Course - Laravel Framework

Giống dữ liệu

Laravel đưa vào tính năng giống cơ sở dữ liệu của bạn với dữ liệu bằng cách sử dụng các class seed. Tất cả các class seed được lưu trữ trong thư mục database/seeders. Theo mặc định, một class DatabaseSeeder được xác định cho bạn. Từ class này, bạn có thể sử dụng phương thức gọi để chạy các class seed, cho phép bạn kiểm soát thứ tự giống dữ liệu.

Tags: Seeding, giong du lieu, laravel

Giới thiệu

Laravel đưa vào tính năng gióng cơ sở dữ liệu của bạn với dữ liệu bằng cách sử dụng các class seed. Tất cả các class seed được lưu trữ trong thư mục *database/seeders*. Theo mặc định, một class `DatabaseSeeder` được xác định cho bạn. Từ class này, bạn có thể sử dụng phương thức gọi để chạy các class seed, cho phép bạn kiểm soát thứ tự gióng dữ liệu.

> Tính năng bảo vệ gán hàng loạt sẽ tự động bị vô hiệu hóa trong quá trình tạo cơ sở dữ liệu.

Viết chương trình gióng dữ liệu - seeder

Để tạo ra seeder, hãy chạy lệnh Artisan `make:seeder`. Tất cả các seeder được tạo bởi framework sẽ được đặt trong thư mục *database/seeders*:

```
php artisan make:seeder UserSeeder
```

Một class seeder chỉ chứa một phương thức theo mặc định: `run`. Phương thức này được gọi khi lệnh Artisan `db:seed` được thực thi. Trong phương thức `run`, bạn có thể chèn dữ liệu vào cơ sở dữ liệu của mình theo bất kỳ cách nào bạn muốn. Bạn có thể sử dụng query builder để chèn dữ liệu theo cách thủ công hoặc bạn có thể sử dụng các Eloquent model factories.

Ví dụ: hãy sửa đổi class `DatabaseSeeder` mặc định và thêm câu lệnh chèn cơ sở dữ liệu vào phương thức `run`:

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Str;

class DatabaseSeeder extends Seeder
{
  /**
```

```
     * Run the database seeders.
     *
     * @return void
     */
    public function run()
    {
        DB::table('users')->insert([
            'name' => Str::random(10),
            'email' => Str::random(10).'@gmail.com',
            'password' => Hash::make('password'),
        ]);
    }
}
```

Bạn có thể khai báo kiểu bất kỳ thư viện nào bạn cần trong cú pháp của phương thức `run`. Chúng sẽ tự động được trả lại thông qua service container Laravel.

Sử dụng model factories

Việc khai báo thủ công các thuộc tính cho từng model seed là rất phức tạp. Thay vào đó, bạn có thể sử dụng các model factory để tạo ra một lượng lớn các record cơ sở dữ liệu một cách thuận tiện. Trước tiên, hãy xem lại tài liệu về model factory để tìm hiểu cách tạo factory của bạn.

Ví dụ: hãy tạo 50 người dùng mà mỗi người có một bài đăng của họ:

```
use App\Models\User;

/**
 * Run the database seeders.
 *
 * @return void
 */
public function run()
{
    User::factory()
        ->count(50)
```

```
    ->hasPosts(1)
    ->create();
}
```

Gọi các seeder bổ sung

Trong class `DatabaseSeeder`, bạn có thể sử dụng phương thức `call` để thực thi các class hạt giống bổ sung. Sử dụng phương thức `call` cho phép bạn chia nhỏ việc gióng cơ sở dữ liệu của mình thành nhiều tập tin để không có class seeder nào trở nên quá lớn. Phương thức `call` chấp nhận một mảng các class seeder sẽ được thực thi:

```
/**
 * Run the database seeders.
 *
 * @return void
 */
public function run()
{
  $this->call([
    UserSeeder::class,
    PostSeeder::class,
    CommentSeeder::class,
  ]);
}
```

Chạy các seeder

Bạn có thể thực hiện lệnh Artisan `db:seed` để bắt đầu cơ sở dữ liệu của bạn. Theo mặc định, lệnh `db:seed` chạy class `Database\Seeders\DatabaseSeeder`, class này có thể lần lượt gọi các class hạt giống khác. Tuy nhiên, bạn có thể sử dụng tùy chọn `--class` để chỉ định một class seeder cụ thể để chạy riêng lẻ:

```
php artisan db:seed
```

```
php artisan db:seed --class=UserSeeder
```

Bạn cũng có thể khởi tạo cơ sở dữ liệu của mình bằng cách sử dụng lệnh migrate:fresh kết hợp với tùy chọn `--seed`, tùy chọn này sẽ loại bỏ tất cả các bảng và chạy lại tất cả các migration của bạn. Lệnh này rất hữu ích để xây dựng lại hoàn toàn cơ sở dữ liệu của bạn:

```
php artisan migrate:fresh --seed
```

Cưỡng buộc seeder chạy trong môi trường product

Một số thao tác gióng dữ liệu có thể khiến bạn thay đổi hoặc mất dữ liệu. Mặc định, để bảo vệ bạn khỏi việc chạy các lệnh seed trên cơ sở dữ liệu trong mội trường product của bạn, bạn sẽ được nhắc xác nhận trước khi seeder được thực thi trong môi trường `production`. Để buộc seeder chạy mà không có lời nhắc, hãy sử dụng cờ `--force`:

```
php artisan db:seed --force
```