsTableSeeder 类

```
    public function run()
    {
        $this->call(AdminsTableSeeder::class);
    }
```

执行数据库迁移命令

```
php artisan migrate --seed
```

数据库里会创建 admins 表，并且生成了3条数据

创建后台页面

创建控制器

```
php artisan make:controller Admin/LoginController
php artisan make:controller Admin/IndexController
```

其中， Admin/LoginController 负责登录逻辑； Admin/IndexController 管理登录后的首页。

编辑 Admin/LoginController.php

```
<?php

```

```php
namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Foundation\Auth\AuthenticatesUsers;

class LoginController extends Controller
{
    /*
    |--------------------------------------------------------------------------
    | Login Controller
    |--------------------------------------------------------------------------
    |
    | This controller handles authenticating users for the application and
    | redirecting them to your home screen. The controller uses a trait
    | to conveniently provide its functionality to your applications.
    |
    */

    use AuthenticatesUsers;

    /**
     * Where to redirect users after login / registration.
     *
     * @var string
     */
    protected $redirectTo = '/admin';

    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('guest.admin', ['except' => 'logout']);
    }

    /**
     * 显示后台登录模板
     */
    public function showLoginForm()
    {
        return view('admin.login');
    }

    /**
     * 使用 admin guard
     */
    protected function guard()
    {
        return auth()->guard('admin');
    }

    /**
     * 重写验证时使用的用户名字段
     */
    public function username()
    {
        return 'name';
    }
}
```

编辑 Admin/IndexController.php

```php
<?php

namespace App\Http\Controllers\Admin;

use Illuminate\Http\Request;

use App\Http\Requests;
use App\Http\Controllers\Controller;

class IndexController extends Controller
{
    /**
     * 显示后台管理模板首页
     */
    public function index()
    {
        return view('admin.index');
    }
}
```

后台显示模板

复制 views/layouts/app.blade.php 成 views/layouts/admin.blade.php

编辑后台管理布局模板

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- CSRF Token -->
    <meta name="csrf-token" content="{{ csrf_token() }}">

    <title>{{ config('app.name', 'Laravel') }} - Admin</title>

    <!-- Styles -->
    <link href="{{ asset('css/app.css') }}" rel="stylesheet">
</head>
<body>
<nav class="navbar navbar-default navbar-static-top">
    <div class="container">
        <div class="navbar-header">

            <!-- Collapsed Hamburger -->
            <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse"
                    data-target="#app-navbar-collapse">
                <span class="sr-only">Toggle Navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>

            <!-- Branding Image -->
            <a class="navbar-brand" href="{{ url('/') }}">
```

```
                {{ config('app.name', 'Laravel') }}
            </a>
        </div>

        <div class="collapse navbar-collapse" id="app-navbar-collapse">
            <!-- Left Side Of Navbar -->
            <ul class="nav navbar-nav">
                &nbsp;
            </ul>

            <!-- Right Side Of Navbar -->
            <ul class="nav navbar-nav navbar-right">
                <!-- Authentication Links -->
                @if (auth()->guard('admin')->guest())
                    <li><a href="{{ url('/admin/login') }}">Login</a></li>
                    {{--<li><a href="{{ route('register') }}">Register</a></li>-
-}}
                @else
                    <li class="dropdown">
                        <a href="#" class="dropdown-toggle" data-
toggle="dropdown" role="button"
                           aria-expanded="false" aria-haspopup="true">
                            {{ auth()->guard('admin')->user()->name }} <span 
class="caret"></span>
                        </a>

                        <ul class="dropdown-menu">
                            <li>
                                <a href="{{ url('/admin/logout')}}"
                                   onclick="event.preventDefault();
                                                 
document.getElementById('logout-form').submit();">
                                    Logout
                                </a>

                                <form id="logout-form" action="{{ 
url('/admin/logout')}}" method="POST"
                                      style="display: none;">
                                    {{ csrf_field() }}
                                </form>
                            </li>
                        </ul>
                    </li>
                @endif
            </ul>
        </div>
    </div>
</nav>

@yield('content')

<!-- Scripts -->
<script src="{{ asset('js/app.js') }}"></script>
</body>
</html>
```

复制 views/auth/login.blade.php 成 views/admin/login.blade.php
编辑该模板，更改布局文件为 layouts.admin， 把表单的提交 url 改为 admin/login，email 字段改成 name字段，去掉找回密码的部分

```
@extends('layouts.admin')

@section('content')
    <div class="container">
        <div class="row">
            <div class="col-md-8 col-md-offset-2">
                <div class="panel panel-default">
                    <div class="panel-heading">Admin Login</div>
                    <div class="panel-body">
                        <form class="form-horizontal" role="form" method="POST"
action="{{ url('/admin/login') }}">
                            {{ csrf_field() }}

                            <div class="form-group{{ $errors->has('name') ? '
has-error' : '' }}">
                                <label for="name" class="col-md-4 control-
label">Name</label>

                                <div class="col-md-6">
```

```
                                    <input id="name" type="text" class="form-
control" name="name" value="{{ old('name') }}" required autofocus>

                                    @if ($errors->has('name'))
                                        <span class="help-block">
                                        <strong>{{ $errors->first('name') }}
</strong>
                                    </span>
                                    @endif
                                </div>
                            </div>

                            <div class="form-group{{ $errors->has('password') ?
' has-error' : '' }}">
                                <label for="password" class="col-md-4 control-
label">Password</label>

                                <div class="col-md-6">
                                    <input id="password" type="password"
class="form-control" name="password" required>

                                    @if ($errors->has('password'))
                                        <span class="help-block">
                                        <strong>{{ $errors->first('password') }}
</strong>
                                    </span>
                                    @endif
                                </div>
                            </div>

                            <div class="form-group">
                                <div class="col-md-6 col-md-offset-4">
                                    <div class="checkbox">
                                        <label>
                                            <input type="checkbox"
name="remember"> Remember Me
                                        </label>
                                    </div>
                                </div>
                            </div>
                            <div class="form-group">
                                <div class="col-md-8 col-md-offset-4">
                                    <button type="submit" class="btn btn-primary">
                                        Login
                                    </button>
                                </div>
                            </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </div>
@endsection
```

复制 views/home.blade.php 成 views/admin/index.blade.php

编辑该模板

```
@extends('layouts.admin')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-8 col-md-offset-2">
            <div class="panel panel-default">
                <div class="panel-heading">Dashboard</div>

                <div class="panel-body">
                    You are logged in admin dashboard!
                </div>
            </div>
        </div>
    </div>
</div>
@endsection
```

添加后台路由

编辑 routes/web.php， 添加以下内容

```
Route::group(['prefix' => 'admin'], function () {
    Route::group(['middleware' => 'auth.admin'], function () {
        Route::get('/', 'Admin\IndexController@index');
    });

    Route::get('login', 'Admin\LoginController@showLoginForm')->name('admin.login');
    Route::post('login', 'Admin\LoginController@login');
    Route::post('logout', 'Admin\LoginController@logout');
});
```

后台管理认证中间件

创建后台管理认证中间件

```
php artisan make:middleware AuthAdmin
```

编辑 AuthAdmin

```
<?php

namespace App\Http\Middleware;

use Closure;

class AuthAdmin
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if (auth()->guard('admin')->guest()) {
            if ($request->ajax() || $request->wantsJson()) {
                return response('Unauthorized.', 401);
            } else {
                return redirect()->guest('admin/login');
            }
        }

        return $next($request);
    }
}
```

创建后台管理登录跳转中间件，用于有些操作在登录之后的跳转

```
php artisan make:middleware GuestAdmin
```

编辑该中间件的 handle 方法

```
    public function handle($request, Closure $next)
    {
        if (auth()->guard('admin')->check()) {
            return redirect('/admin');
        }

        return $next($request);
    }
```

在 app/Http/Kernel.php 中注册以上中间件

```
    protected $routeMiddleware = [
        ......
        'auth.admin' => \App\Http\Middleware\AuthAdmin::class,
        'guest.admin' => \App\Http\Middleware\GuestAdmin::class,
    ];
```

处理注销

经过上面的步骤，已经实现了前后台分离登录，但是不管是在前台注销，还是在后台注销，都销毁了所有的 session，导致前后台注销连在一起。所以我们还要对注销的方法处理一下。

原来的 logout 方法是这样写的，在 Illuminate\Foundation\Auth\AuthenticatesUsers 里

```
    public function logout(Request $request)
    {
        $this->guard()->logout();

        $request->session()->flush();

        $request->session()->regenerate();

        return redirect('/');
    }
```

注意这一句

```
$request->session()->flush();
```

将所有的 session 全部清除，这里不分前台、后台，所以要对这里进行改造。
因为前台、后台注销都要修改，所以我们新建一个 trait，前后台都可以使用。
新建一个文件 app/Extensions/AuthenticatesLogout.php

```php
<?php
namespace App\Extensions;

use Illuminate\Http\Request;


trait AuthenticatesLogout
{
    public function logout(Request $request)
    {
        $this->guard()->logout();

        $request->session()->forget($this->guard()->getName());

        $request->session()->regenerate();

        return redirect('/');
    }
}
```

我们将上面的那一句改成

```php
$request->session()->forget($this->guard()->getName());
```

只是删除掉当前 guard 所创建的 session，这样就达到了分别注销的目的。
修改 Auth/LoginController.php 和 Admin/LoginController.php，将

```php
class LoginController extends Controller
{
    use AuthenticatesUsers;
```

改掉，在文件的前面别忘了加上 use 语句

```php
use App\Extensions\AuthenticatesLogout;

...

class LoginController extends Controller
{
    use AuthenticatesUsers, AuthenticatesLogout {
        AuthenticatesLogout::logout insteadof AuthenticatesUsers;
    }
...
```

到这里，就完成了整个不同用户表登录认证的过程。

参考Laravel5.3多用户表登录