Course - Laravel Framework

Khôi phục mật khẩu

Hầu hết các ứng dụng web đều cung cấp tính năng để người dùng khôi phục lại mật khẩu đã quên của họ. Thay vì bạn phải thực hiện tính năng này bằng tay cho mọi ứng dụng bạn tạo, Laravel cung cấp các service tiện lợi có sẵn để gửi liên kết khôi phục lại mật khẩu.

Tags: reseting password, reset password, Khôi phục mật khẩu

Giới thiệu

Hầu hết các ứng dụng web đều cung cấp tính năng để người dùng khôi phục lại mật khẩu đã quên của họ. Thay vì bạn phải thực hiện tính năng này bằng tay cho mọi ứng dụng bạn tạo, Laravel cung cấp các service tiện lợi có sẵn để gửi liên kết khôi phục lại mật khẩu.

> Bạn muốn bắt đầu nhanh? Cài đặt bộ khởi động nhanh ứng dụng Laravel trong ứng dụng Laravel mới. Bộ công cụ khởi động nhanh của Laravel sẽ chăm sóc toàn bộ hệ thống xác thực của bạn, bao gồm cả việc khôi phục lại mật khẩu đã quên.

Chuẩn bị model

Trước khi sử dụng các tính năng khôi phục lại mật khẩu của Laravel, model `App\Models\User` của ứng dụng của bạn phải sử dụng trait `Illuminate\Notifications\Notifiable`. Thông thường, trait này đã được đưa vào bên trong model `App\Models\User` theo mặc định trong các ứng dụng Laravel mới.

Tiếp theo, việc xác minh rằng model `App\Models\User` của bạn triển khai contract `Illuminate\Contracts\Auth\CanResetPassword`. Model `App\Models\User` đi kèm với framework đã triển khai interface này và sử dụng trait `Illuminate\Auth\Passwords\CanResetPassword` để đưa vào các phương thức cần thiết để thực thi interface.

Chuẩn bị CSDL

Phải có riêng một bảng dữ liệu để lưu trữ mã token cho việc khôi phục mật khẩu của bạn. Một bản migration dùng tạo bảng dữ liệu này đã được đưa sẵn vào trong ứng dụng Laravel mặc định, vì vậy bạn chỉ cần chạy lệnh migrate cơ sở dữ liệu của mình để tạo bảng này:

```
php artisan migrate
```

Cấu hình Trusted Hosts

Theo mặc định, Laravel sẽ phản hồi tất cả các yêu cầu mà nó nhận được bất kể nội dung của header Host của yêu cầu HTTP request. Ngoài ra, giá trị của header Host sẽ được sử dụng khi tạo URL tuyệt đối cho ứng dụng của bạn trong một web request.

Thông thường, bạn nên cấu hình máy chủ web của mình, chẳng hạn như Nginx hoặc Apache, để chỉ gửi các request đến ứng dụng nào đó của bạn thống nhất với tên một máy

chủ cụ thể. Tuy nhiên, nếu bạn không có khả năng tùy chỉnh trực tiếp máy chủ web của mình và cần cho Laravel biết chỉ phản hồi với một số tên máy chủ nhất định, thì bạn có thể làm như vậy bằng cách bật middleware `App\Http\Middleware\TrustHosts` cho ứng dụng của mình. Điều này đặc biệt quan trọng khi ứng dụng của bạn cung cấp chức năng khôi phục mật khẩu.

Để tìm hiểu thêm về middleware này, vui lòng tham khảo tài liệu về middleware TrustHosts.

Routing

Để triển khai đúng cách sự hỗ trợ mà cho phép người dùng khôi phục lại mật khẩu của họ, chúng tôi sẽ cần xác định một số route. Đầu tiên, chúng tôi sẽ cần một cặp route để xử lý, nó cho phép người dùng yêu cầu một đường link khôi phục lại mật khẩu qua địa chỉ email của họ. Thứ hai, chúng tôi sẽ cần một cặp route để thực sự xử lý việc khôi phục lại mật khẩu khi người dùng truy cập đường link khôi phục lại mật khẩu được gửi qua email cho họ và hoàn thành biểu mẫu khôi phục lại mật khẩu.

Đưa ra yêu cầu đường link

Form yêu cầu đường link khôi phục mật khẩu

Đầu tiên, chúng tôi sẽ xác định các route cần thiết để yêu cầu các đường link khôi phục lại mật khẩu. Để bắt đầu, chúng ta sẽ tạo một route trả về một giao diện hiển thị với form yêu cầu đường link khôi phục lại mật khẩu:

```
Route::get('/forgot-password', function () {
    return view('auth.forgot-password');
})->middleware('guest')->name('password.request');
```

Giao diện hiển thị được trả về bởi route này phải có một biểu mẫu form chứa trường email, cho phép người dùng yêu cầu đường link khôi phục lại mật khẩu cho một địa chỉ email mà bạn muốn.

Xử lý việc đệ trình biểu mẫu form

Tiếp theo, chúng tôi sẽ xác định một route xử lý yêu cầu đệ trình biểu mẫu form từ giao diện "quên mật khẩu". Route này sẽ chịu trách nhiệm việc xác thực địa chỉ email và gửi lại

đường link khôi phục mật khẩu cho người dùng tương ứng:

```
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Password;

Route::post('/forgot-password', function (Request $request) {
  $request->validate(['email' => 'required|email']);

  $status = Password::sendResetLink(
    $request->only('email')
  );

  return $status === Password::RESET_LINK_SENT
    ? back()->with(['status' => __($status)])
    : back()->withErrors(['email' => __($status)]);
})->middleware('guest')->name('password.email');
```

Trước khi tiếp tục, chúng ta hãy xem xét route này chi tiết hơn. Đầu tiên, thuộc tính email trong request được xác thực. Tiếp theo, chúng tôi sẽ sử dụng "nhà môi giới mật khẩu" được tích hợp sẵn của Laravel (thông qua facade Password) để gửi lại đường link khôi phục mật khẩu cho người dùng. Nhà môi giới mật khẩu sẽ đảm nhận việc truy xuất người dùng theo trường yêu cầu (trong trường hợp này là địa chỉ email) và gửi cho người dùng đường link khôi phục mật khẩu thông qua hệ thống mail thông báo của Laravel.

Phương thức `sendResetLink` sẽ trả về một slug "status". Status này có thể được thông dịch bằng cách sử dụng các tính năng thông dịch ngôn ngữ của Laravel để hiển thị thông báo có vẻ quen thuộc với người dùng về trạng thái được phản hồi họ. Bản dịch của trạng thái khôi phục lại mật khẩu được xác định bởi tập tin ngôn ngữ *resources/lang/{lang}/passwords.php* trong ứng dụng của bạn. Mục giá trị của slug trạng thái nằm trong tập tin ngôn ngữ *passwords*.

Bạn có thể tự hỏi làm thế nào Laravel biết cách truy xuất record người dùng từ cơ sở dữ liệu ứng dụng của bạn khi gọi phương thức `sendResetLink` của facade Password. Nhà môi giới mật khẩu Laravel sử dụng "user providers" của hệ thống xác thực của bạn để truy xuất các record trong cơ sở dữ liệu. User provider mà được nhà môi giới mật khẩu sử dụng được cấu hình trong mảng cấu hình `passwords` của tập tin cấu hình *config/auth.php* của bạn. Để tìm hiểu thêm về cách viết user provider, hãy tham khảo tài liệu xác thực người dùng.

> Khi thực hiện đặt lại mật khẩu theo cách thủ công, bạn bắt buộc phải tự xác định nội

dung của các khung nhìn và tuyến đường. Nếu bạn muốn giàn giáo bao gồm tất cả logic xác thực và xác minh cần thiết, hãy xem bộ công cụ khởi động ứng dụng Laravel.

Khôi phục mật khẩu

Form khôi phục mật khẩu

Tiếp theo, chúng tôi sẽ xác định các route cần thiết để thực sự khôi phục lại mật khẩu khi người dùng nhấp vào liên kết đặt lại mật khẩu đã được gửi qua email cho họ và cung cấp mật khẩu mới. Đầu tiên, hãy xác định route sẽ hiển thị biểu mẫu form khôi phục lại mật khẩu được hiển thị khi người dùng nhấp vào đường link khôi phục mật khẩu. Route này sẽ nhận được một tham số mã token mà chúng ta sẽ sử dụng sau này để xác minh yêu cầu khôi phục mật khẩu:

```
Route::get('/reset-password/{token}', function ($token) {
  return view('auth.reset-password', ['token' => $token]);
})->middleware('guest')->name('password.reset');
```

Giao diện hiển thị được trả về bởi route này sẽ hiển thị một biểu mẫu form chứa trường `email`, trường `password`, trường `password_confirmation` và trường mã token bị ẩn giấu, nó chứa giá trị của mã bí mật $token mà route của chúng ta sẽ nhận được.

Xử lý đệ trình form

Tất nhiên, chúng ta cần xác định một route để thực sự xử lý việc gửi biểu mẫu khôi phục lại mật khẩu. Route sẽ chịu trách nhiệm xác thực yêu cầu được gửi đến và cập nhật mật khẩu của người dùng trong cơ sở dữ liệu:

```
use Illuminate\Auth\Events\PasswordReset;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Password;
use Illuminate\Support\Str;

Route::post('/reset-password', function (Request $request) {
  $request->validate([
```

```
        'token' => 'required',
        'email' => 'required|email',
        'password' => 'required|min:8|confirmed',
    ]);

    $status = Password::reset(
        $request->only('email', 'password', 'password_confirmation', 'token'),
        function ($user, $password) {
            $user->forceFill([
                'password' => Hash::make($password)
            ])->setRememberToken(Str::random(60));

            $user->save();

            event(new PasswordReset($user));
        }
    );

    return $status === Password::PASSWORD_RESET
        ? redirect()->route('login')->with('status', __($status))
        : back()->withErrors(['email' => [__($status)]]);
})->middleware('guest')->name('password.update');
```

Trước khi tiếp tục, chúng ta hãy xem xét route này chi tiết hơn. Đầu tiên, xác định các thuộc tính mã `token`, `email` và `password` của yêu cầu đã được xác thực. Tiếp theo, chúng ta sẽ sử dụng "nhà môi giới mật khẩu" được tích hợp sẵn của Laravel (thông qua facade Password) để xác thực thông tin đăng nhập yêu cầu khôi phục mật khẩu.

Nếu mã token, địa chỉ email và mật khẩu được cung cấp cho nhà môi giới mật khẩu là hợp lệ, thì hàm xử lý được truyền vào cho phương thức reset sẽ được gọi. Trong hàm xử lý, nơi nhận được đối tượng người dùng và mật khẩu trên văn bản thuần túy sẽ được cung cấp cho biểu mẫu form khôi phục mật khẩu, chúng tôi có thể cập nhật mật khẩu của người dùng trong cơ sở dữ liệu.

Phương thức reset trả về một slug "status". Status này có thể được thông dịch bằng cách sử dụng hàm thông dịch ngôn ngữ của Laravel để hiển thị thông báo quen thuộc với người dùng về trạng thái được phản hồi cho họ. Bản dịch của status khôi phục mật khẩu được xác định bởi tập tin ngôn ngữ *resources/lang/{lang}/passwords.php* của ứng dụng của bạn. Mục giá trị của slug trạng thái nằm trong tập tin ngôn ngữ *passwords*.

Trước khi tiếp tục, bạn có thể tự hỏi làm thế nào Laravel biết cách truy xuất record người dùng từ cơ sở dữ liệu của ứng dụng của bạn khi gọi phương thức `reset` của facade `Password`. Nhà môi giới mật khẩu Laravel sử dụng "user providers" của hệ thống xác thực của bạn để truy xuất các record trong cơ sở dữ liệu. User provider được nhà môi giới mật khẩu sử dụng được cấu hình trong mảng cấu hình `passwords` của tập tin cấu hình *config/auth.php* của bạn. Để tìm hiểu thêm về cách tự tạo user provider, hãy tham khảo tài liệu xác thực người dùng.

Xóa các token đáo hạn

Mã token khôi phục mật khẩu đã hết hạn sẽ vẫn còn trong cơ sở dữ liệu của bạn. Tuy nhiên, bạn có thể dễ dàng xóa các record này bằng lệnh Artisan `auth:clear-resets`:

```
php artisan auth:clear-resets
```

Nếu bạn muốn tự động hóa quá trình này, hãy xem xét việc thêm lệnh vào schedule của ứng dụng:

```
$schedule->command('auth:clear-resets')->everyFifteenMinutes();
```

Tự tạo

Tự tạo đường link reset

Bạn có thể tự tạo URL khôi phục mật khẩu bằng phương thức `createUrlUsing` được cung cấp bởi class thông báo `ResetPassword`. Phương thức này chấp nhận một hàm xử lý nhận đối tượng người dùng đang nhận được thông báo cũng như mã token khôi phục mật khẩu. Thông thường, bạn nên gọi phương thức này từ phương thức boot của service provider `App\Providers\AuthServiceProvider`:

```
use Illuminate\Auth\Notifications\ResetPassword;

/**
 * Register any authentication / authorization services.
 *
 * @return void
 */
```

```
public function boot()
{
  $this->registerPolicies();

  ResetPassword::createUrlUsing(function ($user, string $token) {
    return 'https://example.com/reset-password?token='.$token;
  });
}
```

Tự tạo email reset

Bạn có thể dễ dàng chỉnh sửa class thông báo được sử dụng để gửi đường link khôi phục mật khẩu cho người dùng. Để bắt đầu, hãy ghi đè phương thức `sendPasswordResetNotification` trên model `App\Models\User` của bạn. Trong phương pháp này, bạn có thể gửi thông báo bằng cách sử dụng bất kỳ class thông báo nào do bạn tạo ra. Mã `$token` khôi phục mật khẩu là đối số đầu tiên mà phương thức nhận được. Bạn có thể sử dụng biến `$token` này để tạo URL khôi phục mật khẩu mà bạn muốn và gửi thông báo của bạn cho người dùng:

```
use App\Notifications\ResetPasswordNotification;

/**
 * Send a password reset notification to the user.
 *
 * @param  string  $token
 * @return void
 */
public function sendPasswordResetNotification($token)
{
  $url = 'https://example.com/reset-password?token='.$token;

  $this->notify(new ResetPasswordNotification($url));
}
```