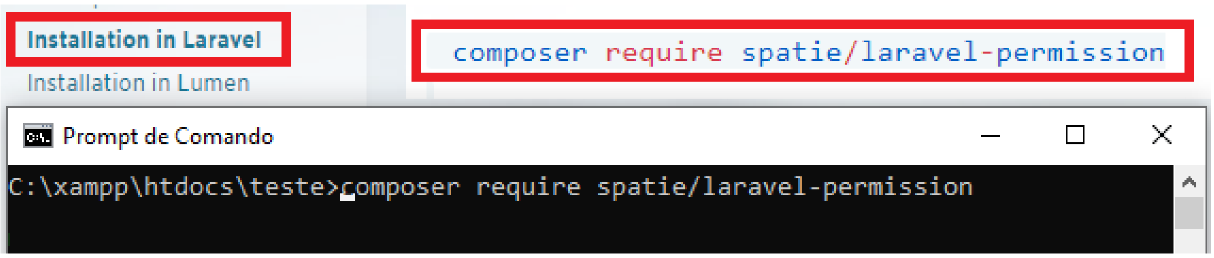
Site: https://spatie.be/docs/laravel-permission/v5/introduction

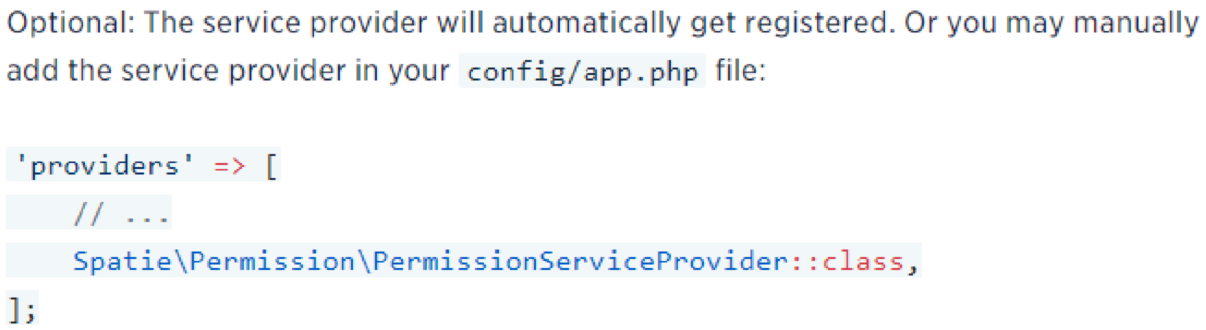
1 - Instala o pacote:composer require spatie/laravel-permission



```
composer require spatie/laravel-permission
```

```
C:\xampp\htdocs\teste>composer require spatie/laravel-permission
```

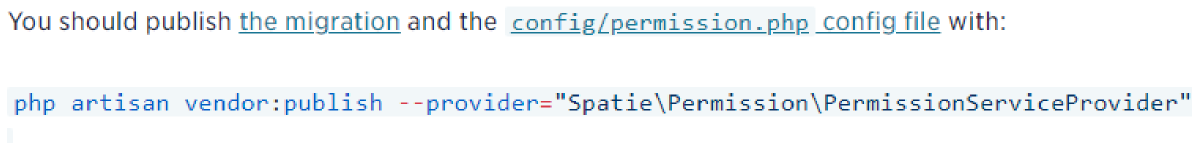
2 - Vai no arquivo “config/app.php” e adiciona no providers: “ Spatie\Permission\PermissionServiceProvider::class,”



Optional: The service provider will automatically get registered. Or you may manually add the service provider in your `config/app.php` file:

```
'providers' => [
    // ...
    Spatie\Permission\PermissionServiceProvider::class,
];
```

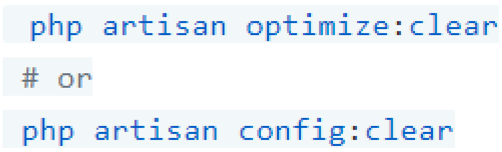
3 - Utiliza o comando php artisan vendor:publish --provider="Spatie\Permission\PermissionServiceProvider" - para que seja gerado o arquivo de configuração e as migrations;



You should publish the migration and the `config/permission.php` config file with:

```
php artisan vendor:publish --provider="Spatie\Permission\PermissionServiceProvider"
```

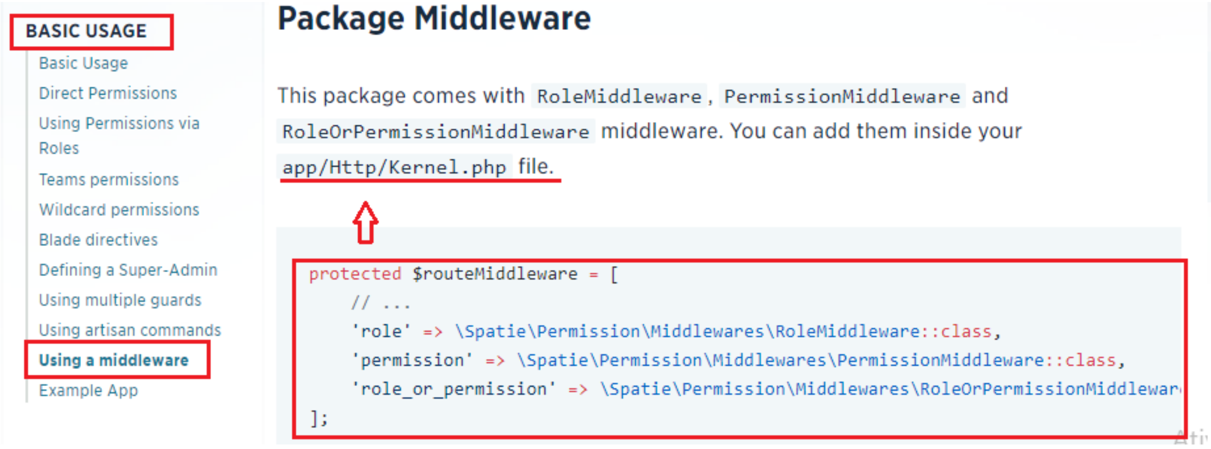
4 - Realize uma limpeza no cache: php artisan config:clear



```
 php artisan optimize:clear
# or
php artisan config:clear
```

5 - Executa as migrations: php artisan migrate

6 - Configurando o arquivo Kernel para utilizar as permissões a partir de middleware

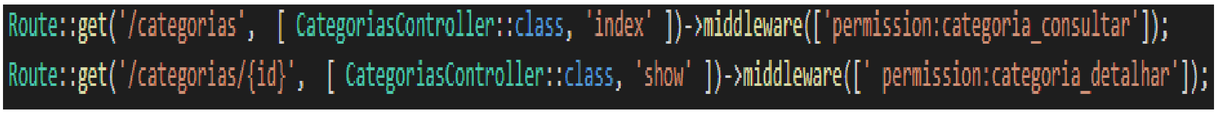


Package Middleware

This package comes with `RoleMiddleware`, `PermissionMiddleware` and `RoleOrPermissionMiddleware` middleware. You can add them inside your `app/Http/Kernel.php` file.

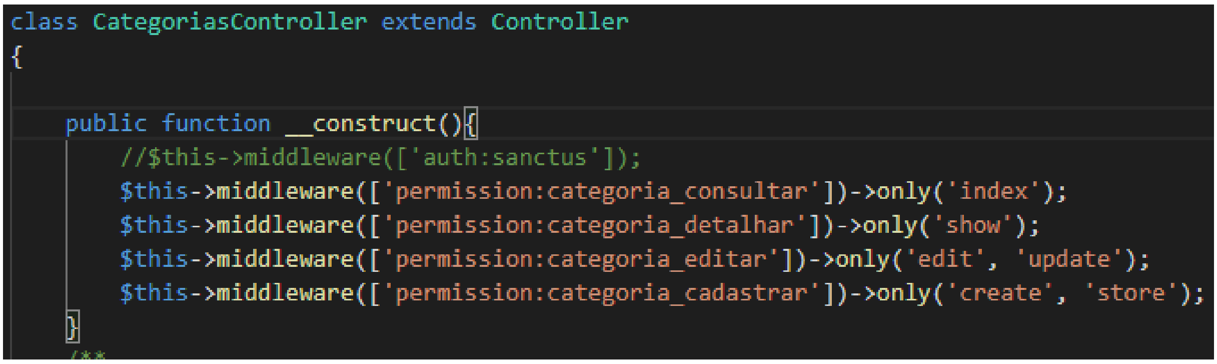
```
protected $routeMiddleware = [
    // ...
    'role' => \Spatie\Permission\Middlewares\RoleMiddleware::class,
    'permission' => \Spatie\Permission\Middlewares\PermissionMiddleware::class,
    'role_or_permission' => \Spatie\Permission\Middlewares\RoleOrPermissionMiddlewar
];
```

7 - Configuração para usar o Middleware de permissão na rota:



```
Route::get('/categorias', [ CategoriasController::class, 'index' ])->middleware(['permission:categoria_consultar']);
Route::get('/categorias/{id}', [ CategoriasController::class, 'show' ])->middleware([' permission:categoria_detalhar']);
```

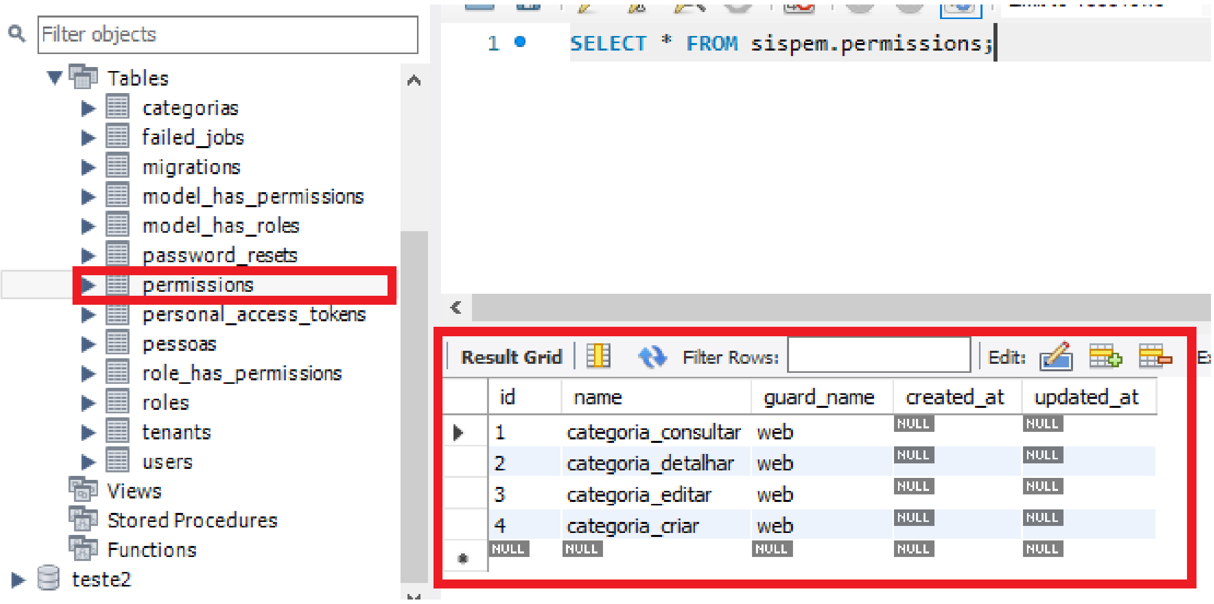
8 - Configuração para usar o Middleware de permissão no controller:



```
class CategoriasController extends Controller
{

    public function __construct(){
        //$this->middleware(['auth:sanctus']);
        $this->middleware(['permission:categoria_consultar'])->only('index');
        $this->middleware(['permission:categoria_detalhar'])->only('show');
        $this->middleware(['permission:categoria_editar'])->only('edit', 'update');
        $this->middleware(['permission:categoria_cadastrar'])->only('create', 'store');
    }
    /**
```

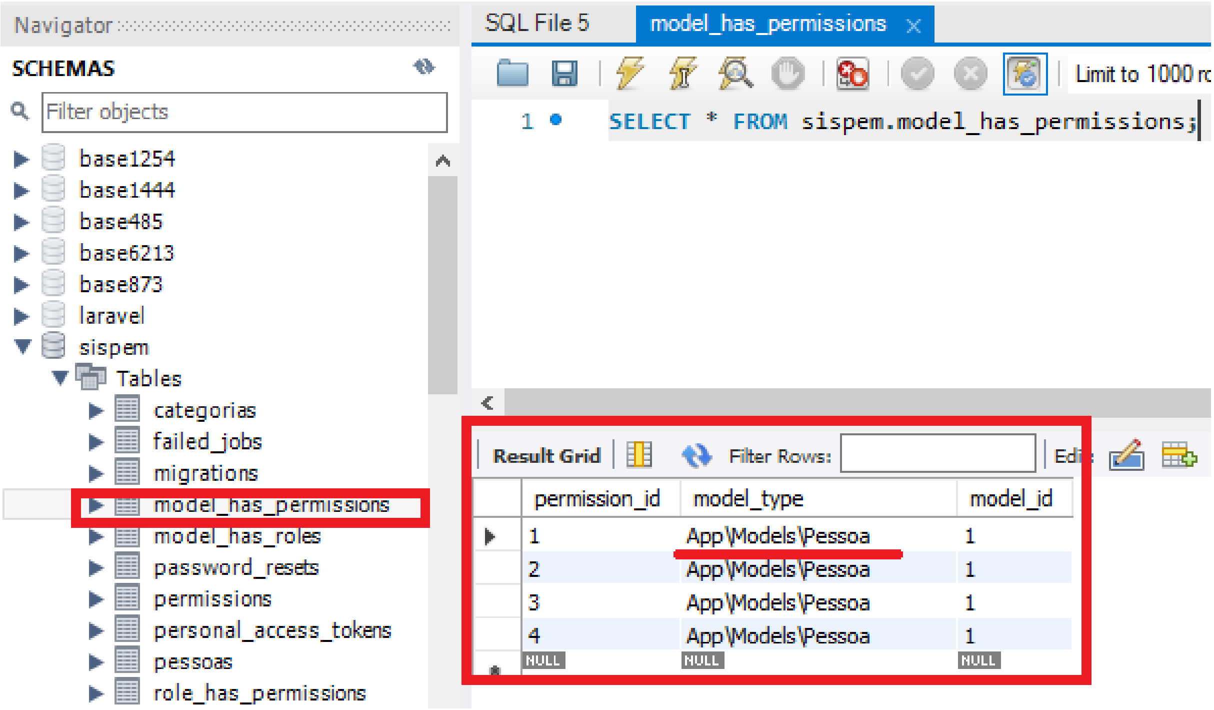
9 - Criar as permissões na tabela permissions do banco de dados:



```sql
SELECT * FROM sispem.permissions;
```

id	name	guard_name	created_at	updated_at
1	categoria_consultar	web	NULL	NULL
2	categoria_detalhar	web	NULL	NULL
3	categoria_editar	web	NULL	NULL
4	categoria_criar	web	NULL	NULL
NULL	NULL	NULL	NULL	NULL

10 - Cria na tabela model_has_permissions os relacionamento das permissões com os models

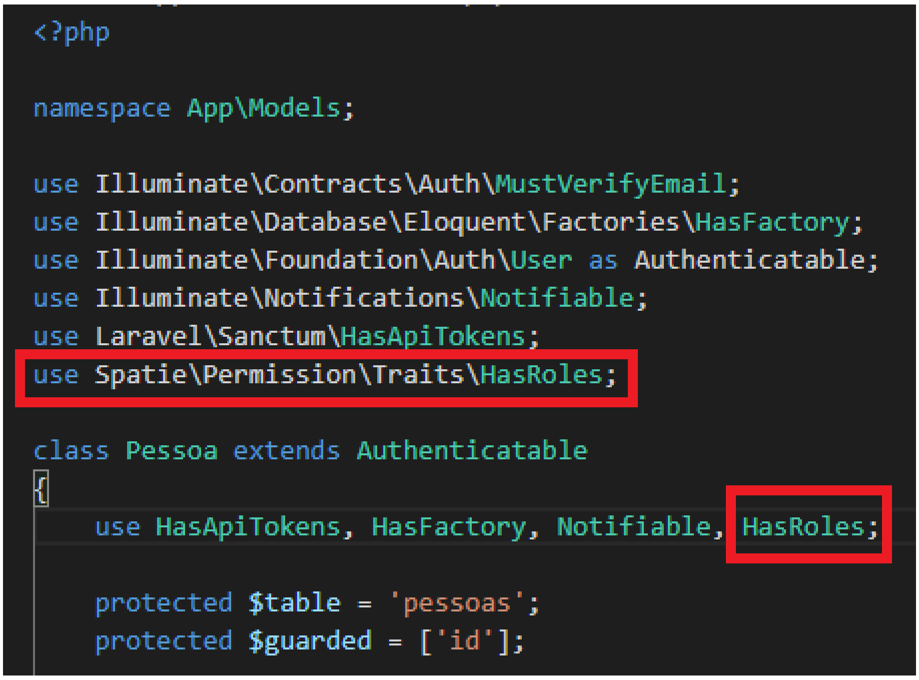


```sql
SELECT * FROM sispem.model_has_permissions;
```

permission_id	model_type	model_id
1	App\Models\Pessoa	1
2	App\Models\Pessoa	1
3	App\Models\Pessoa	1
4	App\Models\Pessoa	1
NULL	NULL	NULL

model_id é o ID do usuário na tabela, model_type é o model de usuário

Acrescenta no model de usuário



```php
<?php

namespace App\Models;

use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;
use Spatie\Permission\Traits\HasRoles;

class Pessoa extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable, HasRoles;

    protected $table = 'pessoas';
    protected $guarded = ['id'];
```