Bài 25: Middleware trong Laravel

## 1, Middleware là gì?

-Middleware theo wikipedia:

**Middleware** là phần mềm máy tính với nhiệm vụ kết nối các thành phần phần mềm hoặc các ứng dụng với nhau. Phần mềm loại này bao gồm một tập các dịch vụ cho phép sự tương tác giữa các tiến trình chạy trên một hoặc nhiều máy khác nhau. Công nghệ middleware đã được phát triển để cung cấp khả năng hoạt động tương hỗ, phục vụ cho các kiến trúc phân tán thường được để hỗ trợ và đơn giản hóa các ứng dụng phân tán phức tạp.

-Thì ở đây đối với Laravel nó cũng hoàn toàn là như thế.

## 2, Tạo Middleware trong Laravel.

-Để tạo một middleware, trong Laravel đã cung cấp cho chúng ta một lệnh:

php artisan make:middleware MiddelwareName

Trong đó: MiddlewareName là tên của middleware các bạn muốn Tạo.

VD:

php artisan make:middleware CheckLevel

-Nếu nhận được thông báo : **Middleware created successfully.**thì đã tạo thành công middleware.

-middleware mặc định sẽ phải nằm trong thư mục: **app/Http/Middleware**.

Tiếp đó các bạn mở file vừa tạo ra sẽ thấy có dạng:

<?php

namespace App\Http\Middleware;

use Closure;

class CheckLevel

{

/\*\*

\* Handle an incoming request.

\*

\* @param \Illuminate\Http\Request $request

\* @param \Closure $next

\* @return mixed

\*/

public function handle($request, Closure $next)

{

return $next($request);

}

}

## 3, Sử dụng Middleware.

-Mọi xử lý trong middleware đều nằm trong handle function. Mọi người xem qua ví dụ cho dễ hiểu nhé:

VD: Mình muốn user vừa đăng nhập phải có level >1 mới được vào admin:

<?php

namespace App\Http\Middleware;

use Closure;

use Illuminate\Support\Facades\Auth;

class CheckLevel

{

/\*\*

\* Handle an incoming request.

\*

\* @param \Illuminate\Http\Request $request

\* @param \Closure $next

\* @return mixed

\*/

public function handle($request, Closure $next)

{

if(Auth::check()&&Auth::user()->level>1)

{

return $next($request);

}else{

return redirect('/');

}

}

}

Xem VD: trên chắc mọi người cũng hiểu về cách hoạt động của nó rồi nhỉ!.

## 4, Đăng ký middleware.

-Các phần trên mới chỉ là tạo ra middleware và viết code cho nó nhưng chưa sử dụng được, giờ nếu muốn sử dụng được middleware trong project thì chúng ta cần phải đăng ký middleware bằng các kiểu middleware sau đây:

### Global middleware.

-Global middleware là một middel là một middleware mà bất cứ HTTP request nào muốn thực hiện được cũng bắt buộc phải qua nó.

Để đăng ký **global middleware** bạn sẽ phải vào: **app/Http/Kernel.php**tìm đến đoạn:

/\*\*

\* The application's global HTTP middleware stack.

\*

\* These middleware are run during every request to your application.

\*

\* @var array

\*/

protected $middleware = [

\Illuminate\Foundation\Http\Middleware\CheckForMaintenanceMode::class,

];

Đăng ký với cú pháp:

\App\Http\Middleware\MiddlewareName::class

Trong đó : MiddlewareName là tên middleware các bạn muốn thêm.

VD: Mình muốn thêm middleware CheckLevel ở VD trên:

protected $middleware = [

\Illuminate\Foundation\Http\Middleware\CheckForMaintenanceMode::class,

\App\Http\Middleware\CheckLevel::class,

];

### Route middleware.

-Khác với global middleware Route middleware chỉ sử dụng được khi bạn gọi nó ở trong Route.

Để đăng ký Route middleware thì mọi người cũng cần phải vào app/Http/Kernel.php tìm đến đoạn:

protected $routeMiddleware = [

'auth' => \Illuminate\Auth\Middleware\Authenticate::class,

'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,

'bindings' => \Illuminate\Routing\Middleware\SubstituteBindings::class,

'can' => \Illuminate\Auth\Middleware\Authorize::class,

'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,

'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,

];

Đăng ký với cú pháp:

'Name'=> \App\Http\Middleware\MiddlewareName::class

**Trong đó:**

* **Name:**Là tên các bạn muốn đặt cho middleware.
* MiddlewareName là tên của Middleware các bạn muốn thêm.

VD: Mình sẽ thêm middleware CheckLevel vào Route middleware.

'Checklevel'=>\App\Http\Middleware\CheckLevel::class,

Sau khi đăng ký xong middleware các bạn có thể sử dụng nó với route như sau:

Route::get('admin/profile', function () {

//

})->middleware('Checklevel');

Hoặc với nhiều middleware:

Route::get('admin/profile', function () {

//

})->middleware('Middleware1','Middleware2');

Hoặc cũng có thể sử dụng tên đầy đủ của Middleware như sau:

use App\Http\Middleware\CheckLevel;

Route::get('admin/profile', function () {

//

})->middleware(CheckLevel::class);

### Nhóm middleware.

-Với loại middleware này chúng ta có thể gộp các middleware thành các nhóm để gọi cho nhanh.

Để khai báo nhóm middleware chúng ta tìm đến đoạn:

protected $middlewareGroups = [

'web' => [

\App\Http\Middleware\EncryptCookies::class,

\Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,

\Illuminate\Session\Middleware\StartSession::class,

\Illuminate\View\Middleware\ShareErrorsFromSession::class,

\App\Http\Middleware\VerifyCsrfToken::class,

\Illuminate\Routing\Middleware\SubstituteBindings::class,

],

'api' => [

'throttle:60,1',

'bindings',

],

];

-Mặc định của Laravel đã có 2 nhóm middleware là web và api**.**

**-**Các bạn muốn đăng ký thêm middleware thì đăng ký theo cú pháp tương tự như mặc định của Laravel.

Và sử dụng trong Route với cú pháp:

Route::get('/', function () {

//

})->middleware('web');

Hoặc

Route::group(['middleware' => ['web']], function () {

//

});