Manual para la gestión de los Roles y Permisos en Laravel.

1.-Instalar Laravel Collective

Laravel collective

composer require "laravelcollective/html":"^5.4.0"

Agregar

'providers' => [

// ...

Collective\Html\HtmlServiceProvider::class,

// ...

],

'aliases' => [

// ...

'Form' => Collective\Html\FormFacade::class,

'Html' => Collective\Html\HtmlFacade::class,

// ...

],

2.-Instalar SHINOBI

composer require caffeinated/shinobi

Agregar

Service Provider

Caffeinated\Shinobi\ShinobiServiceProvider::class

3.-Configurar el largo de los emails.

use Illuminate\Support\Facades\Schema;

class AppServiceProvider extends ServiceProvider

{

/\*\*

\* Bootstrap any application services.

\*

\* @return void

\*/

public function boot()

{

Schema::defaultStringLength(191);

}

/\*\*

\* Register any application services.

\*

\* @return void

\*/

public function register()

{

//

}

}

4.-Crear la tabla de Seeder para los Permisos.

Php artisan make:seeder PermissionsTableSeeder -> Importante

5.-Crear todos los permisos de la tabla.

[Link->archivo->TableSeeder.](PermissionsTableSeeder.php)

Ojo las rutas tienen que coincidir.

6.-Se activan en el Database Seeder uno por uno los que se hagan

public function run()

{

$this->call(PermissionsTableSeeder::class);

$this->call(UsersTableSeeder::class);

$this->call(ProductsTableSeeder::class);

}

Php artisan migrate:refresh --seed

Recordar que los permisos se dan sobre la ruta.

->Se crean los archivos seeder de cada uno de los modulos.

Despues de generar todo el modelo

Php artisan make:model Product -a

Php artisan make:seeder ProductsTableSeeder

Lo mismo por cada módulo.

7.-Crear los archivos semilla de cada uno el especial es el de user por que se crea un rol por defecto.

[Link->Archivo.](UsersTableSeeder.php)

8.-Middleware y Permisos.

Php artisan make:auth

Se generan el sistema de Login.

Configurar el Middleware.

Ir a app->http->Kernel Muy importante.

/\*\*

\* The application's route middleware.

\*

\* These middleware may be assigned to groups or used individually.

\*

\* @var array

\*/

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,

'role' => \Caffeinated\Shinobi\Middleware\UserHasRole::class,

'permission' => \Caffeinated\Shinobi\Middleware\UserHasPermission::class,

];

Se agregan los dos de abajo.

[Link->archivo.](Kernel.php)

9.-Lista de Rutas

Las rutas deben de coincidir con la tabla Permision Seeder.

[Link->Archivo.](web.php)

10.-Restringir los links rutas.

<div class="collapse navbar-collapse" id="app-navbar-collapse">

<!-- Left Side Of Navbar -->

<ul class="nav navbar-nav">

@can('products.index')

<li class="nav-item">

<a class="nav-link" href="{{ route('products.index') }}">Productos</a>

</li>

@endcan

@can('users.index')

<li class="nav-item">

<a class="nav-link" href="{{ route('users.index') }}">Usuarios</a>

</li>

@endcan

@can('roles.index')

<li class="nav-item">

<a class="nav-link" href="{{ route('roles.index') }}">Roles</a>

</li>

@endcan

</ul>

[Link->Archivo.](app.blade.php)

11.-Para que los permisos jalen importante.

En app->user.php

Agregar el ShinobiTrait sino no funciona.

use Caffeinated\Shinobi\Traits\ShinobiTrait;

use Illuminate\Notifications\Notifiable;

use Illuminate\Foundation\Auth\User as Authenticatable;

class User extends Authenticatable

{

use Notifiable, ShinobiTrait;

/\*\*

\* The attributes that are mass assignable.

\*

\* @var array

\*/

protected $fillable = [

'name', 'email', 'password',

];