

Course - Laravel Framework

Vòng đời của request

Trong bài này chúng ta sẽ tìm hiểu rõ hơn về request, khi bạn hiểu rõ về công cụ phát triển của bạn, thì bạn sẽ trở nên thoải mái và tự tin hơn khi sử dụng chúng.

Tags: Vòng đời của request

Khi bạn phát triển một phần mềm bạn sẽ cảm thấy tự tin hơn khi bạn hiểu về các chức năng của công cụ hoặc framework, cách thức mà framework đó hoạt động. Trong tài liệu này sẽ trọng tâm vào kiến thức tổng quát cách thức mà framework chạy trong bộ máy.

Tổng quan vòng đời

Những bước đầu tiên

Entry point cho tất cả các request được gửi đến các ứng dụng Laravel chính là tập tin *public/index.php*. Tất cả các request đều được điều hướng đến tập tin này bởi cấu hình web-server của bạn. Tập tin *index.php* này không chứa quá nhiều code, nó chỉ đơn thuần tải những phần còn lại của framework.

Tập tin *index.php* sẽ tải bản khai báo autoloader trong cơ cấu Composer, sau đó nó sẽ nhận một instance ứng dụng Laravel từ tập tin *bootstrap/app.php*. Hành động đầu tiên được thực hiện bởi Laravel là tự tạo ra một instance của ứng dụng / service container.

HTTP/Console Kernel

Kế tiếp đến tất cả các request này sẽ được gửi đến kernel HTTP hoặc Console tùy theo cách gửi request vào ứng dụng. Hai request này được đặt tại vị trí trung tâm nơi mà các request đều tại đi qua. Nhưng hiện giờ chúng ta chỉ quan tâm đến HTTP kernel, nó được đặt trong tập tin *app/Http/Kernel.php*.

HTTP kernel sẽ mở rộng cho class **`Illuminate\Foundation\Http\Kernel`**, đây là class khai báo một mảng **`bootstrappers`** mà sẽ được chạy trước khi một request được thực thi. Những bootstrappers này sẽ cấu hình xử lý lỗi, cấu hình logging, phát hiện môi trường ứng dụng, và thực thi một số lệnh khác cần phải hoàn tất trước khi một request thực sự được xử lý. Thông thường những class này chỉ xử lý cấu hình nội bộ Laravel nên bạn không cần phải quá bận tâm về chúng.

HTTP kernel còn khai báo một loạt các HTTP middleware mà tất cả các request phải đi qua trước khi được xử lý bởi ứng dụng. Những middleware này xử lý việc đọc và viết các HTTP session, nếu ứng dụng của bạn đang trong chế độ bảo trì thì nó sẽ xác minh CSRF token, và nhiều thứ khác.

Signature của phương thức **`handle`** của HTTP kernel cực kỳ đơn giản, nó là một Response. Hãy xem HTTP kernel như một chiếc hộp đen cho toàn bộ ứng dụng của bạn, phản hồi các request của nó với các HTTP response.

Service Provider

Một trong những hành động boot kernel quan trọng nhất đó là tải các service provider cho ứng dụng của bạn. Tất cả các service provider cho ứng dụng đều được cấu hình trong mảng **providers** nằm trong tập tin cấu hình *config/app.php*.

Laravel sẽ lướt qua danh sách các provider này và sẽ lần lượt khởi tạo từng provider. Sau đó phương thức **register** sẽ được gọi cho tất cả các providers này. Một khi các providers này đã được đăng ký hết thì phương thức **boot** sẽ được gọi để boot từng provider lên. Service provider có thể phụ thuộc vào mỗi ràng buộc container đang được đăng ký và sẵn sàng tại thời điểm phương thức **boot** được gọi.

Service provider sẽ chịu trách nhiệm việc boot tất cả các thành phần khác nhau của framework, chẳng hạn như database, queue, validation và routing. Nói chung các tính năng của Laravel được boot và cấu hình bởi service provider, service provider là một phần quan trọng trong tiến trình boot ứng dụng Laravel.

Routing

Một trong những service provider quan trọng có **App\Providers\RouteServiceProvider**. Service provider này làm nhiệm vụ tải các tập tin route được chứa trong thư mục *routes* của ứng dụng. Bây giờ, chúng ta sẽ tìm hiểu xem cách thức mà **RouteServiceProvider** hoạt động.

Một khi ứng dụng đã được boot và tất cả service provider được đăng ký, thì router này sẽ gửi request đến các route hay controller, cũng như các middleware của route tương khớp.

Middleware cung cấp tiện ích trong việc lọc hoặc kiểm định các HTTP request khi được gửi vào ứng dụng. Ví dụ, Laravel sử dụng middleware xác minh người dùng nào đó đã được xác thực, nếu người dùng chưa được xác thực thì ứng dụng sẽ chuyển người đó về màn hình đăng nhập. Nhưng nếu người dùng đã được xác thực thì có thể yêu cầu các tính năng khác của ứng dụng. Có vài middleware được gắn lên toàn bộ ứng dụng của bạn, chúng được khai báo trong thuộc tính **\$middleware** của HTTP kernel ứng dụng. Cũng có một số middleware chỉ được gắn trên một vài route hoặc nhóm route nào đó. Bạn có thể tìm hiểu thêm về middleware trong phần tài liệu về middleware.

Nếu request đã thông qua tất cả các middleware của route tương khớp, thì route hoặc controller sẽ bắt đầu xử lý và các response được trả lại bởi route hoặc controller này sẽ được gửi ngược trở về qua chuỗi các middleware của route.

Hoàn tất

Các request đã được lọc và kiểm định bởi các middleware của route, thì các response được trả về bởi route hoặc controller cũng sẽ được gửi lại qua các middleware của route, điều này giúp chúng ta có thêm cơ hội để sửa chữa hoặc kiểm định lại response.

Cuối cùng, một khi response đã đi qua các middleware. Thì phương thức **handle** của HTTP kernel sẽ trả lại đối tượng response. Và tập tin *index.php* sẽ gọi phương thức **send** lên response được trả lại. Cụ thể là phương thức **send** sẽ gửi nội dung response đến trình duyệt web của người dùng. Như vậy là chúng ta vừa hoàn thành chuyến du hành của chúng ta qua toàn bộ vòng đời của ứng dụng Laravel.

Đọc thêm về Service Provider

Service provider thực sự là chìa khóa để boot ứng dụng Laravel. Ứng dụng được khởi tạo, các service provider được đăng ký, và request được xử lý. Các service provider mặc định của ứng dụng Laravel được đặt trong thư mục *app/Providers*.

Mặc định, **AppServiceProvider** khá trống, Provider này là một nơi tốt để thêm các nhiệm vụ boot và ràng buộc service container. Đối với các dự án lớn hơn, sẽ cần nhiều service provider hơn, và từng service sẽ được ứng với nhiệm vụ boot phù hợp.