Course - Laravel Framework

Mã hóa

Các service mã hóa của Laravel cung cấp một interface đơn giản, thuận tiện để mã hóa và giải mã văn bản qua OpenSSL bằng cách sử dụng mã hóa AES-256 và AES-128. Tất cả các giá trị được mã hóa của Laravel đều được đánh dấu bằng mã xác thực tin nhắn (MAC) để giá trị cơ bản của chúng không thể bị sửa đổi hoặc giả mạo sau khi được mã hóa.

Tags: encrytion, ma hoa, laravel

Giới thiệu

Các service mã hóa của Laravel cung cấp một interface đơn giản, thuận tiện để mã hóa và giải mã văn bản qua OpenSSL bằng cách sử dụng mã hóa AES-256 và AES-128. Tất cả các giá trị được mã hóa của Laravel đều được đánh dấu bằng mã xác thực tin nhắn (MAC) để giá trị cơ bản của chúng không thể bị sửa đổi hoặc giả mạo sau khi được mã hóa.

Cấu hình

Trước khi sử dụng chương trình mã hóa của Laravel, bạn phải cài đặt tùy chọn cấu hình kiểu khóa trong tập tin cấu hình *config/app.php* của mình. Giá trị cấu hình này được điều khiển bởi biến môi trường `APP_KEY`. Bạn nên sử dụng lệnh php Artian `key:generate` để tạo giá trị của biến này vì lệnh `key:generate` sẽ sử dụng trình tạo byte ngẫu nhiên an toàn của PHP để tạo khóa an toàn bằng mật mã cho ứng dụng của bạn. Thông thường, giá trị của biến môi trường `APP_KEY` sẽ được tạo cho bạn trong quá trình cài đặt Laravel.

Sử dụng chức năng mã hóa

Mã hóa một giá trị

Bạn có thể mã hóa một giá trị bằng phương thức `encryptString` được cung cấp bởi facade Crypt. Tất cả các giá trị được mã hóa đều được mã hóa bằng OpenSSL và mật mã AES-256-CBC. Hơn nữa, tất cả các giá trị được mã hóa đều được đánh dấu bằng mã xác thực tin nhắn (MAC). Mã xác thực tin nhắn tích hợp sẵn sẽ ngăn chặn việc giải mã bất kỳ giá trị nào đã bị giả mạo bởi người dùng độc hại:

```php
<?php

namespace App\Http\Controllers;

use App\Http\Controllers\Controller;
use App\Models\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Crypt;

class DigitalOceanTokenController extends Controller
{
  /**
```

```
     * Store a DigitalOcean API token for the user.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return \Illuminate\Http\Response
     */
    public function storeSecret(Request $request)
    {
        $request->user()->fill([
            'token' => Crypt::encryptString($request->token),
        ])->save();
    }
}
```

Giải mã một giá trị

Bạn có thể giải mã các giá trị bằng phương thức `decryptString` được cung cấp bởi facade Crypt. Nếu giá trị không thể được giải mã đúng cách, chẳng hạn như khi mã xác thực tin nhắn không hợp lệ, `Illuminate\Contracts\Encryption\DecryptException` sẽ được đưa ra:

```
use Illuminate\Contracts\Encryption\DecryptException;
use Illuminate\Support\Facades\Crypt;

try {
    $decrypted = Crypt::decryptString($encryptedValue);
} catch (DecryptException $e) {
    //
}
```