Request -> Middleware -> Response

Vamos checar o roleto do usuário

rafael.franco@NOTWEB-O1 MINGW64 /e/PROGRAMAS/www/code-delivery-laravel-ionic

$ php artisan make:middleware CheckRole

Middleware created successfully.

Método handle

**public function** handle($request, Closure $next)  
{  
 **echo "Rodo Middleware"**;  
 **return** $next($request);  
}

Registrar o nome do Middleware em Kernel.php

**protected $routeMiddleware** = [  
 **'auth'** => \CodeDelivery\Http\Middleware\Authenticate::***class***,  
 **'auth.basic'** => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::***class***,  
 **'guest'** => \CodeDelivery\Http\Middleware\RedirectIfAuthenticated::***class***,  
 **'auth.checkrole'** => \CodeDelivery\Http\Middleware\CheckRole::***class***,  
];

Colocando ele nas rotas

Route::*group*([**'prefix'**=>**'admin'**, **'middleware'** => **'auth.checkrole'**, **'as'** => **'admin.'**], **function**() {

Ele vai ser executado em todas as requisições

Vamos utilizar a façade Atuh::check do laravel

CheckRole.php

**<?php  
  
namespace** CodeDelivery\Http\Middleware;  
  
**use** Closure;  
**use** Illuminate\Support\Facades\Auth;  
  
**class** CheckRole  
{  
 */\*\*  
 \* Handle an incoming request.  
 \*  
 \** ***@param*** *\Illuminate\Http\Request $request  
 \** ***@param*** *\Closure $next  
 \** ***@return*** *mixed  
 \*/* **public function** handle($request, Closure $next)  
 {  
 **if**(!Auth::*check*())  
 {  
 **return** redirect(**'/auth/login'**);  
 }  
 **return** $next($request);  
 }  
}