






LARAVEL FRAMEWORK

BÀI 7: AUTHENTICATION




- ⊙ Tổng quan Authentication
- ⊙ Cấu hình sử dụng Auth
- ⊙ HTTP Basic Authentication



Phần I: Tổng quan Authentication

-  Giới thiệu Auth
-  Các thành phần trong Auth
-  Cấu hình sử dụng Auth

Phần II: HTTP Basic Authentication

-  Adding Custom Guards
-  Adding Custom User Providers
-  Events



- ❑ Authentication là hành động xác thực một cái gì đó có được hoạt động tiếp không hay dừng lại.



- ❑ Việc xác thực trong Laravel được dựng sẵn trong framework
- ❑ Tập tin chứa các hành động và tùy chọn cho việc xác thực config/auth.php
 - ❖ model name
 - ❖ users table
 - ❖ Password
 - ❖ Table
 - ❖ Driver

```
| -----  
|  Default Authentication Driver  
| -----  
| /*  
| 'driver' => 'eloquent',  
| /*  
| -----  
|  Authentication Model  
| -----  
| /*  
| 'model' => App\User::class,  
| /*  
| -----  
|  Authentication Table  
| -----  
| /*  
| 'table' => 'users',  
| /*  
| -----  
|  Password Reset Settings  
| -----  
| /*  
| 'password' => [  
|     'email' => 'emails.password',  
|     'table' => 'password_resets',  
|     'expire' => 60,  
| ],
```

- ❑ Laravel xây dựng sẵn các authentication controller tại `App\Http\Controllers\Auth` namespace
 - ❖ RegisterController: xác thực đăng ký mới thành viên
 - ❖ LoginController: xác thực thành viên đăng nhập
 - ❖ ForgotPasswordController: cung cấp xác thực phục hồi mật khẩu thông qua email link
- ❑ Các controller này đã cung cấp đầy đủ các method phục vụ tốt các nghiệp vụ liên quan

❑ Routing

- ❖ Laravel cung cấp lệnh `make:auth` chạy thông qua `artisan`: `php artisan make:auth`
- ❖ Sau khi run thành công thì Laravel phát sinh đầy đủ các controller, view, và route file cần thiết cho việc xác thực
- ❖ Tại HomeController cũng sinh ra các yêu cầu xác thực khi truy cập ứng dụng

```
public function __construct()  
{  
    $this->middleware('auth');  
}
```

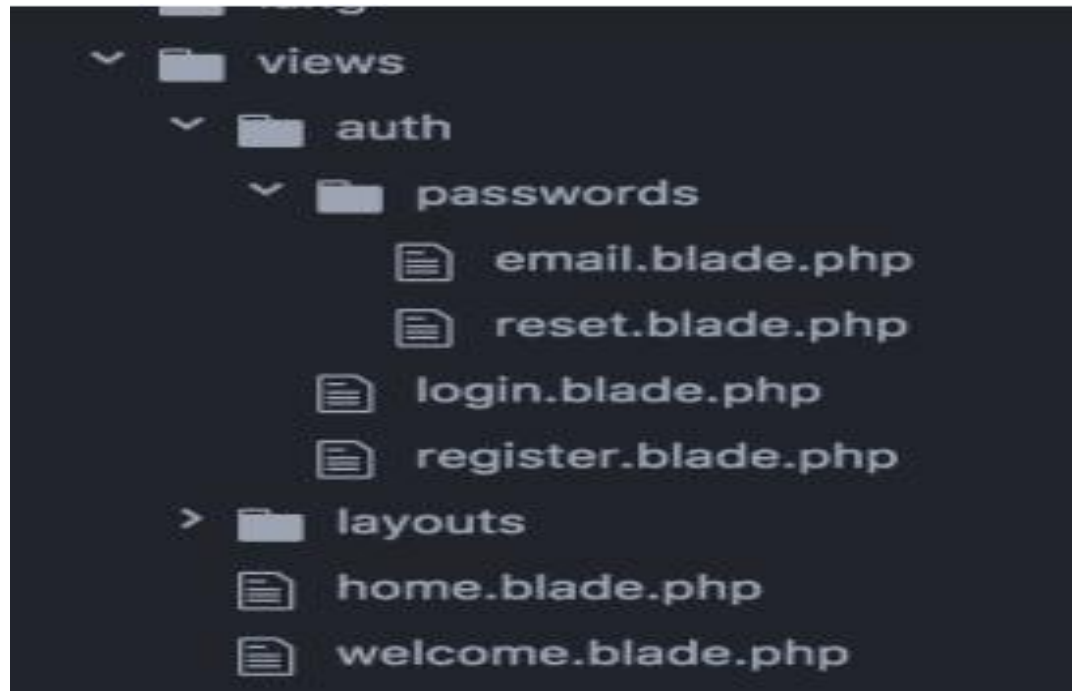

□ Routing

- ❖ Tại (resources > routes > web.php) sẽ tạo phương thức Auth::routes();

```
$this->get('login', 'Auth\LoginController@showLoginForm')->name('login');  
$this->post('login', 'Auth\LoginController@login');  
$this->post('logout', 'Auth\LoginController@logout')->name('logout');  
  
// Registration Routes...  
$this->get('register', 'Auth\RegisterController@showRegistrationForm')->name('register');  
$this->post('register', 'Auth\RegisterController@register');  
  
// Password Reset Routes...  
$this->get('password/reset', 'Auth\ForgotPasswordController@showLinkRequestForm');  
$this->post('password/email', 'Auth\ForgotPasswordController@sendResetLinkEmail');  
$this->get('password/reset/{token}', 'Auth\ResetPasswordController@showResetForm');  
$this->post('password/reset', 'Auth\ResetPasswordController@reset');
```

□ View

- ❖ Các View cũng được sinh ra trong thư mục `resources/views/auth`
- ❖ Thư mục `resources/views/layouts` chứa các layout cơ bản của ứng dụng



❑ Authenticating

- ❖ Sau khi setup routes, view, controller thì đã sẵn sàng cho việc xác thực đăng ký mới, đăng nhập, quên mật khẩu...
- ❖ Mặc định sau khi xác thực thành công thì sẽ chuyển trang về /home uri
- ❖ Có thể điều chỉnh chuyển trang thông qua thuộc tính redirectTo tại LoginController, RegisterController, and ResetPasswordController

```
protected $redirectTo = '/';
```

```
protected function redirectTo()  
{  
    return '/path';  
}
```

❑ Username Customization

- ❖ Mặc định Laravel dùng Email field cho việc xác thực, chúng ta có thể thay đổi bằng cách định nghĩa phương thức username bên trong LoginController

```
public function username()  
{  
    return 'username';  
}
```

❑ Retrieving The Authenticated User

- ❖ Có thể truy cập xác thực user thông qua Auth

```
use Illuminate\Support\Facades\Auth;

// Get the currently authenticated user...
$user = Auth::user();

// Get the currently authenticated user's ID...
$id = Auth::id();
```

❑ Protecting Routes

❖ Sử dụng Route middleware tại route:

```
Route::get('profile', function () {  
    // Only authenticated users may enter...  
})->middleware('auth');
```

❖ Sử dụng Route middleware tại controller

```
public function __construct()  
{  
    $this->middleware('auth');  
}
```



DEMO

Demo xác thực đăng ký mới user





LARAVEL FRAMEWORK

BÀI 7: AUTHENTICATION

- ❑ HTTP Basic Authentication cung cấp cách xác thực không cần sử dụng login page.
- ❑ Khai báo sử dụng `auth.basic` middleware trong route

```
Route::get('profile', function () {  
    // Only authenticated users may enter...  
})->middleware('auth.basic');
```

- ❑ Mặc định `auth.basic` middleware sử dụng cột Email xác thực

- ❑ Trường hợp sử dụng PHP FastCGI, HTTP thì thêm đoạn code sau trong .htaccess

```
RewriteCond %{HTTP:Authorization} ^(.+)$  
RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]
```

❑ Thêm Custom Guards

- ❖ Có thể thêm các lựa chọn xác thực bằng cách kế thừa các phương thức của Auth

```
use App\Services\Auth\JwtGuard;
use Illuminate\Support\Facades\Auth;
use Illuminate\Foundation\Support\Providers\AuthServiceProvider as ServiceProvider;

class AuthServiceProvider extends ServiceProvider
{
    /**
     * Register any application authentication / authorization services.
     *
     * @return void
     */
    public function boot()
    {
        $this->registerPolicies();

        Auth::extend('jwt', function ($app, $name, array $config) {
            // Return an instance of Illuminate\Contracts\Auth\Guard...

            return new JwtGuard(Auth::createUserProvider($config['provider']));
        });
    }
}
```

□ Adding Custom User Providers

```
<?php

namespace App\Providers;

use Illuminate\Support\Facades\Auth;
use App\Extensions\RiakUserProvider;
use Illuminate\Foundation\Support\Providers\AuthServiceProvider as ServiceProvider;

class AuthServiceProvider extends ServiceProvider
{
    public function boot()
    {
        $this->registerPolicies();

        Auth::provider('riak', function ($app, array $config) {
            // Return an instance of Illuminate\Contracts\Auth\UserProvider...

            return new RiakUserProvider($app->make('riak.connection'));
        });
    }
}
```

❑ Adding Custom User Providers

- ❖ Sau khi đăng ký thành công Provider, chúng ta sử dụng provider mới trong auth.php, cần định nghĩa provider:

```
'providers' => [  
    'users' => [  
        'driver' => 'riak',  
    ],  
],
```

Và sử dụng trong file cấu hình guards

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

❑ Sự kiện

- ❖ Trong tiến trình xác thực thì Laravel phát sinh nhiều sự kiện, chúng ta có thể thiết lập cơ chế lắng nghe và xử lý các sự kiện trong EventServiceProvider

```
protected $listen = [  
    'Illuminate\Auth\Events\Registered' => [  
        'App\Listeners\LogRegisteredUser',  
    ],  
  
    'Illuminate\Auth\Events\Attempting' => [  
        'App\Listeners\LogAuthenticationAttempt',  
    ],  
  
    'Illuminate\Auth\Events\Authenticated' => [  
        'App\Listeners\LogAuthenticated',  
    ],  
  
    'Illuminate\Auth\Events>Login' => [  
        'App\Listeners\LogSuccessfulLogin',  
    ],
```

```
'Illuminate\Auth\Events\Failed' => [  
    'App\Listeners\LogFailedLogin',  
],  
  
'Illuminate\Auth\Events\Logout' => [  
    'App\Listeners\LogSuccessfulLogout',  
],  
  
'Illuminate\Auth\Events\Lockout' => [  
    'App\Listeners\LogLockout',  
],  
  
'Illuminate\Auth\Events>PasswordReset' => [  
    'App\Listeners\LogPasswordReset',  
],
```



DEMO

-Demo xác thực Change Password



Phần I: Tổng quan Authentication

 Giới thiệu Auth

 Các thành phần trong Auth

 Cấu hình sử dụng Auth

Phần II: HTTP Basic Authentication

 Adding Custom Guards

 Adding Custom User Providers

 Events





Cảm ơn