

Course - Laravel Framework

Các packages trong Laravel

Trong tài liệu này chúng ta sẽ được giới thiệu qua các package hỗ trợ cho Laravel.

Tags: Các packages trong Laravel

Các package bao gồm

- [Breeze](#)
- [Cashier \(Stripe\)](#)
- [Cashier \(Paddle\)](#)
- [Dusk](#)
- [Envoy](#)
- [Fortify](#)
- [Homestead](#)
- [Horizon](#)
- [Jetstream](#)
- [Octane](#)
- [Passport](#)
- [Sail](#)
- [Sanctum](#)
- [Scout](#)
- [Socialite](#)
- [Telescope](#)
- [Valet](#)

Breeze

Giới thiệu

Laravel Breeze là một gói triển khai đơn giản, tối thiểu của tất cả các tính năng xác thực của Laravel, bao gồm đăng nhập, đăng ký, đặt lại mật khẩu, xác minh email và xác nhận mật khẩu. Class hiển thị mặc định của Laravel Breeze được tạo thành từ các mẫu Blade đơn giản được tạo kiểu với Tailwind CSS.

Breeze cung cấp điểm khởi đầu tuyệt vời để bắt đầu một ứng dụng Laravel mới và cũng là lựa chọn tuyệt vời cho các dự án có kế hoạch đưa các mẫu Blade của họ lên một tầm cao mới với Laravel Livewire.

Cài đặt

Trước tiên, bạn nên tạo một ứng dụng Laravel mới, cấu hình cơ sở dữ liệu và chạy quá trình tích hợp cơ sở dữ liệu của bạn:

```
curl -s https://laravel.build/example-app | bash
cd example-app
```

```
php artisan migrate
```

Khi bạn đã tạo một ứng dụng Laravel mới, bạn có thể cài đặt Laravel Breeze bằng Composer:

```
composer require laravel/breeze:1.9.2
```

Sau khi Composer cài đặt gói Laravel Breeze, bạn có thể chạy lệnh Artisan **breeze:install**. Lệnh này xuất bản các giao diện hiển thị xác thực, các route, controller và các tài nguyên khác cho ứng dụng của bạn. Laravel Breeze xuất bản tất cả code của nó lên ứng dụng của bạn để bạn có toàn quyền kiểm soát và khả năng hiển thị đối với các tính năng và cách triển khai của nó. Sau khi cài đặt Breeze, bạn cũng nên biên dịch nội dung của mình để tập tin CSS của ứng dụng có sẵn:

```
php artisan breeze:install
```

```
npm install
```

```
npm run dev
```

```
php artisan migrate
```

Tiếp theo, bạn có thể điều hướng đến URL của ứng dụng `/login` hoặc `/register` trong trình duyệt web của bạn. Tất cả các route của Breeze được khai báo trong tập tin `route/auth.php`.

Để tìm hiểu thêm thông tin chi tiết về package Breeze:
<https://laravel.com/docs/8.x/starter-kits#laravel-breeze>

Cashier (Stripe)

Giới thiệu

Laravel Cashier Stripe cung cấp một giao diện linh hoạt, rõ ràng cho các dịch vụ thanh toán thuê bao của Stripe. Nó xử lý gần như tất cả các mã thanh toán đăng ký soạn sẵn mà bạn đang sợ viết. Ngoài quản lý đăng ký cơ bản, Cashier có thể xử lý phiếu giảm giá, hoán đổi đăng ký, "số lượng" đăng ký, thời gian gia hạn hủy và thậm chí tạo các tập tin PDF hóa đơn.

Cài đặt

Trước tiên, hãy cài đặt gói Cashier cho Stripe bằng trình quản lý gói Composer:

```
composer require laravel/cashier
```

Chú ý: Để đảm bảo Cashier xử lý đúng tất cả các sự kiện của Stripe, hãy nhớ thiết lập xử lý webhook của Cashier.

Để tìm hiểu thêm thông tin chi tiết về package Cashier (Stripe):
<https://laravel.com/docs/8.x/billing>

Cashier (Paddle)

Giới thiệu

Laravel Cashier Paddle cung cấp một giao diện linh hoạt, rõ ràng cho các dịch vụ thanh toán đăng ký của Paddle. Nó xử lý gần như tất cả các code thanh toán đăng ký soạn sẵn mà bạn đang sợ. Ngoài quản lý đăng ký cơ bản, Cashier có thể xử lý: phiếu giảm giá, hoán đổi đăng ký, "số lượng" đăng ký, thời gian gia hạn hủy và hơn thế nữa.

Trong khi làm việc với Cashier, chúng tôi khuyên bạn cũng nên xem lại hướng dẫn sử dụng và tài liệu API của Paddle.

Cài đặt

Trước tiên, hãy cài đặt gói Cashier cho Paddle bằng trình quản lý gói Composer:

```
composer require laravel/cashier-paddle
```

Chú ý: Để đảm bảo Cashier xử lý đúng tất cả các sự kiện Paddle, hãy nhớ thiết lập xử lý webhook của Cashier.

Để tìm hiểu thêm thông tin chi tiết về package Cashier (Paddle):
<https://laravel.com/docs/8.x/cashier-paddle>

Dusk

Giới thiệu

Laravel Dusk cung cấp một API kiểm tra và tự động hóa trình duyệt nhanh chóng, dễ sử dụng. Theo mặc định, Dusk không yêu cầu bạn cài đặt JDK hoặc Selenium trên máy tính cục bộ của bạn. Thay vào đó, Dusk sử dụng ChromeDriver. Tuy nhiên, bạn có thể tự do sử dụng bất kỳ driver tương thích Selenium nào khác mà bạn muốn.

Cài đặt

Để bắt đầu, bạn nên cài đặt Google Chrome và thêm package **laravel/dusk** vào dự án của mình:

```
composer require --dev laravel/dusk
```

Chú ý: Nếu bạn đang đăng ký nhà cung cấp dịch vụ của Dusk theo cách thủ công, bạn không nên đăng ký nó trong môi trường production của mình, vì làm như vậy có thể dẫn đến việc người dùng tùy ý có thể xác thực với ứng dụng của bạn.

Để tìm hiểu thêm thông tin chi tiết về package Dusk:
<https://laravel.com/docs/8.x/dusk>

Envoy

Giới thiệu

Laravel Envoy là một công cụ để thực thi các tác vụ phổ biến mà bạn chạy trên các máy chủ từ xa của mình. Sử dụng cú pháp kiểu Blade, bạn có thể dễ dàng thiết lập các tác vụ để triển khai, lệnh Artisan, v.v. Hiện tại, Envoy chỉ hỗ trợ hệ điều hành Mac và Linux. Tuy nhiên, hỗ trợ Windows có thể đạt được bằng cách sử dụng WSL2.

Cài đặt

Trước tiên, hãy cài đặt Envoy vào dự án của bạn bằng trình quản lý gói Composer:

```
composer require laravel/envoy --dev
```

Khi Envoy đã được cài đặt, nhị phân Envoy sẽ có sẵn trong thư mục vendor/bin của ứng dụng của bạn:

```
php vendor/bin/envoy
```

Để tìm hiểu thêm thông tin chi tiết về package Envoy:
<https://laravel.com/docs/8.x/envoy>

Fortify

Giới thiệu

Laravel Fortify là một triển khai phụ trợ xác thực bất khả tri giao diện người dùng cho Laravel. Fortify đăng ký các route và controller cần thiết để triển khai tất cả các tính năng xác thực của Laravel, bao gồm đăng nhập, đăng ký, đặt lại mật khẩu, xác minh email, v.v. Sau khi cài đặt Fortify, bạn có thể chạy lệnh **route:list** Artisan để xem các route mà Fortify đã đăng ký.

Vì Fortify không cung cấp giao diện người dùng của riêng mình, nên nó có nghĩa là được ghép nối với giao diện người dùng của riêng bạn để đưa ra yêu cầu đối với các tuyến đường mà nó đăng ký. Chúng ta sẽ thảo luận chính xác cách thực hiện yêu cầu đối với các route này trong phần còn lại của tài liệu này.

Cài đặt

Để bắt đầu, hãy cài đặt Fortify bằng trình quản lý gói Composer:

```
composer require laravel/fortify
```

Tiếp theo, xuất bản tài nguyên của Fortify bằng lệnh **vendor:publish**:

```
php artisan vendor:publish --provider="Laravel\Fortify\FortifyServiceProvider"
```

Lệnh này sẽ xuất bản các hành động của Fortify lên thư mục **app/Actions** của bạn, thư mục này sẽ được tạo nếu nó không tồn tại. Ngoài ra, tập tin cấu hình và migration của Fortify sẽ được xuất bản.

Tiếp theo, bạn nên di chuyển cơ sở dữ liệu của mình:

```
php artisan migrate
```

Để tìm hiểu thêm thông tin chi tiết về package Fortify:
<https://laravel.com/docs/8.x/fortify>

Homestead

Giới thiệu

Laravel cố gắng làm cho toàn bộ trải nghiệm phát triển PHP trở nên thú vị, bao gồm cả môi trường phát triển cục bộ của bạn. Laravel Homestead là một Vagrant đóng gói sẵn, chính thức, cung cấp cho bạn một môi trường phát triển tuyệt vời mà không yêu cầu bạn cài đặt PHP, máy chủ web và bất kỳ phần mềm máy chủ nào khác trên máy cục bộ của bạn.

Vagrant cung cấp một cách đơn giản, gọn gàng để quản lý và cung cấp Virtual Machine. Vagrant hoàn toàn dùng một lần. Nếu xảy ra sự cố, bạn có thể phá hủy và tạo lại trong vài phút!

Homestead chạy trên bất kỳ hệ thống Windows, macOS hoặc Linux nào và bao gồm Nginx, PHP, MySQL, PostgreSQL, Redis, Memcached, Node và tất cả các phần mềm khác mà bạn cần để phát triển các ứng dụng Laravel tuyệt vời.

Chú ý: Nếu đang sử dụng Windows, bạn có thể cần bật ảo hóa phần cứng (VT-x). Nó thường có thể được kích hoạt thông qua BIOS của bạn. Nếu bạn đang sử dụng Hyper-V trên hệ thống UEFI, bạn cũng có thể cần phải tắt Hyper-V để truy cập VT-x.

Cài đặt và cấu hình

Để tìm hiểu thêm thông tin chi tiết về package Homestead:
<https://laravel.com/docs/8.x/homestead>

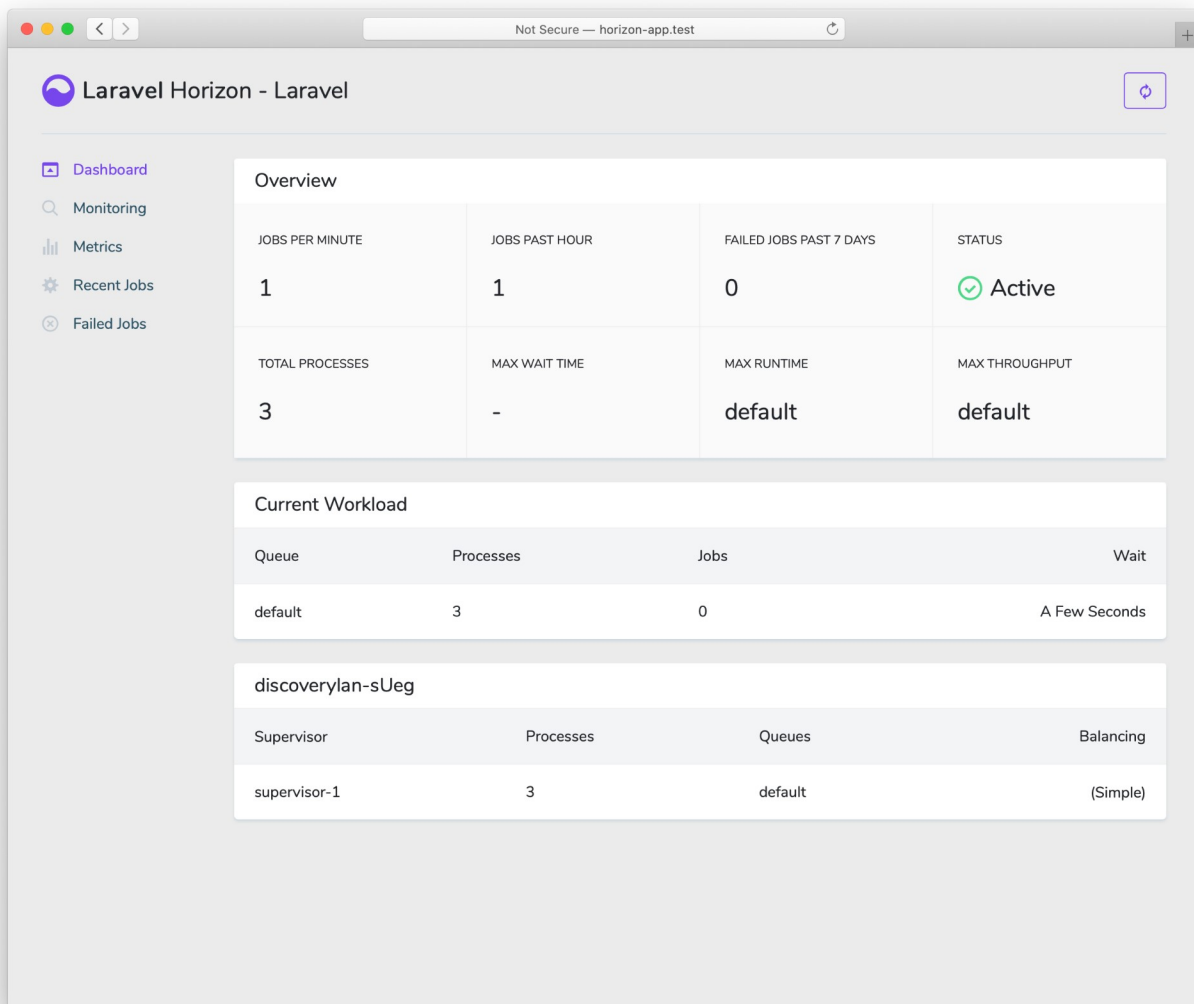
Horizon

Giới thiệu

Trước khi tìm hiểu về Laravel Horizon, bạn nên tự làm quen với các dịch vụ hàng chờ cơ sở của Laravel. Horizon tăng cường hàng chờ của Laravel với các tính năng bổ sung có thể gây nhầm lẫn nếu bạn chưa quen với các tính năng hàng đợi cơ bản do Laravel cung cấp.

Laravel Horizon cung cấp một bảng điều khiển đẹp mắt và cấu hình theo hướng code cho các hàng đợi Redis được hỗ trợ bởi Laravel của bạn. Horizon cho phép bạn dễ dàng theo dõi các chỉ số chính của hệ thống hàng chờ của mình như thông lượng tác vụ, thời gian chạy và lỗi tác vụ.

Khi sử dụng Horizon, tất cả cấu hình worker hàng chờ của bạn được lưu trữ trong một tập tin cấu hình đơn giản. Bằng cách xác định cấu hình worker của ứng dụng trong tập tin được kiểm soát phiên bản, bạn có thể dễ dàng mở rộng quy mô hoặc sửa đổi worker hàng chờ của ứng dụng khi triển khai ứng dụng của mình.



Cài đặt

Chú ý: Laravel Horizon yêu cầu bạn sử dụng Redis để cấp nguồn cho hàng đợi của mình. Do đó, bạn nên đảm bảo rằng kết nối hàng chờ của bạn được đặt thành redis

trong tập tin cấu hình *config/queue.php* của ứng dụng.

Bạn có thể cài đặt Horizon vào dự án của mình bằng trình quản lý gói Composer:

```
composer require laravel/horizon
```

Sau khi cài đặt Horizon, hãy xuất bản nội dung của nó bằng cách sử dụng lệnh Artisan `horizon:install`:

```
php artisan horizon:install
```

Để tìm hiểu thêm thông tin chi tiết về package Horizon:

<https://laravel.com/docs/8.x/horizon>

Jetstream

Giới thiệu

Laravel Jetstream là bộ khởi động ứng dụng được thiết kế đẹp mắt dành cho Laravel và cung cấp điểm khởi đầu hoàn hảo cho ứng dụng Laravel tiếp theo của bạn. Jetstream cung cấp việc triển khai cho ứng dụng của bạn đăng nhập, đăng ký, xác minh email, xác thực hai yếu tố, quản lý phiên, API qua Laravel Sanctum và các tính năng quản lý nhóm tùy chọn.

Cài đặt

Bạn có thể sử dụng Composer để cài đặt Jetstream vào dự án Laravel mới của mình:

```
composer require laravel/jetstream
```

Sau khi cài đặt gói Jetstream, bạn có thể thực hiện lệnh Artisan **jetstream:install**.

Lệnh này chấp nhận tên của ngăn xếp mà bạn thích (trực tiếp hoặc quán tính). Ngoài ra, bạn có thể sử dụng nút chuyển đổi **--teams** để bật hỗ trợ nhóm.

Để tìm hiểu thêm thông tin chi tiết về package Jetstream:

<https://jetstream.laravel.com/>

Octane

Giới thiệu

Laravel Octane tăng cường hiệu suất ứng dụng của bạn bằng cách cung cấp ứng dụng của bạn bằng cách sử dụng các máy chủ ứng dụng công suất cao, bao gồm Open Swoole, Swoole và RoadRunner. Octane khởi động ứng dụng của bạn một lần, giữ nó trong bộ nhớ và sau đó cung cấp các yêu cầu cho ứng dụng ở tốc độ siêu thanh.

Cài đặt

Octane có thể được cài đặt thông qua trình quản lý gói Composer:

```
composer require laravel/octane
```

Sau khi cài đặt Octane, bạn có thể thực thi lệnh Artisan **octane:install**, lệnh này sẽ cài đặt tập tin cấu hình của Octane vào ứng dụng của bạn:

```
php artisan octane:install
```

Để tìm hiểu thêm thông tin chi tiết về package Octane:
<https://laravel.com/docs/8.x/octane>

Passport

Giới thiệu

Laravel Passport cung cấp triển khai máy chủ OAuth2 đầy đủ cho ứng dụng Laravel của bạn chỉ trong vài phút. Passport được xây dựng trên máy chủ League OAuth2 do Andy Millington và Simon Hamp bảo trì.

Cài đặt

Để bắt đầu, hãy cài đặt Passport qua trình quản lý gói Composer:

```
composer require laravel/passport
```

Service provider của Passport đăng ký thư mục tích hợp cơ sở dữ liệu của riêng họ, vì vậy bạn nên tích hợp cơ sở dữ liệu của mình sau khi cài đặt package này. Các bản tích hợp dữ liệu Passport sẽ tạo các bảng mà ứng dụng của bạn cần để lưu trữ các ứng dụng khách OAuth2 và mã token truy cập:

```
php artisan migrate
```

Tiếp theo, bạn nên thực hiện lệnh Artisan **passport:install**. Lệnh này sẽ tạo các khóa mã hóa cần thiết để tạo mã token truy cập an toàn. Ngoài ra, lệnh sẽ tạo ứng dụng khách "truy cập cá nhân" và "cấp mật khẩu" sẽ được sử dụng để tạo mã token truy cập:

```
php artisan passport:install
```

Để tìm hiểu thêm thông tin chi tiết về package Passport:
<https://laravel.com/docs/8.x/passport>

Sail

Giới thiệu

Laravel Sail là một giao diện dòng lệnh nhẹ để tương tác với môi trường phát triển Docker mặc định của Laravel. Sail cung cấp một điểm khởi đầu tuyệt vời để xây dựng một ứng dụng Laravel bằng PHP, MySQL và Redis mà không yêu cầu kinh nghiệm Docker trước đó.

Về cơ bản, Sail là tập tin *docker-compose.yml* và tập lệnh *sail* được lưu trữ tại thư mục root của dự án của bạn. Tập lệnh *sail* cung cấp một CLI với các phương thức thuận tiện để tương tác với các container của Docker được tạo bởi tập tin *docker-compose.yml*.

Laravel Sail được hỗ trợ trên macOS, Linux và Windows (thông qua WSL2).

Cài đặt

Laravel Sail được cài đặt tự động với tất cả các ứng dụng Laravel mới nên bạn có thể bắt đầu sử dụng nó ngay lập tức. Để tìm hiểu cách tạo một ứng dụng Laravel mới, vui lòng tham khảo tài liệu cài đặt của Laravel cho hệ điều hành của bạn. Trong khi cài đặt, bạn sẽ được yêu cầu chọn các dịch vụ hỗ trợ Sail mà ứng dụng của bạn sẽ tương tác.

Cài đặt Sail vào các ứng dụng hiện có

Nếu bạn quan tâm đến việc sử dụng Sail với ứng dụng Laravel hiện có, bạn có thể chỉ cần cài đặt Sail bằng trình quản lý gói Composer. Tất nhiên, các bước này giả định rằng môi trường phát triển cục bộ hiện tại của bạn cho phép bạn cài đặt các phần thư viện laravel của Composer:

```
composer require laravel/sail --dev
```

Sau khi Sail đã được cài đặt, bạn có thể chạy lệnh Artisan **sail:install**. Lệnh này sẽ xuất bản tập tin *docker-compose.yml* của Sail lên thư mục root của ứng dụng của bạn:

```
php artisan sail:install
```

Cuối cùng, bạn có thể bắt đầu Sail. Để tiếp tục tìm hiểu cách sử dụng Sail, vui lòng tiếp tục đọc phần còn lại của tài liệu này:

```
./vendor/bin/sail up
```

Để tìm hiểu thêm thông tin chi tiết về package Sail:
<https://laravel.com/docs/8.x/sail>

Sanctum

Giới thiệu

Laravel Sanctum cung cấp một hệ thống xác thực nhẹ cho các SPA (ứng dụng trang đơn), ứng dụng di động và các API đơn giản dựa trên mã token. Sanctum cho phép mỗi người dùng ứng dụng của bạn tạo nhiều mã token API cho tài khoản của họ. Các mã token này có thể được cấp các khả năng / phạm vi chỉ định các hành động mà mã token được phép thực hiện.

Cài đặt

Các phiên bản mới nhất của Laravel đã bao gồm Laravel Sanctum. Tuy nhiên, nếu tập tin *composer.json* của ứng dụng của bạn không bao gồm **laravel/sanctum**, bạn có thể làm theo hướng dẫn cài đặt bên dưới.

Bạn có thể cài đặt Laravel Sanctum thông qua trình quản lý gói Composer:

```
composer require laravel/sanctum
```

Tiếp theo, bạn nên xuất bản cấu hình Sanctum và các tập tin tích hợp dữ liệu bằng lệnh Artisan **vendor:publish**. Tập tin cấu hình sanctum sẽ được đặt trong thư mục cấu hình ứng dụng của bạn:

```
php artisan vendor:publish --provider="Laravel\Sanctum\SanctumServiceProvider"
```

Cuối cùng, bạn nên chạy tích hợp dữ liệu cơ sở dữ liệu của mình. Sanctum sẽ tạo một bảng cơ sở dữ liệu để lưu trữ các mã token API:

```
php artisan migrate
```

Tiếp theo, nếu bạn định sử dụng Sanctum để xác thực một SPA, bạn nên thêm middleware của Sanctum vào nhóm middleware api trong tập tin *app/Http/Kernel.php* của ứng dụng:

```
'api' => [  
    \Laravel\Sanctum\Http\Middleware\EnsureFrontendRequestsAreStateful::class,  
    'throttle:api',  
    \Illuminate\Routing\Middleware\SubstituteBindings::class,  
],
```

Để tìm hiểu thêm thông tin chi tiết về package Sanctum:

<https://laravel.com/docs/8.x/sanctum>

Scout

Giới thiệu

Laravel Scout cung cấp một giải pháp đơn giản, dựa trên driver để thêm tìm kiếm toàn văn bản vào các model Eloquent của bạn. Sử dụng model observers, Scout sẽ tự động giữ các chỉ mục tìm kiếm của bạn đồng bộ với các record Eloquent của bạn.

Hiện tại, Scout xuất xưởng với tài xế Algolia và MeiliSearch. Ngoài ra, Scout bao gồm một driver "bộ sưu tập" được thiết kế để sử dụng phát triển cục bộ và không yêu cầu bất kỳ phụ thuộc bên ngoài hoặc dịch vụ của bên thứ ba. Hơn nữa, việc viết driver tùy chỉnh rất đơn giản và bạn có thể tự do mở rộng Scout với các triển khai tìm kiếm của riêng mình.

Cài đặt

Đầu tiên, cài đặt Scout thông qua trình quản lý gói Composer:

```
composer require laravel/scout
```

Sau khi cài đặt Scout, bạn nên xuất bản tập tin cấu hình Scout bằng lệnh Artisan **vendor:publish**. Lệnh này sẽ xuất bản tập tin cấu hình *scout.php* vào thư mục cấu hình của ứng dụng của bạn:

```
php artisan vendor:publish --provider="Laravel\Scout\ScoutServiceProvider"
```

Cuối cùng, thêm trait **Laravel\Scout\Searchable** vào model bạn muốn làm cho có thể tìm kiếm được. Trait này sẽ đăng ký một model observer mà sẽ tự động giữ cho model đồng bộ với driver tìm kiếm của bạn:

```
<?php

namespace App\Models;
use Illuminate\Database\Eloquent\Model;
use Laravel\Scout\Searchable;

class Post extends Model
{
    use Searchable;
}
```

Để tìm hiểu thêm thông tin chi tiết về package Scout:
<https://laravel.com/docs/8.x/scout>

Socialite

Giới thiệu

Ngoài xác thực dựa trên biểu mẫu điển hình, Laravel cũng cung cấp một cách đơn giản, thuận tiện để xác thực với các nhà cung cấp OAuth bằng Laravel Socialite. Socialite hiện hỗ trợ xác thực với Facebook, Twitter, LinkedIn, Google, GitHub, GitLab và Bitbucket.

Bộ điều hợp cho các nền tảng khác được liệt kê tại trang web Nhà cung cấp Socialite do cộng đồng định hướng.

Cài đặt

Để bắt đầu với Socialite, hãy sử dụng trình quản lý gói Composer để thêm gói vào các phần thư viện Laravel của dự án của bạn:

```
composer require laravel/socialite
```

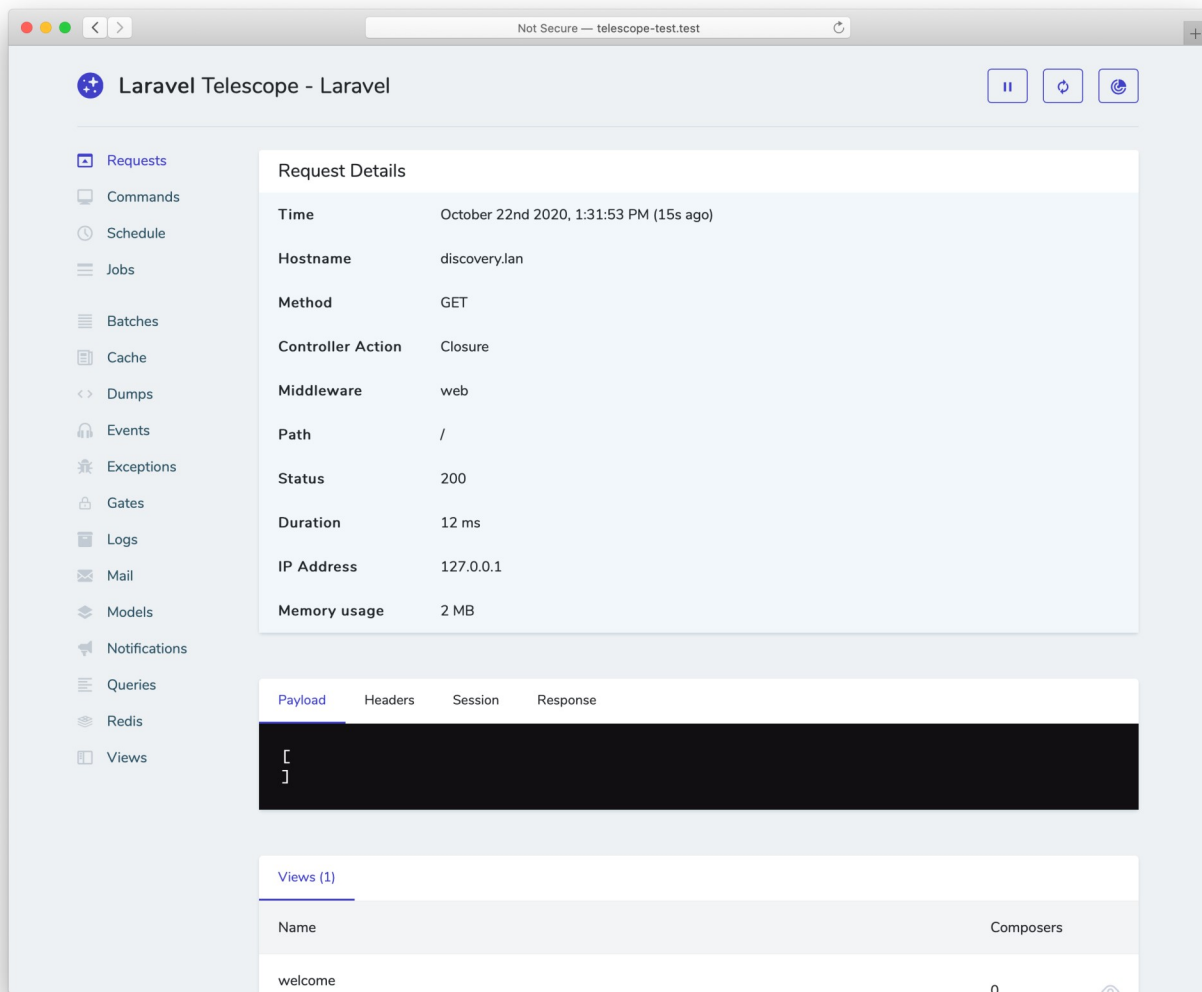
Để tìm hiểu thêm thông tin chi tiết về package Socialite:

<https://laravel.com/docs/8.x/socialite>

Telescope

Giới thiệu

Kính viễn vọng Laravel là người bạn đồng hành tuyệt vời với môi trường phát triển Laravel tại địa phương của bạn. Telescope cung cấp thông tin chi tiết về các yêu cầu gửi đến ứng dụng của bạn, ngoại lệ, mục nhập nhật ký, truy vấn cơ sở dữ liệu, công việc được xếp hàng đợi, thư, thông báo, hoạt động bộ nhớ cache, tác vụ đã lên lịch, kết xuất có thể thay đổi và hơn thế nữa.



Cài đặt

Bạn có thể sử dụng trình quản lý gói Composer để cài đặt Telescope vào dự án Laravel của mình:

```
composer require laravel/telescope
```

Sau khi cài đặt Telescope, hãy xuất bản nội dung của nó bằng cách sử dụng telescope: install Artisan command. Sau khi cài đặt Telescope, bạn cũng nên chạy lệnh di chuyển để tạo các bảng cần thiết để lưu trữ dữ liệu của Telescope:


```
php artisan telescope:install
```

```
php artisan migrate
```

Tùy biến tích hợp dữ liệu

Nếu bạn không sử dụng tích hợp dữ liệu mặc định của Telescope, bạn nên gọi phương thức **Telescope::ignoreMigrations** trong phương thức register của class **App\Providers\AppServiceProvider** của ứng dụng của bạn. Bạn có thể xuất các tích hợp dữ liệu mặc định bằng cách sử dụng lệnh sau:

```
php artisan vendor:publish --tag=telescope-migrations
```

Để tìm hiểu thêm thông tin chi tiết về package Telescope:
<https://laravel.com/docs/8.x/telescope>

Valet

Giới thiệu

Laravel Valet là một môi trường phát triển dành cho những người theo chủ nghĩa tối giản macOS. Laravel Valet định cấu hình máy Mac của bạn luôn chạy Nginx ở chế độ nền khi máy của bạn khởi động. Sau đó, sử dụng DnsMasq, Valet proxy tất cả các yêu cầu trên miền ***.test** để trở đến các trang web được cài đặt trên máy cục bộ của bạn.

Nói cách khác, Valet là một môi trường phát triển Laravel nhanh như chớp sử dụng khoảng 7 MB RAM. Valet không phải là sự thay thế hoàn toàn cho Sail hoặc Homestead, nhưng cung cấp một giải pháp thay thế tuyệt vời nếu bạn muốn có những điều cơ bản linh hoạt, thích tốc độ cực cao hoặc đang làm việc trên một máy có số lượng RAM hạn chế.

Ngoài ra, hỗ trợ Valet bao gồm, nhưng không giới hạn:

[Laravel](#)

[Concrete5](#)

[ExpressionEngine](#)

[Lumen](#)

[Contao](#)

[Jigsaw](#)

[Bedrock](#)

[Craft](#)

[Joomla](#)

[CakePHP 3](#)

[Drupal](#)

[Katana](#)

[Kirby](#)

[Slim](#)

[WordPress](#)

[Magento](#)

[Statamic](#)

[Zend](#)

[OctoberCMS](#)

[Static HTML](#)

[Sculpin](#)

[Symfony](#)

Tuy nhiên, bạn có thể mở rộng Valet với các driver tùy chỉnh của riêng bạn.

Cài đặt

Chú ý: Valet yêu cầu macOS và Homebrew. Trước khi cài đặt, bạn nên đảm bảo rằng không có chương trình nào khác như Apache hoặc Nginx liên kết với cổng 80 của máy cục bộ của bạn.

Để bắt đầu, trước tiên bạn cần đảm bảo rằng Homebrew được cập nhật bằng cách sử dụng lệnh cập nhật:

```
brew update
```

Tiếp theo, bạn nên sử dụng Homebrew để cài đặt PHP:

```
brew install php
```

Sau khi cài đặt PHP, bạn đã sẵn sàng cài đặt trình quản lý gói Composer. Ngoài ra, bạn nên đảm bảo rằng thư mục `~/.composer/vendor/bin` nằm trong "PATH" của hệ thống. Sau khi cài đặt Composer, bạn có thể cài đặt Laravel Valet như một gói Composer chung:

```
composer global require laravel/valet
```

Cuối cùng, bạn có thể thực hiện lệnh **install** của Valet. Điều này sẽ cấu hình và cài đặt Valet và DnsMasq. Ngoài ra, daemons Valet phụ thuộc vào sẽ được cấu hình để khởi chạy khi hệ thống của bạn khởi động:

```
valet install
```

Sau khi Valet được cài đặt, hãy thử ping bất kỳ miền ***.test** nào trên thiết bị đầu cuối của bạn bằng cách sử dụng lệnh như **ping foobar.test**. Nếu Valet được cài đặt đúng cách, bạn sẽ thấy miền này phản hồi trên `127.0.0.1`.

Valet sẽ tự động khởi động các dịch vụ cần thiết mỗi khi máy của bạn khởi động.

Các phiên bản PHP

Valet cho phép bạn chuyển đổi các phiên bản PHP bằng lệnh **valet use php@version**. Valet sẽ cài đặt phiên bản PHP được chỉ định thông qua Homebrew nếu nó chưa được cài đặt:

```
valet use php@7.2
```

```
valet use php
```

Bạn cũng có thể tạo tập tin **.valetphprc** trong thư mục root của dự án của bạn. Tập tin **.valetphprc** phải chứa phiên bản PHP mà trang web sẽ sử dụng:

```
php@7.2
```

Khi tập tin này đã được tạo, bạn có thể chỉ cần thực hiện lệnh **valet use** và lệnh sẽ xác định phiên bản PHP ưa thích của trang web bằng cách đọc tập tin.

Chú ý: Valet chỉ cung cấp một phiên bản PHP tại một thời điểm, ngay cả khi bạn đã cài đặt nhiều phiên bản PHP.

Cơ sở dữ liệu

Nếu ứng dụng của bạn cần một cơ sở dữ liệu, hãy xem DBngin. DBngin cung cấp một công cụ quản lý cơ sở dữ liệu miễn phí, tất cả trong một bao gồm MySQL, PostgreSQL và Redis. Sau khi DBngin đã được cài đặt, bạn có thể kết nối với cơ sở dữ liệu của mình tại **127.0.0.1** bằng tên người dùng **root** và một chuỗi trống cho mật khẩu.

Đặt lại cài đặt của bạn

Nếu bạn gặp sự cố cài đặt Valet của mình chạy đúng cách, việc thực thi lệnh **composer global update** ngay sau **valet install** sẽ thiết lập lại cài đặt của bạn và có thể giải quyết nhiều vấn đề khác nhau. Trong một số trường hợp hiếm hoi, có thể cần phải "hard reset" Valet bằng cách thực thi lệnh **valet uninstall --force** sau đó là **valet install**.

Để tìm hiểu thêm thông tin chi tiết về package Valet:
[*https://laravel.com/docs/8.x/valet*](https://laravel.com/docs/8.x/valet)