

Authentication Laravel

Authentication trong Laravel hỗ trợ các tính năng liên quan đến xác thực người dùng: Đăng ký, đăng nhập, quên mật khẩu, kích hoạt tài khoản,...

Các thiết lập cấu hình trong Laravel Authentication

Laravel đã tạo sẵn các Controller, View, Middleware, Model... và cả file cấu hình config/auth.php để chúng ta chỉ cần thiết lập là có thể chạy được ngay.

Để thiết lập, chúng ta cấu hình trong file `config/auth.php`

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

    'api' => [
        'driver' => 'token',
        'provider' => 'users',
    ],
],
'providers' => [
    'users' => [
        'driver' => 'eloquent',
        'model' => App\User::class,
    ],

    // 'users' => [
    //     'driver' => 'database',
    //     'table' => 'users',
    // ],
],
```

Trong file này có hai phần cần quan tâm là guards và providers

- **Guards** định nghĩa cách thức xác thực người dùng cho các request, mặc định guards sử dụng Laravel Session để lưu trữ các thông tin trạng thái
- **Providers** định nghĩa các cách thức lưu trữ và lấy dữ liệu để xác thực người dùng.

Laravel hỗ trợ hai cách thức là Laravel Eloquent ORM hoặc Laravel Query Builder để tương tác với CSDL trong xác thực, ngoài ra nó cũng cho phép định nghĩa riêng các **providers** của chính bạn để sử dụng, tuy nhiên chúng ta có lựa chọn các cách thức mặc định này là đã đáp ứng rất tốt cho ứng dụng web

Model, View, Controller tạo sẵn cho Laravel Authentication

Mặc định một Eloquent Model `app\User.php` được tạo ra, model này được sử dụng mặc định trong file cấu hình `config/auth.php`

Có một số chú ý khi tạo bảng users tương ứng trong CSDL:

- cột password phải có độ dài ít nhất là 60 ký tự
- phải có cột remember_token với độ dài 100 ký tự và có thể null, cột này sử dụng cho tùy chọn nhớ xác thực trong form xác thực

Laravel cũng tạo ra một loạt các Controller trong `app\Http\Controllers\Auth` như:

- RegisterController: Quản lý việc đăng ký thành viên mới.
- LoginController: Quản lý việc đăng nhập các thành viên.
- ForgotPasswordController: gửi email với đường link sử dụng cho reset password.
- ResetPasswordController: kiểm soát việc reset mật khẩu với các logic do lập trình viên thêm vào.

Với phần lớn các ứng dụng web, các Controller ở trên không cần thay đổi gì. Các view sử dụng trong Laravel Authentication cũng được tạo ra sẵn nằm trong thư mục `resources/views/auth`.

Cài đặt Laravel Authentication

Để cài đặt, chúng ta sử dụng các câu lệnh sau:

```
composer require laravel/ui
```

```
php artisan ui:auth
```

Tiếp theo, hãy làm theo hướng dẫn của giao diện dòng lệnh để hoàn thiện.

Tùy biến Path trong controller

Chúng ta có thể thiết lập Path sau khi hoàn thiện việc: đăng nhập, đăng ký, đăng xuất,... bằng cách điều chỉnh thuộc tính `$redirectTo` trong các Controller tương ứng

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

Hoặc điều chỉnh trong Middleware `RedirectIfAuthenticated`

```
public function handle($request, Closure $next, $guard = null)
{
    if (Auth::guard($guard)->check()) {
        return redirect('/home');
    }

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

Tùy biến Username

Mặc định Laravel sẽ dùng email để đăng nhập, chúng ta có thể thay đổi bằng cách thêm phương thức sau trong `LoginController`

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

Tùy biến Validator, Store

Trên thực tế có nhiều trường hợp muốn thêm các trường trong users khi đăng ký, để tùy biến hãy bổ sung vào RegisterController phương thức sau:

```
protected function validator(array $data)
{
    return Validator::make();
}

protected function create(array $data)
{
    return User::create();
}
```

Truy xuất người dùng đã xác thực

```
$id = Auth::user()->id; hoặc Auth::id() //Lấy về ID người
$user = Auth::user() // Lấy về tất cả các thông tin của người dùng
$email = Auth::user()->email // Lấy về email người dùng

//kiểm tra người dùng đăng nhập
if (Auth::check()) {
    echo "Nguoi dung da dang nhap he thong";
}
```

```
}
```

Middleware

Mặc định, Laravel đã có sẵn Middleware **auth**, bạn chỉ cần gọi nó ra là có thể sử dụng

```
Route::get('thong-tin-ca-nhan', function () {  
    //Code here  
})->middleware('auth');
```

Ngoài ra, chúng ta có thể chỉ định **guard** khi gọi Middleware

```
Route::get('thong-tin-ca-nhan', function () {  
    //Code here  
})->middleware('auth:api');
```

Xác thực người dùng thủ công

Để xác thực thủ công, chúng ta sử dụng phương thức `Auth::attempt()`

```
if (Auth::attempt(['email' => $email, 'password' =>  
$password])) {  
    //Thành công  
}
```

Ngoài ra, chúng ta có thể đăng xuất bằng phương thức **`Auth::logout()`**