**Bài 26: Hướng dẫn tích hợp và sử dụng laravel debugbar package**

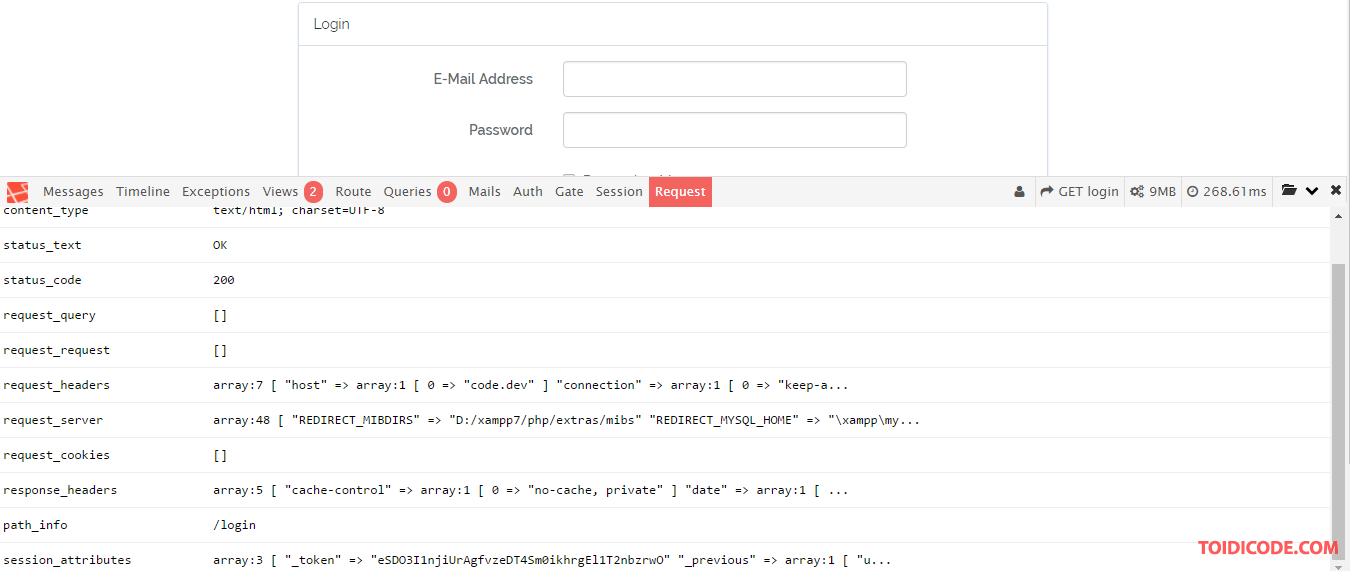
Sau một thời gian dài bỏ qua series này, hôm nay chúng ta sẽ tiếp tục quay lại với series Laravel và tìm hiểu về **package laravel debugbar**...

**1, Laravel debugbar là gì?**

-Laravel debugbar là một gói package được viết ra bởi [Barry vd. Heuvel](https://github.com/barryvdh), với mục đích giúp cho các nhà phát triển trên laravel có thể debug 1 cách nhanh chóng và dễ dàng.

-Package này hỗ trợ chúng ta một số các tính năng nổi bật như:

* Hiện tất cả các câu truy vấn khi load 1 trang đến database.
* Hiện ra tất cả các view cấu thành lên trang hiện tại.
* Hiện thông tin của route hiện tại.
* Hiển thị tất cả các sự kiện trên trang hiện tại.
* Hiển thị toàn bộ các session đang được lưu trữ.
* Hiển thị thông tin của request.
* Hiển thị mức tiêu thụ RAM của ứng dụng.
* ... và còn nhiều chức năng khác nữa...



giao diện của laravel debugbar

**2, Tích hợp Laravel debugbar vào project.**

-Để có thể tích hợp được laravel debugbar bào ứng dụng thì việc đầu tiên của bạn là phải có bộ source của laravel.

-Tiếp đó bạn sử dụng cmd (trên window) hoặc terminal (trên linux) để di chuyển đến project laravel của bạn và chạy lệnh

composer require barryvdh/laravel-debugbar

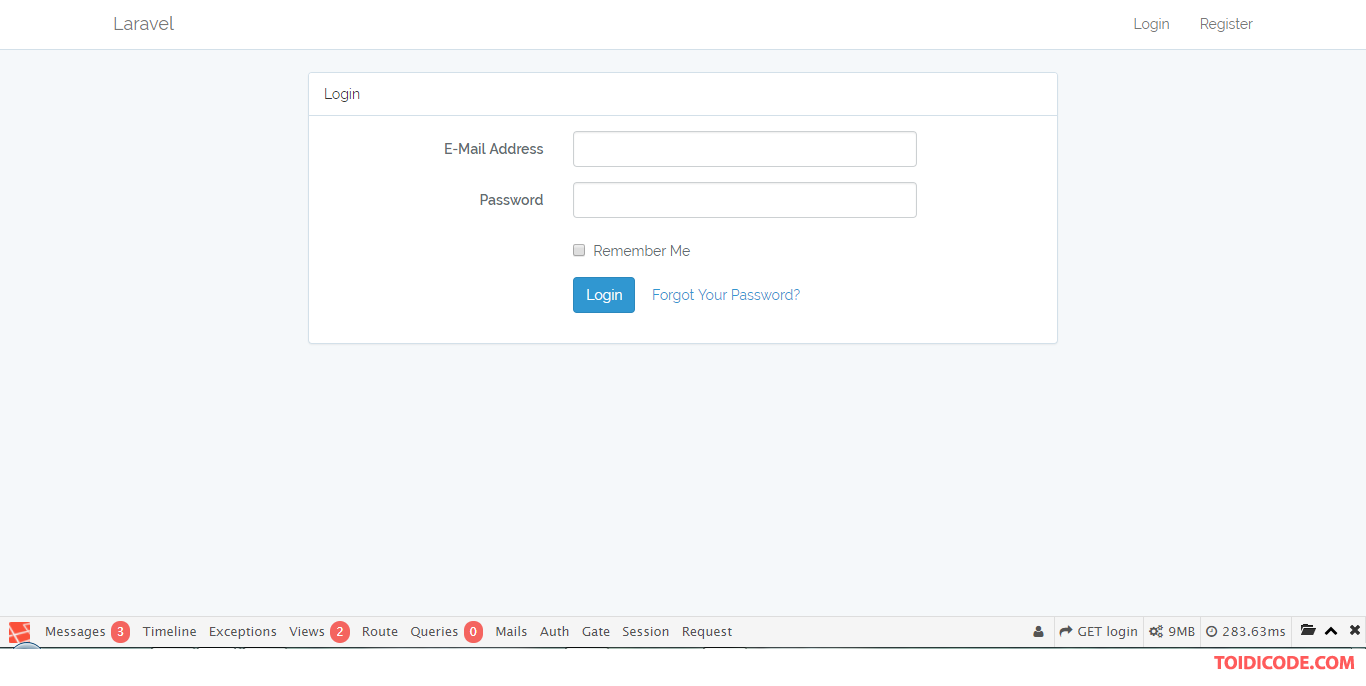
Sau khi đã tải thành công package, thì các bạn vào project laravel của mình và truy cập config/app.php tìm đến key providers và thêm đoạn code sau vào.

Barryvdh\Debugbar\ServiceProvider::class,

-Tiếp đó, nếu như bạn muốn sử dụng **facade** của laravel debugbar thì bạn cần thêm đoạn code sau vào aliases trong file config/app.php.

'Debugbar' => Barryvdh\Debugbar\Facade::class,

-Lúc này bạn quay lại ứng dụng và chạy lại, nếu như ứng dụng không báo lỗi gì và ở phía dưới trang web có một thanh bar dạng như sau thì bạn đã cài đặt thành công.



**3, Sử dụng laravel debugbar.**

-Sau khi đã cài đặt được laravel debugbar rồi thì những chức năng đã được tích hợp sẵn trên đó nhìn vào rất dễ hiểu rồi nên mình không trình bày nữa.

-Và nếu như ở phần trên bạn có tích hợp facede của laravel debugbar thì bạn sẽ sử dụng được các đoạn code dưới đây.

+ Thêm messages:

//Thêm message info

Debugbar::info($object);

//Thêm message error

Debugbar::error('Error!');

//Thêm message warning

Debugbar::warning('Watch out…');

//Thêm message value and name

Debugbar::addMessage('Another message', 'mylabel');

+ Thêm Timeline:

Timeline là công cụ giúp cho bạn có thể check xem trang hiện tại load trong vòng bao nhiêu lâu. Dựa vào đó bạn có thể phân tích xem trên trang có những đoạn code nào gây tốn thời gian nhất và khắc phục.

//Bắt đầu

Debugbar::startMeasure('render','Time for rendering');

//Kết thúc

Debugbar::stopMeasure('render');

Debugbar::addMeasure('now', LARAVEL\_START, microtime(true));

Debugbar::measure('My long operation', function() {

// Do something…

});

 Và còn nhiều chức năng hữu ích khác nữa. Bạn nào muốn tìm hiểu thêm có thể tham khảo tại [đây](https://github.com/barryvdh/laravel-debugbar).