

Cours Laravel - Spatie (Roles & Permissions)

Ce support présente le fichier fourni concernant l'utilisation du package Spatie pour la gestion des rôles et permissions.

```
<?php

// https://spatie.be/docs/laravel-permission/v6/installation-laravel

/*
Outils liés à Spatie :
-----
Gestion des rôles■ Créer, modifier, supprimer et attribuer des rôles aux utilisateurs
Gestion des permissions■ Créer, modifier, supprimer et attribuer des permissions
Middlewares intégrés■ rôle, permission, role_or_permission pour sécuriser les routes
Helpers dans les vues■ Vérifier un rôle ou une permission directement dans Blade
Commandes Artisan■ Réinitialiser le cache, assigner des rôles rapidement
*/

/*
En ajoutant le trait HasRoles dans un model, on obtient un ensemble de methode pour les roles et pour
*/
use Spatie\Permission\Traits\HasRoles;

class User extends Authenticatable
{
    use HasRoles;
}

// Enregistrer le ou les middleware dans bootstrap/app.php
...
->withMiddleware(function (Middleware $middleware) {
    $middleware->alias([
        'role' => \Spatie\Permission\Middleware\RoleMiddleware::class,
        'permission' => \Spatie\Permission\Middleware\PermissionMiddleware::class,
        'role_or_permission' => \Spatie\Permission\Middleware\RoleOrPermissionMiddleware::class,
    ]);
})
...
// Roles
//-----
// Assigner un rôle à un utilisateur■
$user->assignRole('admin');
// Supprimer un rôle à un utilisateur■
$user->removeRole('admin');
// Supprime tous les rôles et en ajoute de nouveaux■
$user->syncRoles(['admin', 'client']);
// Vérifie si un utilisateur a un rôle■
$user->hasRole('admin');
// Récupère tous les rôles d'un utilisateur■
$user->getRoleNames();

// Permissions
//-----
// Donner une permission à un utilisateur■
$user->givePermissionTo('gerer_projets');
// Supprimer une permission à un utilisateur■
$user->revokePermissionTo('gerer_projets');
// Supprime toutes les permissions et en ajoute de nouvelles■
$user->syncPermissions(['voir_projets', 'gerer_taches']);
// Vérifie si un utilisateur a une permission■
$user->hasPermissionTo('gerer_projets');
// Récupère les permissions données via les rôles■
$user->getPermissionsViaRoles();

//-----
//-----

// Pour les routes :
//-----
// Vérifier un rôle
Route::get('/admin', function () {
    return 'Admin Dashboard';
});
```

```

})->middleware(['auth', 'role:admin']);

// Vérifier une permission
Route::get('/projects', function () {
    return 'Gérer les projets';
})->middleware(['auth', 'permission:gerer_projets']);

// Vérifier un rôle OU une permission
Route::get('/dashboard', function () {
    return 'Accessible si admin OU a la permission de voir le dashboard';
})->middleware(['auth', 'role_or_permission:admin,voir_dashboard']);

// pour les vues :
//-----
// Vérifier un rôle
@role('admin')
    <p>Ce contenu est visible uniquement pour les Admins.</p>
@endrole

// Vérifier plusieurs rôles
@hasrole('admin|client')
    <p>Ce contenu est visible pour les Admins et Clients.</p>
@endhasrole

// Vérifier une permission
@can('gerer_projets')
    <p>Tu peux gérer les projets.</p>
@endcan

// Vérifier un rôle OU une permission
@canany(['gerer_projets', 'voir_dashboard'])
    <p>Tu peux voir ceci si tu as au moins une des permissions.</p>
@endcanany

// Vérifier si un utilisateur N'A PAS un rôle
@unlessrole('client')
    <p>Ce contenu est caché aux Clients.</p>
@endunlessrole

// Commandes Artisan utiles
//-----
// Vider le cache des permissions et rôles
php artisan permission:cache-reset

// Créer un rôle via Artisan
php artisan permission:create-role admin

// Créer une permission via Artisan
php artisan permission:create-permission gerer_projets

```