

Atelier 11:

Securisation de l'accès aux routes par rôles

1- Installer Laravel UI :

```
houdaifa@houdaifaPC:~/everything/laravel/Ecommerce$ 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)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
As there is no 'unzip' nor '7z' command installed zip files are being unpacked using the PHP zip extension.
This may cause invalid reports of corrupted archives. Besides, any UNIX permissions (e.g. executable) defined in the archives will be lost.
Installing 'unzip' or '7z' (21.01+) may remediate them.
- Downloading laravel/ui (v4.6.1)
- Installing laravel/ui (v4.6.1): Extracting archive
```

2- Générer les vues et routes d'authentification :

```
houdaifa@houdaifaPC:~/everything/laravel/Ecommerce$ php artisan ui vue --auth

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

INFO Authentication scaffolding generated successfully.

INFO Vue scaffolding installed successfully.
```

3- Installer et compiler le front :

```
houdaifa@houdaifaPC:~/everything/laravel/Ecommerce$ npm install --save-dev vite@latest @vitejs/plugin-vue@latest
added 112 packages, and audited 113 packages in 12s

28 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
houdaifa@houdaifaPC:~/everything/laravel/Ecommerce$ npm run build

> build
> vite build
```

4- routes/web.php :

```
Route::resource('articles', RProduitController::class);
Auth::routes();
```

5- Ajouter une colonne role dans la table user

a. Créer la migration :

```
houdaifa@houdaifaPC:~/everything/laravel/Ecommerce$ php artisan make:migration migaddrolee

INFO Migration [2024_04_24_14534_migaddrolee.php] created successfully.
Focus folder in explorer (ctrl + click)
houdaifa@houdaifaPC:~/everything/laravel/Ecommerce$
```

b. Modifier la methode up :

```
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::table('users', function ($table) {
            $table->string('role')->nullable();
        });
    }
}
```

c. Executer la migration :

```
houdaifa@houdaifaPC:~/everything/laravel/Ecommerce$ php artisan migrate --path=database/migrations/2026_01_22_144534_migaddrole.php
INFO Running migrations.
2026_01_22_144534_migaddrolee ..... 101.08ms DONE
houdaifa@houdaifaPC:~/everything/laravel/Ecommerce$
```

6- Declarer les roles dans le model User :

```
const USER_ROLE = 'user';
const ADMIN_ROLE = 'admin';
protected $fillable = [
    'name',
    'email',
    'password',
    'role',
];
```

7- Role par default a l'inscription :

```
protected function create(array $data)
{
    return User::create([
        'name' => $data['name'],
        'email' => $data['email'],
        'password' => Hash::make($data['password']),
        'role' => User::USER_ROLE,
    ]);
}
```

8- Configuration des middlewares :

a. Admin middleware :

```
public function handle(Request $request, Closure $next)
{
    if (Auth::check() && Auth::user()->role === 'admin') {
        return $next($request);
    }

    return redirect('/')->with('error', 'Vous n\'avez pas accès à cette page.');
```

b. User middleware :

```
public function handle(Request $request, Closure $next)
{
    if (Auth::check() && Auth::user()->role === 'user') {
        return $next($request);
    }

    return redirect('/')->with('error', 'Vous n\'avez pas accès à cette page.');
```

9- Declarer les middlewares dans Bootstrap/app.php :

```
->withMiddleware(function (Middleware $middleware): void {
    $middleware->trustProxies(
        '*',
        Request::HEADER_X_FORWARDED_FOR |
        Request::HEADER_X_FORWARDED_HOST |
        Request::HEADER_X_FORWARDED_PORT |
        Request::HEADER_X_FORWARDED_PROTO
    );

    $middleware->alias([
        'admin' => AdminMiddleware::class,
        'user' => UserMiddleware::class,
    ]);
})
```

10- Dans routes/web.php :

```
2 // User Routes (Espace Client) - Protected by User Middleware
3 Route::middleware(['auth', 'user'])->group(function () {
4     Route::get('/espaceclient', [ProduitController::class, 'espaceclient'])->name('espaceclient');
5 });
6
7 // Admin Routes (Product Management) - Protected by Admin Middleware
8 Route::middleware(['auth', 'admin'])->group(function () {
9     Route::get('/articles', [RProduitController::class, 'index'])->name('articles.index');
10    Route::get('/articles/create', [RProduitController::class, 'create'])->name('articles.create');
11    Route::post('/articles', [RProduitController::class, 'store'])->name('articles.store');
12    Route::get('/articles/{id}/edit', [RProduitController::class, 'edit'])->name('articles.edit');
13    Route::put('/articles/{id}', [RProduitController::class, 'update'])->name('articles.update');
14    Route::delete('/articles/{id}', [RProduitController::class, 'destroy'])->name('articles.destroy');
15 });
```