Uso de gates en controladores:

<?php

namespace App\Http\Controllers;

use App\Models\Comment;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Gate;

class CommentController extends Controller

{

// ...

public function update(Request $request, Comment $comment)

{

// Opción 1: Usando el Gate Facade

if (! Gate::allows('edit-comment', $comment)) {

abort(403);

}

// Opción 2: Usando el método authorize del controlador (más común)

$this->authorize('edit-comment', $comment);

// ... Lógica para actualizar el comentario

}

}

Uso de gates en rutas protegiéndolas usando el middleware can:

// Solo los administradores pueden acceder a las rutas dentro de este grupo

Route::middleware(['auth', 'can:access-admin-panel'])->group(function () {

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

return 'Bienvenido al panel de administración';

});

// Aquí puedes añadir más rutas de administración

});

// Solo los administradores pueden crear posts

Route::get('/posts/create', [PostController::class, 'create'])->middleware('can:create-post');

Uso de gates en vistas blade para ocultar o mostrar objetos:

{{-- Mostrar un enlace al panel de administración solo a los administradores --}}

@can('access-admin-panel')

<a href="/admin">Panel de Administración</a>

@endcan

{{-- Mostrar el botón de crear post solo a los administradores --}}

@can('create-post')

<a href="{{ route('posts.create') }}">Crear un nuevo Post</a>

@endcan

{{-- Dentro de un bucle que muestra comentarios --}}

@foreach ($post->comments as $comment)

<p>{{ $comment->content }}</p>

{{-- Permitir editar solo al autor o a un admin --}}

@can('edit-comment', $comment)

<a href="{{ route('comments.edit', $comment) }}">Editar</a>

@endcan

@endforeach