

# Développement WEB PHP - Séance 8

**BUT Informatique parcours DACS**



# Table of contents:

-  | Introduction à Laravel
  - Laravel : Un Framework PHP Moderne
  - Pourquoi utiliser Laravel ?
  - Installation de Laravel
    - Pré-requis
    - Installer Laravel via Composer
    - Démarrer le serveur de développement
  - Structure d'un Projet Laravel
  - Les Bases de Laravel
    - Les Routes
    - Les Contrôleurs
    - Les Vues (Blade)
  - Interactions avec la Base de Données
    - Configuration
    - Migrations
    - Eloquent ORM
  - Gestion des Commandes avec Artisan
    - Commandes courantes
  - Points Forts à Retenir
  - Petit exercice
-  | TD 7 : Eco-système de Laravel
  - Objectif
  - Consigne
    - Éléments à présenter
    - Livrable attendu
    - Évaluation

# | Introduction à Laravel

## Laravel : Un Framework PHP Moderne

Laravel est un framework PHP open-source conçu pour le développement rapide et structuré d'applications web. Connu pour sa syntaxe élégante et ses outils puissants, il est idéal pour créer des applications robustes tout en simplifiant les tâches courantes du développement.

# Pourquoi utiliser Laravel ?

Laravel se distingue par plusieurs caractéristiques qui en font un choix populaire parmi les développeurs :

- **Facilité d'utilisation** : Une syntaxe simple et intuitive.
- **MVC (Modèle-Vue-Contrôleur)** : Une architecture claire pour structurer les applications.
- **Fonctionnalités intégrées** :
  - Système de routage avancé.
  - Gestionnaire de base de données avec Eloquent ORM.
  - Outils pour l'authentification et les sessions.
  - Gestion des tâches planifiées et des files d'attente.
- **Communauté et écosystème** : Une documentation exhaustive, des tutoriels, et des extensions comme **Laravel Nova** ou **Laravel Breeze**.

[Lien vers la documentation de Laravel](#)

# Installation de Laravel

Laravel nécessite PHP 8.1 ou plus, Composer, et un serveur web comme Apache ou Nginx. Voici comment démarrer :

## Pré-requis

- Installer le gestionnaire de paquets PHP **Composer** (c'est déjà fait)

## Installer Laravel via Composer

Utilisez la commande suivante pour créer une nouvelle application Laravel :

```
composer create-project laravel/laravel nom_du_projet
```

## Démarrer le serveur de développement

Entrez dans le répertoire du projet et lancez le serveur intégré de Laravel :

```
cd nom_du_projet  
php artisan serve
```

Votre application est accessible à l'adresse : `http://localhost:8000`.

# Structure d'un Projet Laravel

Laravel organise le code selon une architecture claire. Voici les répertoires principaux :

- **app/** : Contient le cœur de votre application, y compris les contrôleurs, modèles et services.
- **routes/** : Définit les routes de votre application.
- **resources/** : Contient les fichiers de vue (Blade), CSS, JS, et autres ressources.
- **database/** : Contient les migrations et les seeds pour la gestion des bases de données.
- **public/** : Répertoire racine accessible au navigateur (contient `index.php`).

# Les Bases de Laravel

## Les Routes

Les routes définissent les points d'entrée de votre application. Elles sont définies dans le fichier `routes/web.php`.

**Exemple :**

```
use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return view('welcome'); // Charge la vue 'welcome.blade.php'
});

Route::get('/hello', function () {
    return "Hello, Laravel !";
});
```

## Les Contrôleurs

Les contrôleurs regroupent la logique métier. Créez un contrôleur avec la commande Artisan :

```
php artisan make:controller MonContrôleur
```

**Exemple :**

```
namespace App\Http\Controllers;

use Illuminate\Http\Request;

class MonContrôleur extends Controller
{
    public function accueil() {
        return view('accueil'); // Charge la vue 'accueil.blade.php'
    }
}
```

Ajoutez une route pointant vers le contrôleur :

```
Route::get('/accueil', [MonContrôleur::class, 'accueil']);
```

## Les Vues (Blade)

Blade est le moteur de templates de Laravel. Les fichiers Blade se trouvent dans `resources/views` et ont l'extension `.blade.php`.

**Exemple :** Fichier `resources/views/accueil.blade.php` :

```
<!DOCTYPE html>
<html>
<head>
    <title>Accueil</title>
</head>
<body>
    <h1>Bienvenue sur Laravel</h1>
</body>
</html>
```



# Interactions avec la Base de Données

## Configuration

Configurez votre base de données dans le fichier `.env` :

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=nom_de_la_base
DB_USERNAME=utilisateur
DB_PASSWORD=mot_de_passe
```

## Migrations

Les migrations permettent de versionner la structure de votre base de données. Créez une migration avec :

```
php artisan make:migration create_utilisateurs_table
```

Modifiez le fichier généré dans `database/migrations` :

```
Schema::create('utilisateurs', function (Blueprint $table) {
    $table->id();
    $table->string('nom');
    $table->string('email')->unique();
    $table->timestamps();
});
```

Appliquez les migrations avec :

```
php artisan migrate
```

## Eloquent ORM

Eloquent simplifie la manipulation des données en utilisant des modèles. Créez un modèle pour votre table :

```
php artisan make:model Utilisateur
```

Ajoutez des données :

```
use App\Models\Utilisateur;

Utilisateur::create([
    'nom' => 'Alice',
    'email' => 'alice@example.com'
]);
```

Récupérez des données :

```
$utilisateurs = Utilisateur::all();
```

# Gestion des Commandes avec Artisan

Laravel inclut **Artisan**, une interface en ligne de commande pour automatiser les tâches courantes.

## Commandes courantes

- `php artisan list` : Liste toutes les commandes disponibles.
- `php artisan migrate` : Applique les migrations.
- `php artisan make:controller NomContrôleur` : Crée un nouveau contrôleur.
- `php artisan route:list` : Liste toutes les routes définies.

# Points Forts à Retenir

- Laravel simplifie les tâches complexes grâce à ses outils intégrés (migrations, routage, validation).
- Son écosystème riche (comme **Laravel Mix**, **Laravel Telescope**) étend ses capacités.
- Il favorise les bonnes pratiques de développement grâce à son architecture propre (MVC).

Laravel est un excellent choix pour les développeurs PHP souhaitant gagner en productivité tout en produisant du code maintenable et évolutif. Avec sa syntaxe élégante et sa large communauté, il vous permet de construire des applications modernes en un temps record.

## Petit exercice

- Créer un endpoint/une route `/nom-etudiant` (GET).
- Le endpoint renverra vers une fonction dans un contrôleur (StudentController).
- La fonction aura une constante de nom d'étudiant (peu importe le nom) et renverra une page HTML avec ce même nom affiché (via Blade).
- Faire valider par le professeur.



# | TD 7 : Eco-système de Laravel

## Objectif

Explorer l'écosystème Laravel pour comprendre les outils et services qui l'enrichissent. Vous devrez analyser un élément spécifique et le présenter de manière critique.

# Consigne

1. **Formez des groupes de 2 à 3 étudiants.**
2. Chaque groupe se voit attribuer un élément de l'écosystème Laravel (voir liste ci-dessous).
3. Effectuez des recherches sur cet élément (30 minutes) 4Préparez un mini-exposé de 5 minutes sur cet élément.

Votre présentation devra inclure les critères suivants :

- **Description** : À quoi sert cet outil/service ?
- **Fonctionnalités clés** : Quelles sont ses principales caractéristiques ?
- **Avantages** : Quels bénéfices apporte-t-il pour un développeur Laravel ?
- **Inconvénients/Limitations** : Y a-t-il des points faibles ? (Prix, quantité de mises à jour...)
- **Jugement** : Recommanderiez-vous son usage ? Pourquoi ?

## REMARQUE

N'oubliez pas l'esprit critique que nous avons développé lors de la séance précédente !

## Éléments à présenter

1. Laravel **Vapor, Forge**
2. Laravel **Breeze, Jetstream**
3. Laravel **Nova**
4. Laravel **Sanctum**
5. Laravel **Blade, Livewire et Inertia**
6. Laravel **Cashier, Spark**
7. Laravel **Pennant**

---

## Livrable attendu

Un support visuel (slides ou tableau blanc) pour structurer votre présentation.

## Évaluation

- Respect des critères demandés (description, avantages/inconvénients, jugement).
- Clarté de l'exposé et capacité à répondre aux questions des autres groupes.

- Qualité de l'analyse critique.
- 

Cet exercice permettra d'approfondir vos connaissances sur Laravel tout en développant vos compétences en recherche, synthèse et présentation.