

# Controllers

## # في البداية :

إحنا لما نتكلم عن Laravel ك Framework ، بيبقى التركيز دائماً على فصل الكود وتنظيمه. هنا بييجي دور ال Controller، اللي هو عقل المشروع اللي بيتعامل مع الطلبات ويقرر يرد على المستخدم إزاي .

## # فكرة الأساسية :

- المستخدم (User) بيعت طلب (Request).
- ال Controller بيستقبله، ينفذ شوية تعليمات (Logic).
- في النهاية يجهز استجابة (Response)، زي عرض صفحة أو إرسال بيانات.

## # بننشئ Controller ازاي :

- Laravel بيقدّم لنا أداة قوية اسمها **Artisan CLI** عشان نعمل حاجات بسرعة. واحدة من الحاجات دي هي إنشاء **Controller**.

```
1 // To create a new controller named "MyController".
2 php artisan make:controller MyController
3
4 /* The file path of the created controller:
5    "app/Http/Controllers/MyController.php". */
6
```

- بعدين ، لو فتحنا الملف ، هنلاقي إن ال Controller مجرد كلاس جواه دوال (Methods) ، والدوال دي اللي بنكتب فيها ال Logic بتاعنا .

## # ربط ال Controller مع Routes :

ال Routes هي العناوين اللي المستخدم بيزورها ، واللي بتوصلنا بال Controller. يعني لما حد يفتح صفحة ، Laravel بيشفوف ال Route ويربطه بدالة معينة في ال Controller .

مثال عملي :  
في ملف `routes/web.php`

```
1 use App\Http\Controllers\MyController;
2
3 Route::get('/', [MyController::class, 'index']);
4 Route::get('/item/{id}', [MyController::class, 'show']);
5
```

← `Route::get` : لما المستخدم يفتح صفحة باستخدام متصفح (GET Request) .  
← `MyController::class` : ده بيحدد ال Controller .  
← `'index'` أو `'show'` : اسم الدالة اللي هتتنفذ .

## # كتابة أول Controller Logic :

• هنفتح ال Controller ونكتب الدوال بتاعته :

```
1 namespace App\Http\Controllers;
2
3 class MyController extends Controller
4 {
5     public function index()
6     {
7         return view('welcome');
8     }
9
10    public function show($id)
11    {
12        return "Displaying item with ID: " . $id;
13    }
14 }
```



← index : لما يفتح المستخدم الصفحة الرئيسية، هيرجع له ملف View اسمه welcome .

← show(\$id) : لما يدخل على رابط زي /item/٤ ، هنعرض له النص " Displaying item with ID : ٤ "

## # Resource Controllers :

- لو عندك عمليات أساسية زي (إضافة ، عرض ، تعديل ، حذف) ، Laravel بيوفر طريقة سهلة جداً اسمها Resource Controller.

```
1 // To create a new resource controller named "ProductController".
2 php artisan make:controller ProductController --resource
3
```

ده بينشئ كلاس جواه دوال جاهزة زي :

- index : عرض كل العناصر.
- create : عرض فورم الإضافة.
- store : حفظ البيانات.
- show : عرض عنصر معين.
- edit : عرض فورم التعديل.
- update : تحديث البيانات.
- destroy : حذف عنصر.

# ربط ال Resource Controller بال Routes :

```
1 Route::resource('products', ProductController::class);
2
```

Laravel هيعمل كل ال Routes المطلوبة بشكل تلقائي:

- GET /products → يعرض كل المنتجات.
- POST /products → يضيف منتج جديد.
- PUT /products/{id} → يعدّل المنتج.
- DELETE /products/{id} → يحذف المنتج.

## # استخدام Middleware مع Controllers:

- لو عايز تضيف شروط معينة زي حماية الصفحات، Middleware هو الحل. تقدر تضيفه على مستوى الدوال أو على مستوى الـ Controller كله.

- مثال :  
في الـ Controller :

```
1 public function __construct()  
2 {  
3     $this->middleware('auth')->only(['edit', 'update']);  
4 }
```

هنا ← Middleware (auth) بيمنع أي حد مش مسجل دخول من الوصول لدوال معينة زي edit و update .