# 建立Middleware

php artisan make:middleware middleware\_name

建立好的middleware會在App\Http\Middleware底下

    public function handle(Request $request, Closure $next): Response

    {

        return redirect('/');

        return $next($request);

    }

先設置好middleware會做的動作，這邊會將頁面重新導回到主頁面

    protected $middlewareAliases = [

        'auth' => \App\Http\Middleware\Authenticate::class,

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

        'auth.session' => \Illuminate\Session\Middleware\AuthenticateSession::class,

        'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,

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

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

        'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,

        'signed' => \App\Http\Middleware\ValidateSignature::class,

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

        'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,

        'role' => \App\Http\Middleware\RoleMiddleware::class,

    ];

到App\Http\kernal.php設置新增的middleware路徑，並且指派他名稱

Route::get('/admin/user/roles',[ 'middleware'=>'role' , function() {

    return "Middleware role";

}]);

到web.php中設置一個路徑，其中加入先前新建的middleware

原先執行的動作是當使用者連結到laravel-test.work/admin/user/roles時，會執行回傳Middleware role的字串，但是這邊有加入middleware，所以會先執行重新導向的動作。

# 使用Middleware

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

    /\*\*

     \* @var \App\Models\User $user

     \*/

    $user = Auth::user();

    if ($user->isAdmin()) {

        echo "this user is admin";

    }

});

/\*\* \* \*/是標記的動作，這將告訴 IDE，$user對象是一個App\Models\User類的實例，因此 IDE 知道應該對象工具有isAdmin()方法，並且不再顯示紅色底線。

    public function isAdmin() {

        if($this->role->name == 'Admin') {

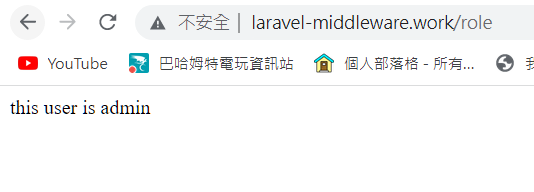
            return true;

        }

        return false;

    }

並且到User的模型中新增一個isAdmin()



# Middleware + Controller + web.php

Route::get('/admin', [AdminController::class, 'index']);

先建立admin的路徑，並且使用AdminController的index邏輯執行

class AdminController extends Controller

{

    public function \_\_construct()

    {

        $this->middleware('IsAdmin');

    }

    public function index()

    {

        return "You are an administrator because you are seeing this page";

    }

}

建構子會使用名稱為IsAdmin的middleware => 獲得true或false

index會回傳指定的字串

    public function handle(Request $request, Closure $next): Response

    {

        /\*\*

         \* @var \App\Models\User $user

         \*/

        $user = Auth::user();

        if ($user->isAdmin()) {

            return $next($request);

            // return redirect()->intended('/admin');

        }

        return redirect('/');

    }

middleware則根據使用者的身分重新導向到不同的頁面，若是admin則會回傳字串，若是其他使用者會回傳主頁面