

Step for Laravel - User Roles and Permissions Management

-----***-----

- **Step 1:** Laravel 12 Installation
- **Step 2:** Add UserController and routing
- **Step 3:** Create Product Migration
- **Step 4:** Create Models
- **Step 5:** Create RoleController and routing
- **Step 6:** Create Authentication
- **Step 7:** Add Package Middleware
- **Step 8:** Add Controllers
- **Step 9:** Add Blade Files
- **Step 10:** Create Seeder For Permissions and AdminUser

1. Laravel 12 Installation

a. [Creating an Application]

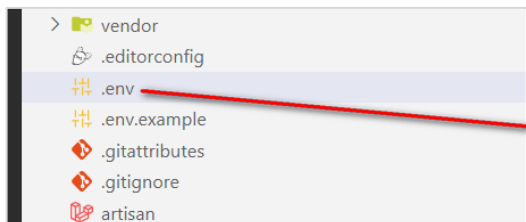
`laravel new` **Laravel_User_Roles_Permissions_mgm**

```
user@MSI MINGW64 /d/wamp64/www/dev/laravel12
$ laravel new Laravel_User_Roles_Permissions_mgm

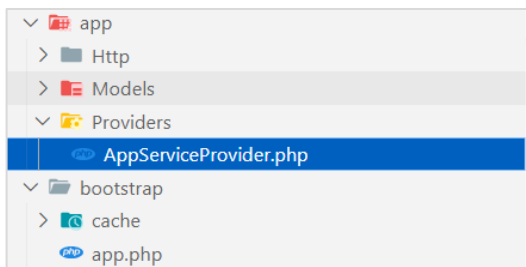
Laravel

Creating a "laravel/laravel" project at "./Laravel_User_Roles_Permissions_mgm"
Installing laravel/laravel (v12.0.7)
```

b. [Migrate database]

	<pre>23 DB_CONNECTION=mysql 24 DB_HOST=127.0.0.1 25 DB_PORT=3306 26 DB_DATABASE=laravel_user_roles_permissions_mgm 27 DB_USERNAME=root 28 DB_PASSWORD=123456</pre>
---	--

Before run `php artisan migrate` go to set **AppServiceProvider.php**

	<pre>app > Providers > AppServiceProvider.php > AppServiceProvider 20 21 public function boot(): void 22 { 23 Schema::defaultStringLength(250); 24 } 25 }</pre>
---	--

Then **php artisan migrate**

```
user@MSI MINGW64 /d/wamp64/www/dev/laravel12/Laravel_User_Roles_Permissions_mgm
$ php artisan migrate

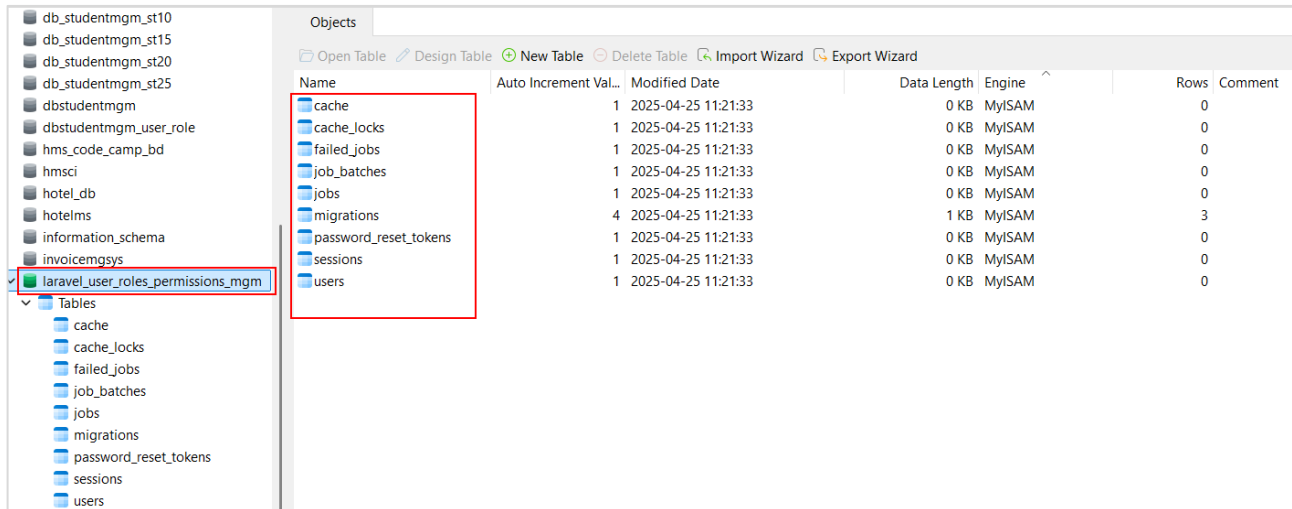
INFO Preparing database.

Creating migration table ..... 12.45ms DONE

INFO Running migrations.

0001_01_01_000000_create_users_table ..... 56.90ms DONE
0001_01_01_000001_create_cache_table ..... 11.15ms DONE
0001_01_01_000002_create_jobs_table ..... 47.86ms DONE
```

Check your database



Name	Auto Increment Val.	Modified Date	Data Length	Engine	Rows	Comment
cache	1	2025-04-25 11:21:33	0 KB	MyISAM	0	
cache_locks	1	2025-04-25 11:21:33	0 KB	MyISAM	0	
failed_jobs	1	2025-04-25 11:21:33	0 KB	MyISAM	0	
job_batches	1	2025-04-25 11:21:33	0 KB	MyISAM	0	
jobs	1	2025-04-25 11:21:33	0 KB	MyISAM	0	
migrations	4	2025-04-25 11:21:33	1 KB	MyISAM	3	
password_reset_tokens	1	2025-04-25 11:21:33	0 KB	MyISAM	0	
sessions	1	2025-04-25 11:21:33	0 KB	MyISAM	0	
users	1	2025-04-25 11:21:33	0 KB	MyISAM	0	

c. Install and configure basic frontend scaffolding provided by the **laravel/ui** package

Link: <https://github.com/laravel/ui>

composer require laravel/ui

```
user@MSI MINGW64 /d/wamp64/www/dev/laravel12/Laravel_User_Roles_Permissions_mgm
$ composer require laravel/ui
./composer.json has been updated
Running composer update laravel/ui
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
- Locking laravel/ui (v4.6.1)
```

d. [Generate login / registration]

php artisan ui bootstrap --auth (choose this For Bootstrap:)

php artisan ui vue --auth

php artisan ui react --auth

```
user@MSI MINGW64 /d/wamp64/www/dev/laravel12/Laravel_User_Roles_Permissions_mgm
$ php artisan ui bootstrap --auth

The [Controller.php] file already exists. Do you want to replace it? (yes/no) [yes]
> yes

INFO Authentication scaffolding generated successfully.
INFO Bootstrap scaffolding installed successfully.
WARN Please run [npm install && npm run dev] to compile your fresh scaffolding.
```

Before compiling your CSS, install your project's frontend dependencies using the [Node package manager \(NPM\)](#):

npm install

```
user@MSI MINGW64 /d/wamp64/www/dev/laravel12/Laravel_User_Roles_Permissions_mgm
$ npm install
added 172 packages, and audited 173 packages in 37s
```

And then running: **npm run build**

php artisan serv



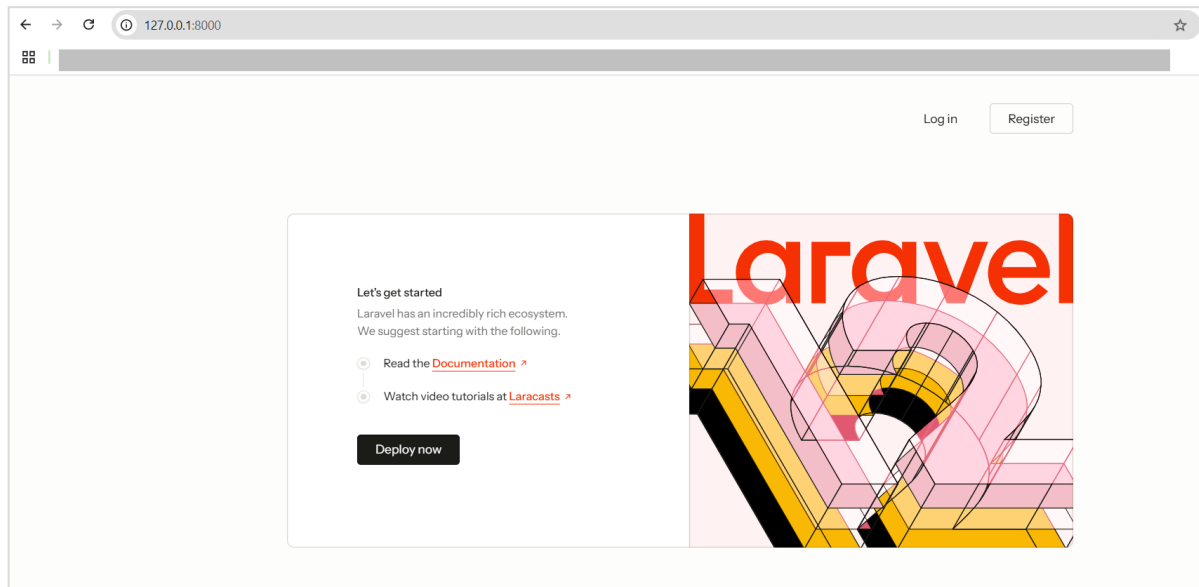
Remove



After



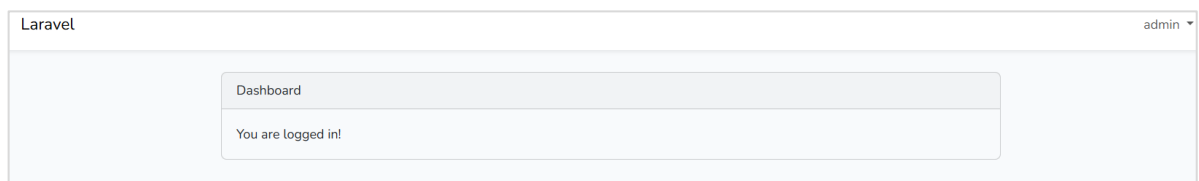
Check browser



Click on the register

A screenshot of the Laravel registration form. The form is titled "Register" and is located in the center of the page. It contains four input fields: "Name" (with the value "admin"), "Email Address" (with the value "admin@gmail.com"), "Password" (with masked characters "*****"), and "Confirm Password" (with masked characters "*****"). Below the "Confirm Password" field is a blue "Register" button. In the top right corner of the page, there are two links: "Login" and "Register". A red arrow points from the "Register" link to the "Register" button.

After registered



2. Add UserController

`php artisan make:controller UserController --resource`

```
user@MSI MINGW64 /d/wamp64/www/dev/laravel12/Laravel_User_Roles_Permissions_mgm
```

```
$ php artisan make:controller UserController --resource
```

```
INFO Controller [D:\wamp64\www\dev\laravel12\Laravel_User_Roles_Permissions_mgm\app\Http\Controllers\UserController.php] created successfully.
```

Creating a route **user**

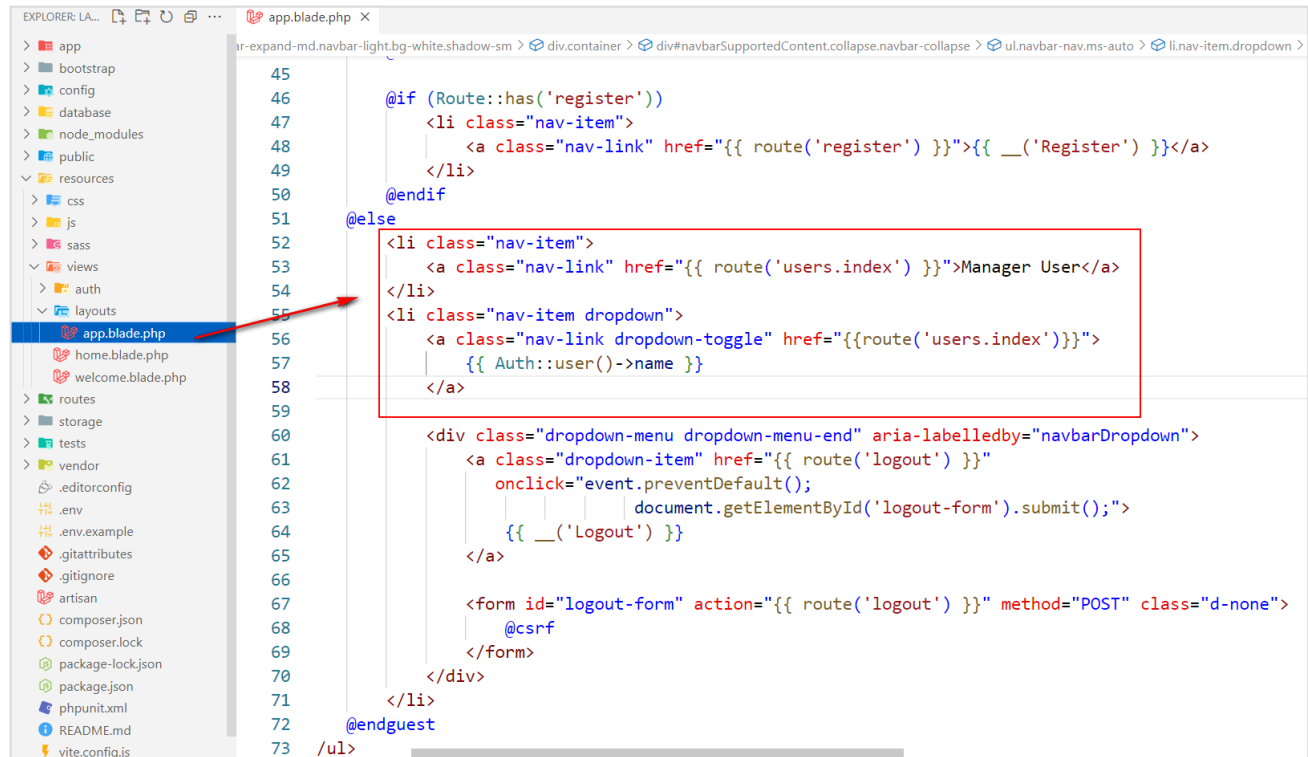
Route::resource('users', UserController::class);

```
user@MSI MINGW64 /d/wamp64/www/dev/laravel12/Laravel_User_Roles_Permissions_mgm
$ php artisan route:list --name='users'
```

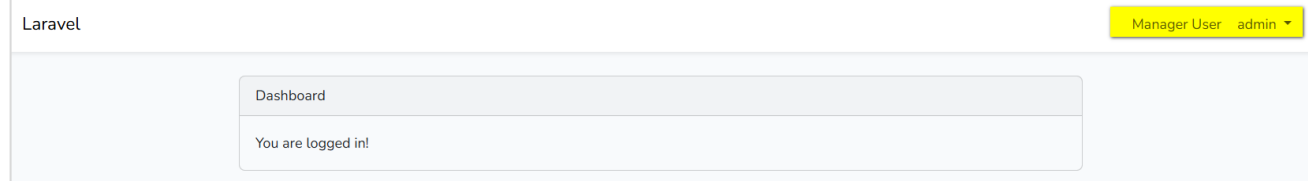
GET HEAD	users	users.index > UserController@index
POST	users	users.store > UserController@store
GET HEAD	users/create	users.create > UserController@create
GET HEAD	users/{user}	users.show > UserController@show
PUT PATCH	users/{user}	users.update > UserController@update
DELETE	users/{user}	users.destroy > UserController@destroy
GET HEAD	users/{user}/edit	users.edit > UserController@edit

Showing [7] routes

Custom **app.blade.php**



```
45
46
47     @if (Route::has('register'))
48         <li class="nav-item">
49             <a class="nav-link" href="{{ route('register') }}">{{ __('Register') }}</a>
50         </li>
51     @endif
52 @else
53     <li class="nav-item">
54         <a class="nav-link" href="{{ route('users.index') }}">Manager User</a>
55     </li>
56     <li class="nav-item dropdown">
57         <a class="nav-link dropdown-toggle" href="{{ route('users.index') }}">
58             {{ Auth::user()->name }}
59         </a>
60
61         <div class="dropdown-menu dropdown-menu-end" aria-labelledby="navbarDropdown">
62             <a class="dropdown-item" href="{{ route('logout') }}"
63                 onclick="event.preventDefault();
64                     document.getElementById('logout-form').submit();"
65             >{{ __('Logout') }}
66             </a>
67
68             <form id="logout-form" action="{{ route('logout') }}" method="POST" class="d-none">
69                 @csrf
70             </form>
71         </div>
72     </li>
73 @endguest
74 </ul>
```



Users Management

Laravel Manager User admin4 ▾

Users Management

Create New User

No	Name	Email	Roles	Action
1	admin	admin@gmail.com		<a>Show <a>Edit <a>Delete
2	admin4	admin4@gmail.com		<a>Show <a>Edit <a>Delete

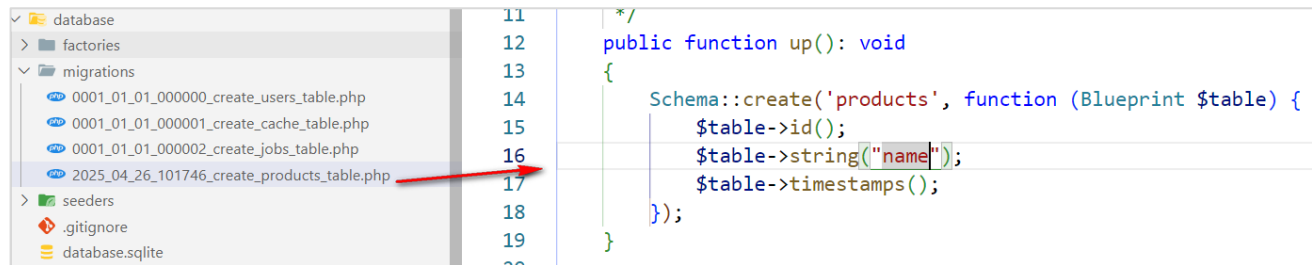
User-Role-Permission

3. Create Product Migration

php artisan make:migration create_products_table

```
user@MSI MINGW64 /d/wamp64/www/dev/laravel12/Laravel_User_Roles_Permissions_mgm
$ php artisan make:migration create_products_table

INFO Migration [D:\wamp64\www\dev\laravel12\Laravel_User_Roles_Permissions_mgm\database\migrations\2025_04_26_101746_create_products_table.php] created successfully.
```



php artisan migrate

```
user@MSI MINGW64 /d/wamp64/www/dev/laravel12/Laravel_User_Roles_Permissions_mgm
$ php artisan migrate

INFO Running migrations.

2025_04_26_101746_create_products_table ..... 110.55ms DONE
```

php artisan make:model Product

```
user@MSI MINGW64 /d/wamp64/www/dev/laravel12/Laravel_User_Roles_Permissions_mgm
$ php artisan make:model Product

INFO Model [D:\wamp64\www\dev\laravel12\Laravel_User_Roles_Permissions_mgm\app\Models\Product.php] created successfully.
```

php artisan make:controller ProductController -r

```
user@MSI MINGW64 /d/wamp64/www/dev/laravel12/Laravel_User_Roles_Permissions_mgm
$ php artisan make:controller ProductController -r

INFO Controller [D:\wamp64\www\dev\laravel12\Laravel_User_Roles_Permissions_mgm\app\Http\Controllers\ProductController.php] created successfully.
```

Check route product

```
user@MSI MINGW64 /d/wamp64/www/dev/laravel12/Laravel_User_Roles_Permissions_mgm
$ php artisan route:list --name='products'

GET|HEAD      products ..... products.index > ProductController@index
POST          products ..... products.store > ProductController@store
GET|HEAD      products/create ..... products.create > ProductController@create
GET|HEAD      products/{product} ..... products.show > ProductController@show
PUT|PATCH    products/{product} ..... products.update > ProductController@update
DELETE        products/{product} ..... products.destroy > ProductController@destroy
GET|HEAD      products/{product}/edit ..... products.edit > ProductController@edit

Showing [7] routes
```

Product Management

Laravel

Manage UserManage Productadmin4 ▾

Product created

Product Management

Create Product

No	Name	Email	Roles	Action
1	Fanta		<div>ShowEditDelete</div>	
2	Popie		<div>ShowEditDelete</div>	
3	Coca		<div>ShowEditDelete</div>	

User-Role-Permission

4. Create Models

Manage **user permissions** and **roles** in a database.

Spatie roles and permissions

link: <https://spatie.be/docs/laravel-permission/v6/introduction>

SPATIE

Vacancies About Blog Docs Guidelines Merch

Products Open Source Courses Web Development

Docs > Laravel-permission > Installation in Laravel

Search...

Version v6

Introduction
Support us
Prerequisites
Installation in Laravel
Installation in Lumen
Upgrading
Questions and issues
Changelog
About us

Basic Usage

Installation in Laravel

Laravel Version Compatibility

See the "Prerequisites" documentation page for compatibility details.

Installing

1. Consult the Prerequisites page for important considerations regarding your User models!
2. This package publishes a `config/permission.php` file. If you already have a file by that name, you must rename or remove it.
3. You can install the package via composer:

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

On this page
Laravel Version Compatibility
Installing
Default config file contents

Ray
Debug your applications faster
Learn more

Help us improve this page

`composer require spatie/laravel-permission`

```
user@MSI MINGW64 /d/wamp64/www/dev/laravel12/Laravel_User_Roles_Permissions_mgm
$ composer require spatie/laravel-permission
./composer.json has been updated
Running composer update spatie/laravel-permission
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
  - Locking spatie/laravel-permission (6.17.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
INFO: Could not find files for the given pattern(s).
  - Installing spatie/laravel-permission (6.17.0): Extracting archive
Generating optimized autoload files
```

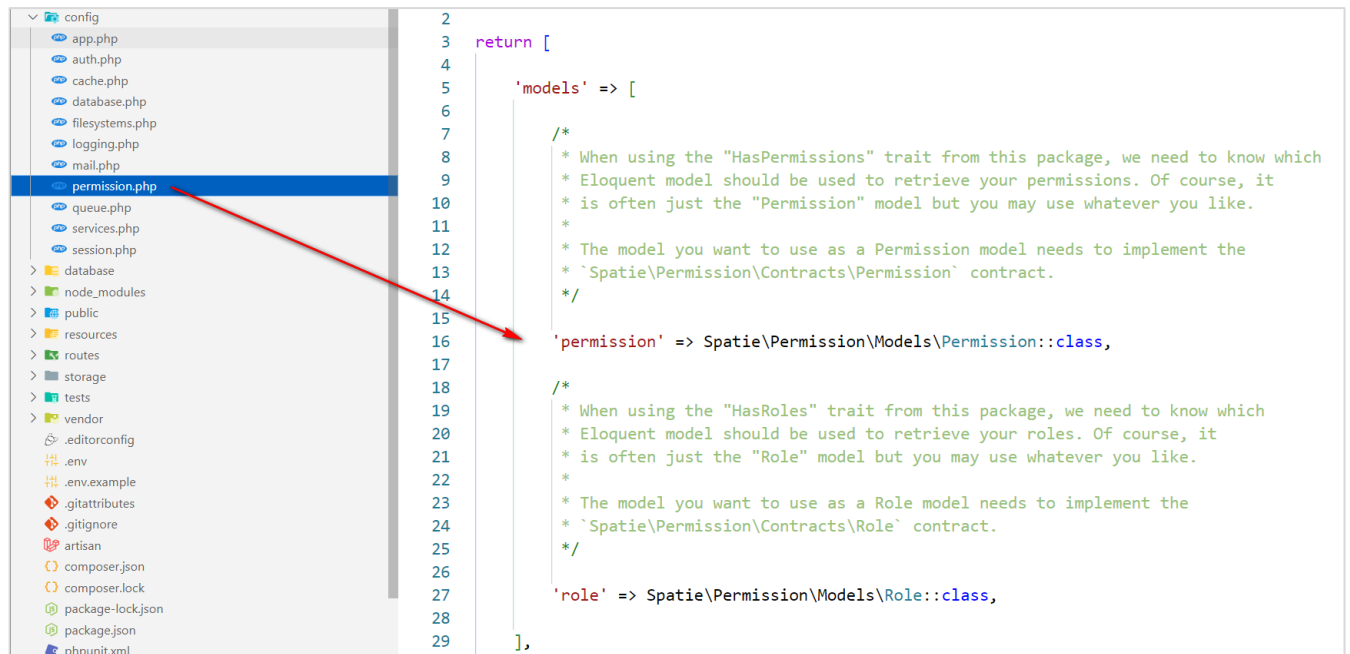
`php artisan vendor:publish --provider="Spatie\Permission\PermissionServiceProvider"`

```
user@MSI MINGW64 /d/wamp64/www/dev/laravel12/Laravel_User_Roles_Permissions_mgm
$ php artisan vendor:publish --provider="Spatie\Permission\PermissionServiceProvider"

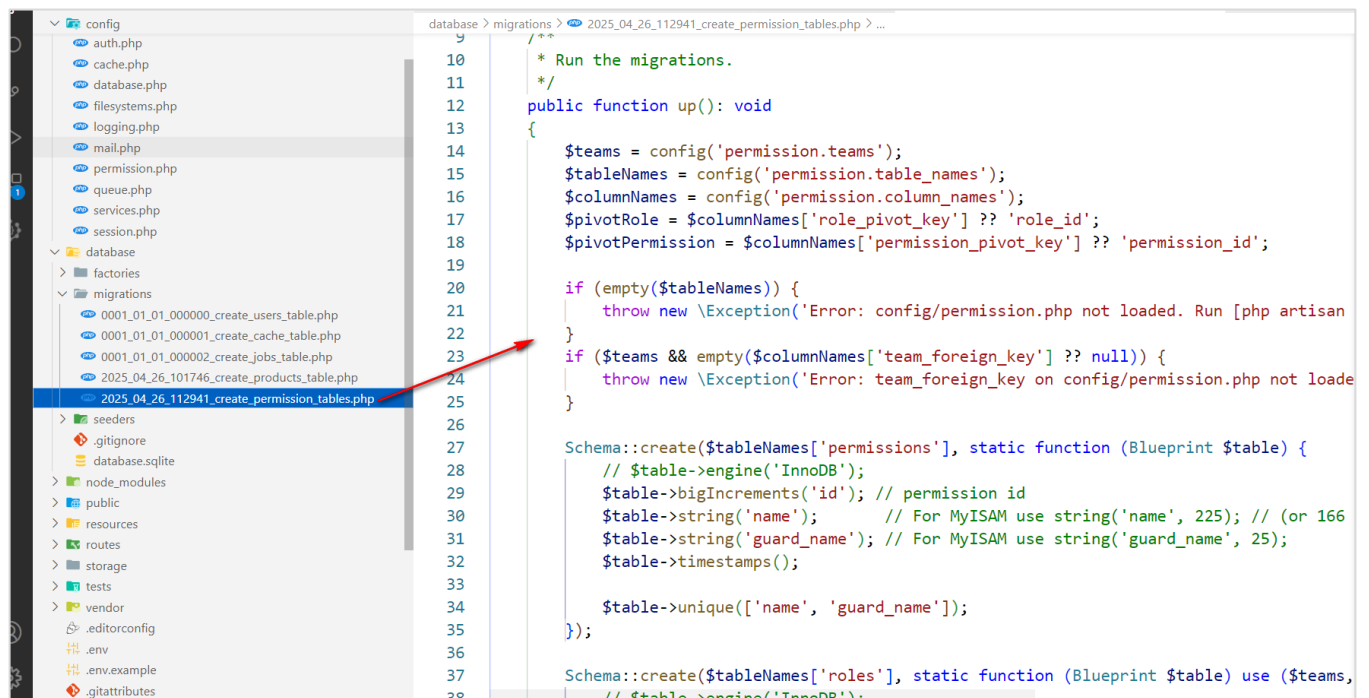
INFO Publishing assets.

Copying file [D:\wamp64\www\dev\laravel12\Laravel_User_Roles_Permissions_mgm\vendor\spatie\laravel-permission\config\permission.php] to [D:\wamp64\www\dev\laravel12\Laravel_User_Roles_Permissions_mgm\config\permission.php] DONE
Copying file [D:\wamp64\www\dev\laravel12\Laravel_User_Roles_Permissions_mgm\vendor\spatie\laravel-permission\database\Migrations\create_permission_tables.php.stub] to [D:\wamp64\www\dev\laravel12\Laravel_User_Roles_Permissions_mgm\database\Migrations\2025_04_26_112941_create_permission_tables.php] DONE
```

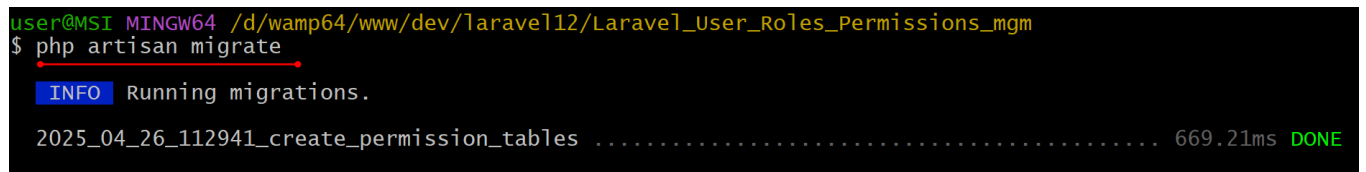
You should publish the migration and the **config/permission.php** config file with:



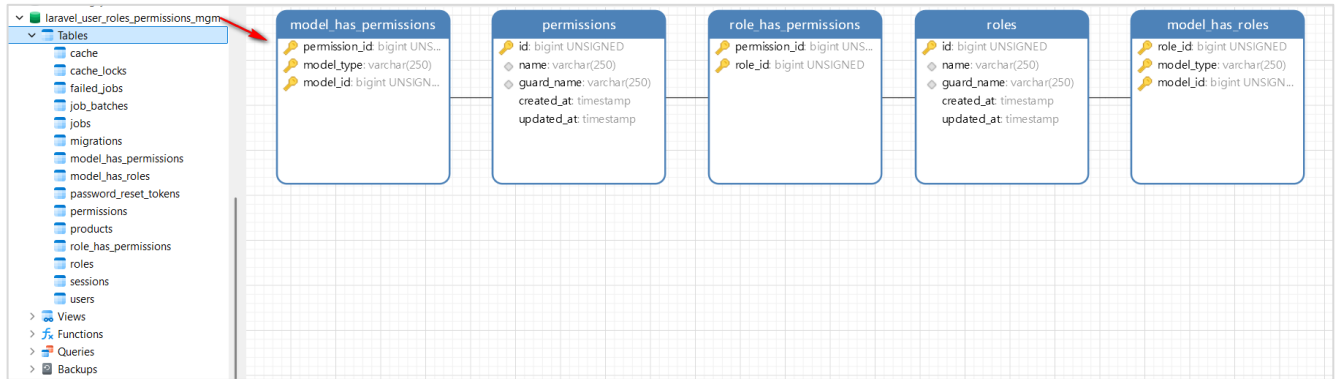
Migrations -> **2025_04_26_112941_create_permission_tables.php**



Migrate **permission** table



Check your database



Basic Usage

← → ↺ 🌐 spatie.be/docs/laravel-permission/v6/basic-usage/basic-usage

Basic Usage

- Basic Usage
- Direct Permissions
- Using Permissions via Roles
- Enums
- Teams permissions
- Wildcard permissions
- Blade directives
- Defining a Super-Admin
- Using multiple guards
- Artisan Commands
- Middleware
- Passport Client Credentials Grant usage
- Example App

Create A Permission

This package allows for users to be associated with permissions and roles. Every role is associated with multiple permissions. A `Role` and a `Permission` are regular Eloquent models. They require a `name` and can be created like this:

```
use Spatie\Permission\Models\Role;
use Spatie\Permission\Models\Permission;

$role = Role::create(['name' => 'writer']);
$permission = Permission::create(['name' => 'edit articles']);
```

Create permissionSeeder

php artisan make:seeder PermissionSeeder

```
user@MSI MINGW64 /d/wamp64/www/dev/laravel12/Laravel_User_Roles_Permissions_mgm
$ php artisan make:seeder PermissionSeeder

[INFO] Seeder [D:\wamp64\www\dev\laravel12\Laravel_User_Roles_Permissions_mgm\database\seeders\PermissionSeeder.php] created successfully.
```

use Spatie\Permission\Models\Permission;

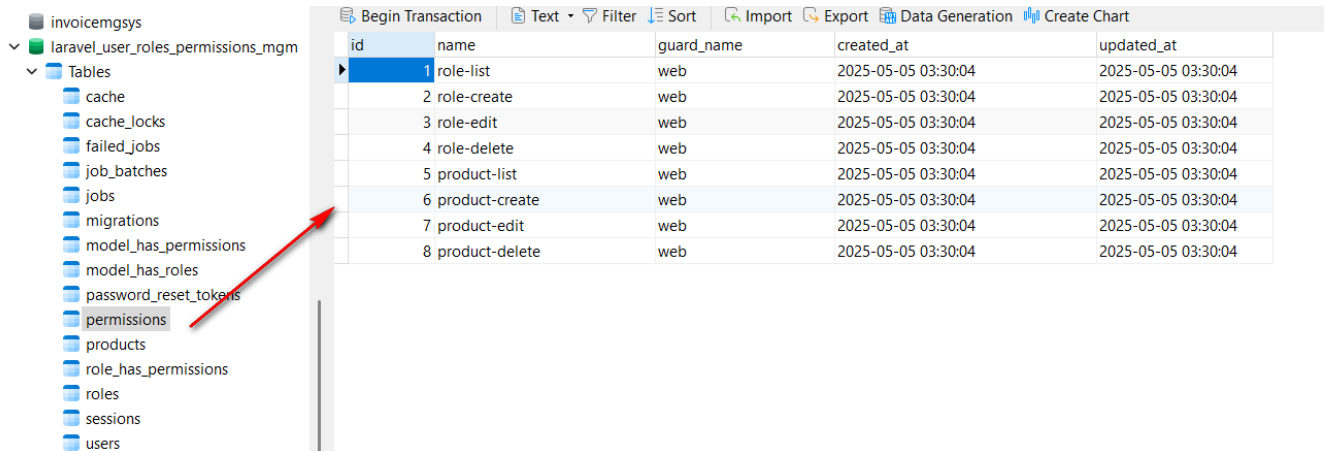
click link: <https://spatie.be/docs/laravel-permission/v6/basic-usage/basic-usage>

Cmd: `php artisan db:seed PermissionSeeder`

```
user@MSI MINGW64 /d/wamp64/www/dev/laravel12/Laravel_User_Roles_Permissions_mgm
$ php artisan db:seed PermissionSeeder
```

INFO Seeding database.

Check permissions table



id	name	guard_name	created_at	updated_at
1	role-list	web	2025-05-05 03:30:04	2025-05-05 03:30:04
2	role-create	web	2025-05-05 03:30:04	2025-05-05 03:30:04
3	role-edit	web	2025-05-05 03:30:04	2025-05-05 03:30:04
4	role-delete	web	2025-05-05 03:30:04	2025-05-05 03:30:04
5	product-list	web	2025-05-05 03:30:04	2025-05-05 03:30:04
6	product-create	web	2025-05-05 03:30:04	2025-05-05 03:30:04
7	product-edit	web	2025-05-05 03:30:04	2025-05-05 03:30:04
8	product-delete	web	2025-05-05 03:30:04	2025-05-05 03:30:04

5. Create RoleController and routing

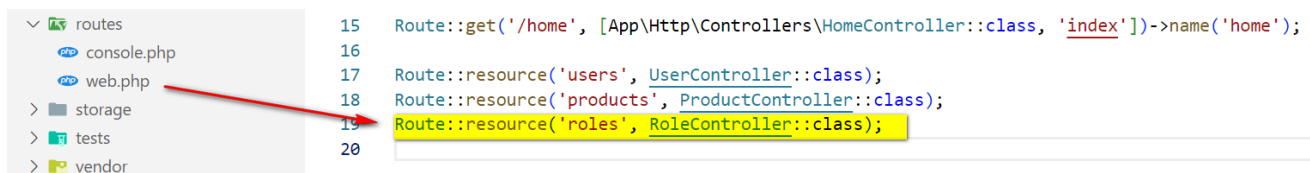
Cmd: `php artisan make:controller RoleController -r`

```
user@MSI MINGW64 /d/wamp64/www/dev/laravel12/Laravel_User_Roles_Permissions_mgm
$ php artisan make:controller RoleController -r
```

INFO Controller [D:\wamp64\www\dev\laravel12\Laravel_User_Roles_Permissions_mgm\app\Http\Controllers\RoleController.php] created successfully.

Create a route role

Route::resource('roles', RoleController::class);



```
15 Route::get('/home', [App\Http\Controllers\HomeController::class, 'index'])->name('home');
16
17 Route::resource('users', UserController::class);
18 Route::resource('products', ProductController::class);
19 Route::resource('roles', RoleController::class);
20
```

Check the route list

```
user@MSI MINGW64 /d/wamp64/www/dev/laravel12/Laravel_User_Roles_Permissions_mgm
$ php artisan route:list --name='roles'
```

GET HEAD	roles	roles.index	> RoleController@index
POST	roles	roles.store	> RoleController@store
GET HEAD	roles/create	roles.create	> RoleController@create
GET HEAD	roles/{role}	roles.show	> RoleController@show
PUT PATCH	roles/{role}	roles.update	> RoleController@update
DELETE	roles/{role}	roles.destroy	> RoleController@destroy
GET HEAD	roles/{role}/edit	roles.edit	> RoleController@edit

showing [7] routes

php

```
class RoleController extends Controller
{

    public function index()
    {
        $roles = Role::all();
        return view('roles.index', compact('roles'));
    }

    public function create()
    {
        $permissions = Permission::all();
        return view('roles.create', compact('permissions'));
    }

    public function store(Request $request)
    {
        //dd($request->all());

        $request->validate([
            'name' => 'required'
        ]);
        $role = Role::create(['name' => $request->name]);
        $role->syncPermissions([$request->permissions]);

        return redirect()->route('roles.index')->with('success', 'Role created');
    }

    public function show(string $id)
    {
        $role = Role::find($id);
        return view('roles.show', compact('role'));
    }

    public function edit(string $id)
    {
        $permissions = Permission::all();
        $role = Role::find($id);
        return view('roles.edit', compact('role', 'permissions'));
    }

    public function update(Request $request, string $id)
    {
        $request->validate([
            'name' => 'required'
        ]);
        $role = Role::find($id);
        $role->name = $request->name;
        $role->save();
        $role->syncPermissions([$request->permissions]);
        return redirect()->route('roles.index')->with('success', 'Role updated');
    }

    public function destroy(string $id)
    {
        $role = Role::find($id);
        $role->delete();
        return redirect()->route('roles.index')->with('success', 'Role deleted');
    }
}
```

6. Create Authentication

Let's check your URL

Laravel

Manage Role Manage User Manage Product admin4 ▾

Role created

Users Role

Create Role

No	Name	Action
1	Reader	Show Edit Delete
2	Admin	Show Edit Delete
3	Manager	Show Edit Delete

User-Role-Permission

Custom User page

Laravel

Manage Role **Manage User** Manage Product admin4 ▾

Users Management

Create New User

No	Name	Email	Roles	Action
1	admin	admin@gmail.com		Show Edit Delete
2	admin4	admin4@gmail.com		Show Edit Delete

User-Role-Permission

Assign role to user

Laravel

Manage Role Manage User Manage Product admin4 ▾

Create User

[Back](#)

Name:
Manager

The name field is required.

Email:
manager@gmail.com

The email field is required.

Password:
.....

Role:
---Choose a role---
Reader
Admin
Manager

[Submit](#)

Fix err

Internal Server Error

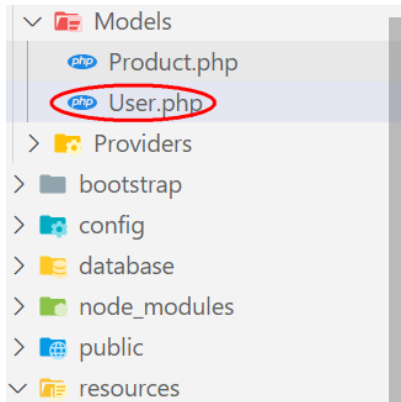
BadMethodCallException

Call to undefined method App\Models\User::syncRoles()

POST 127.0.0.1:8000

PHP 8.2.0 — Laravel 12.10.2

Add **HasRoles** to User model



```
1
2
3 namespace App\Models;
4
5 // use Illuminate\Contracts\Auth\MustVerifyEmail;
6 use Illuminate\Database\Eloquent\Factories\HasFactory;
7 use Illuminate\Foundation\Auth\User as Authenticatable;
8 use Illuminate\Notifications\Notifiable;
9 use Spatie\Permission\Traits\HasRoles;
10
11 class User extends Authenticatable
12 {
13     use HasFactory, Notifiable, HasRoles;
```

Go to create user again

Laravel Manage Role Manage User Manage Product Manager2 ▾

Users Management

[Create New User](#)

No	Name	Email	Roles	Action
1	Manager	manager@gmail.com	Reader Manager	Show Edit Delete
2	admin	admin@gmail.com	Reader Admin Manager	Show Edit Delete
3	Manager2	Manager2@gmail.com	Manager	Show Edit Delete

[User-Role-Permission](#)

Check **model_has_roles** table

laravel_user_roles_permissions_mgm

- Tables
 - cache
 - cache_locks
 - failed_jobs
 - job_batches
 - jobs
 - migrations
 - model_has_permissions
 - model_has_roles**
 - password_reset_tokens
 - permissions
 - products
 - role_has_permissions
 - roles
 - sessions
 - users

Objects users @laravel_user_roles_permissions... products @laravel_user_roles_permissi...

[Begin Transaction](#) [Text](#) [Filter](#) [Sort](#) [Import](#) [Export](#) [Data Generation](#)

role_id	model_type	model_id
1	App\Models\User	4
1	App\Models\User	5
3	App\Models\User	5
4	App\Models\User	5
1	App\Models\User	8
4	App\Models\User	8

Login to different user levels

Laravel Manage Role Manage User Manage Product Reader ▾

[Dashboard](#)

You are logged in!

Blade directives with product

Product Management

Create Product

No	Name	Action
1	Fanta	Show Edit Delete
2	Popie	Show Edit Delete
3	Coca	Show Edit Delete

User-Role-Permission

@can('product-create')

//

@endcan

```
@extends('layouts.app')
@section('content')
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-lg-12 margin-tb">
                <div class="pull-left">
                    <h2>Product Management</h2>
                </div>
                <div class="pull-right">
                    @can('product-create')
                        <a class="btn btn-success" href="{{ route('products.create') }}"> Create Product</a>
                    @endcan
                </div>
            </div>
        </div>
        <table class="table table-bordered">
            <tr>
                <th>No</th>
                <th>Name</th>
                <th width="280px">Action</th>
            </tr>
            @foreach ($products as $key => $product)
                <tr>
                    <td>{{ ++$key }}</td>
                    <td>{{ $product->name }}</td>
                    <td>
                        <form action="{{ route('products.destroy', $product->id) }}" method="POST">
                            @csrf
                            @method('delete')

                            @can('product-list')
                                <a class="btn btn-info btn-sm" href="{{ route('products.show', $product->id) }}">Show</a>
                            @endcan

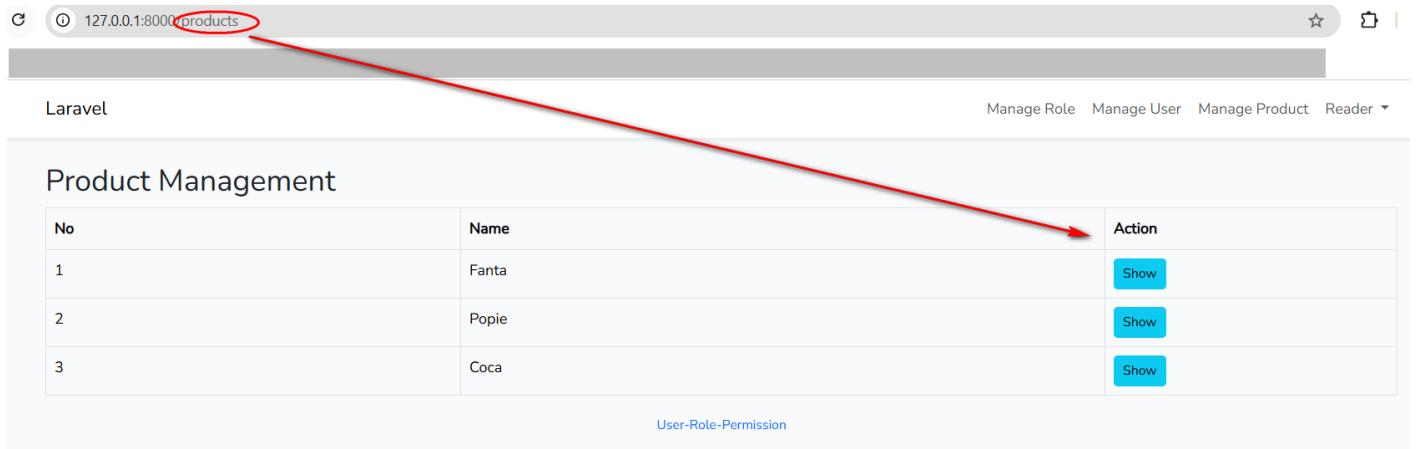
                            @can('product-edit')
                                <a class="btn btn-primary btn-sm" href="{{ route('products.edit', $product->id) }}">Edit</a>
                            @endcan

                            @can('product-delete')
                                <button class="btn btn-danger btn-sm" onclick="return confirm('Are you sure to delete it?')" href="{{ route('products.destroy', $product->id) }}">Delete</button>
                            @endcan
                        </form>
                    </td>
                </tr>
            @endforeach
        </table>

        {{-- {!! $data->render() !!} --}}

        <p class="text-center text-primary"><small>User-Role-Permission</small></p>
    </div>
@endsection
```

After assigned a role to access the **product** action



Laravel

Manage Role Manage User Manage Product Reader ▾

Product Management

No	Name	Action
1	Fanta	Show
2	Popie	Show
3	Coca	Show

User-Role-Permission

7. Add Package Middleware

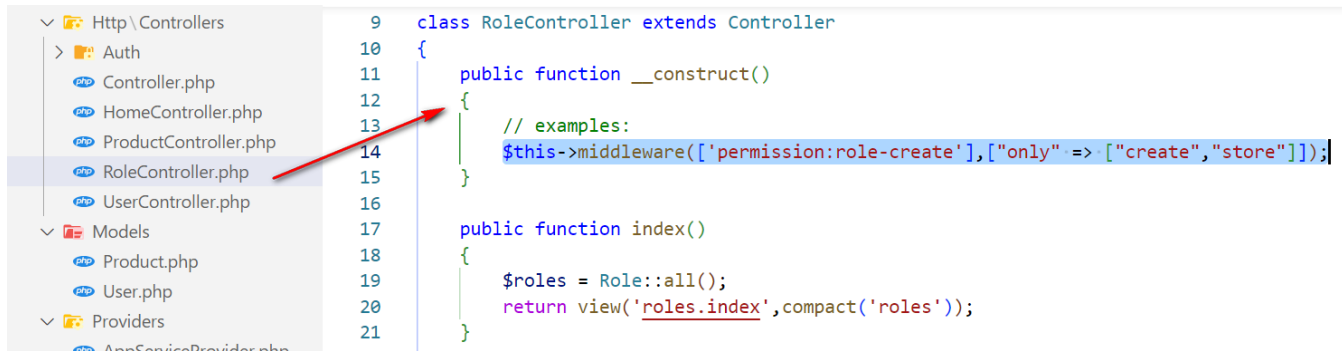
In Laravel 12 open **/bootstrap/app.php** and register them there:

```
$middleware->alias([
    'role' => \Spatie\Permission\Middleware\RoleMiddleware::class,
    'permission' => \Spatie\Permission\Middleware\PermissionMiddleware::class,
    'role_or_permission' => \Spatie\Permission\Middleware\RoleOrPermissionMiddleware::class,
]);
```



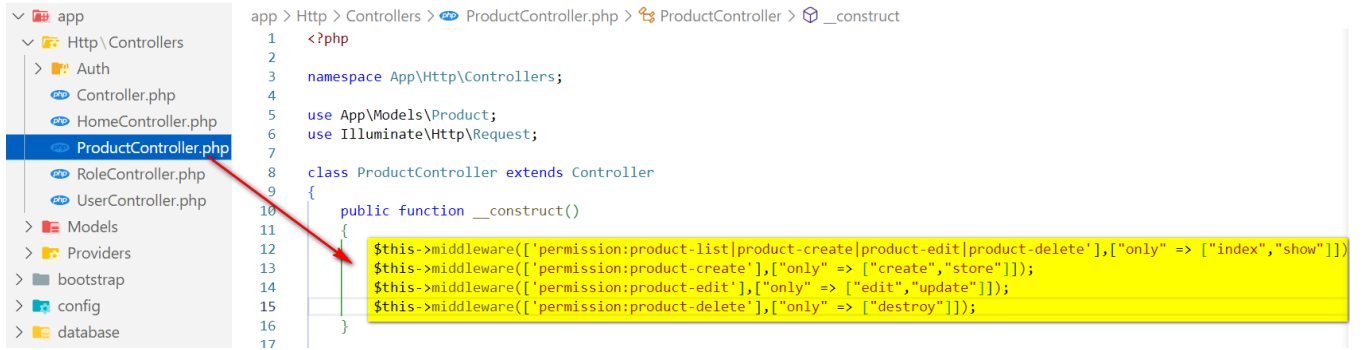
```
7 return Application::configure(basePath: dirname(__DIR__))
8     ->withRouting(
9         web: __DIR__.'/../routes/web.php',
10        commands: __DIR__.'/../routes/console.php',
11        health: '/up',
12    )
13    ->withMiddleware(function (Middleware $middleware) {
14        $middleware->alias([
15            'role' => \Spatie\Permission\Middleware\RoleMiddleware::class,
16            'permission' => \Spatie\Permission\Middleware\PermissionMiddleware::class,
17            'role_or_permission' => \Spatie\Permission\Middleware\RoleOrPermissionMiddleware::class,
18        ]);
19    });
20    ->withExceptions(function (Exceptions $exceptions) {
21        //
22    })->create();
23
```

```
$this->middleware(['permission:role-create'],["only" => ["create","store"]]);
```



```
9 class RoleController extends Controller
10 {
11     public function __construct()
12     {
13         // examples:
14         $this->middleware(['permission:role-create'],["only" => ["create","store"]]);
15     }
16
17     public function index()
18     {
19         $roles = Role::all();
20         return view('roles.index',compact('roles'));
21     }
22 }
```

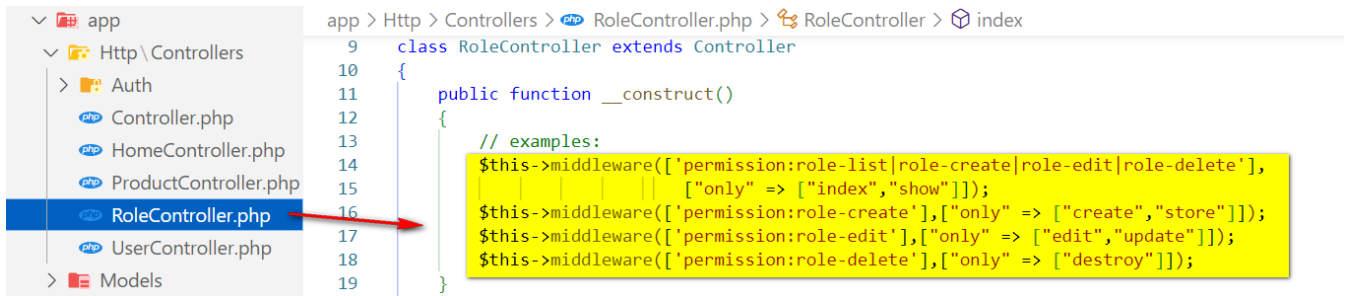
Add middleware to the product



The screenshot shows the Laravel IDE with the file explorer on the left highlighting `ProductController.php`. The main editor shows the code for `ProductController` extending `Controller`. The `__construct` method is highlighted with a yellow box, containing the following middleware assignments:

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Product;
6 use Illuminate\Http\Request;
7
8 class ProductController extends Controller
9 {
10     public function __construct()
11     {
12         $this->middleware(['permission:product-list|product-create|product-edit|product-delete'],["only" => ["index","show"]]);
13         $this->middleware(['permission:product-create'],["only" => ["create","store"]]);
14         $this->middleware(['permission:product-edit'],["only" => ["edit","update"]]);
15         $this->middleware(['permission:product-delete'],["only" => ["destroy"]]);
16     }
17 }
```

Add middleware to the role



The screenshot shows the Laravel IDE with the file explorer on the left highlighting `RoleController.php`. The main editor shows the code for `RoleController` extending `Controller`. The `__construct` method is highlighted with a yellow box, containing the following middleware assignments:

```
9 class RoleController extends Controller
10 {
11     public function __construct()
12     {
13         // examples:
14         $this->middleware(['permission:role-list|role-create|role-edit|role-delete'],
15             ["only" => ["index","show"]]);
16         $this->middleware(['permission:role-create'],["only" => ["create","store"]]);
17         $this->middleware(['permission:role-edit'],["only" => ["edit","update"]]);
18         $this->middleware(['permission:role-delete'],["only" => ["destroy"]]);
19     }
20 }
```

Laravel

Manage Role Manage User Manage Product **admin2**

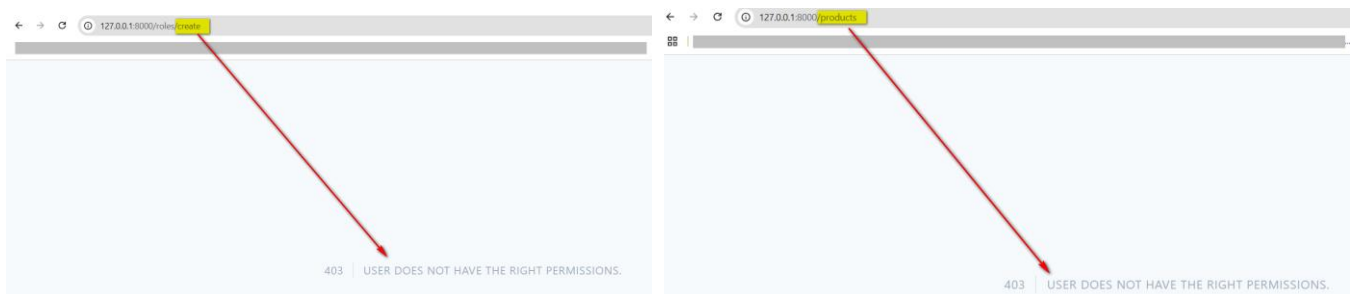
Users Management

Create New User

No	Name	Email	Roles	Action
1	admin2	admin2@gmail.com		Show Edit Delete
2	Manager	manager@gmail.com	Reader Manager	Show Edit Delete
3	admin	admin@gmail.com	Reader Admin Manager	Show Edit Delete
4	Reader	reader@gmail.com	Reader	Show Edit Delete

User-Role-Permission

Check route user doesn't have permission



The image shows two browser screenshots. The left screenshot shows a 403 error message: "403 | USER DOES NOT HAVE THE RIGHT PERMISSIONS." The right screenshot shows a 403 error message: "403 | USER DOES NOT HAVE THE RIGHT PERMISSIONS." Both screenshots have a red arrow pointing to the error message.

Example Assign to user admin2

Laravel Manage Role Manage User Manage Product admin2 ▾

User updated

Users Management

Create New User

No	Name	Email	Roles	Action
1	admin2	admin2@gmail.com	Reader	Show Edit Delete
2	Manager	manager@gmail.com	Reader Manager	Show Edit Delete
3	admin	admin@gmail.com	Reader Admin Manager	Show Edit Delete
4	Reader	reader@gmail.com	Reader	Show Edit Delete

User-Role-Permission

Show Role

Back

Role: Reader

Permission:

- ☒ role-list
- ☒ role-edit
- ☒ product-list
- ☒ product-edit

Show user permission

Admin2 can't access [role-create|role-delete] & [product-create|product-delete]

Laravel Manage Role Manage User Manage Product admin2 ▾

Users Role

no see create button

No	Name	Action
1	Reader	Show Edit no see delete button
2	Admin	Show Edit
3	Manager	Show Edit

User-Role-Permission

Laravel Manage Role Manage User Manage Product admin2 ▾

Product Management

No	Name	Action
1	Fanta	Show Edit
2	Popie	Show Edit
3	Coca	Show Edit

User-Role-Permission