



**IMT Lille Douai**  
École Mines-Télécom  
IMT-Université de Lille

# LARAVEL FRAMEWORK PHP

# SOMMAIRE

## 1. PRÉSENTATION

- 1.1 Le cours
- 1.2 La notation

## 2. LARAVEL

- 2.1 Framework PHP
- 2.2 MVC
- 2.3 Installation
- 2.4 Premiers pas

## 3. LES DONNÉES

- 3.1 Les modèles
- 3.2 Les migrations et population

## 4. LOREM IPSUM

- 4.1 Formulaire
- 4.2 Tests
- 4.3 Authentification



### 1.1 Cours / TD / TP

- ▶ 3 Séances de 4h
- ▶ Alternance entre cours / TP
- ▶ En semi-autonomie
- ▶ BBB – Discord – Trello

### 1.2 Notation

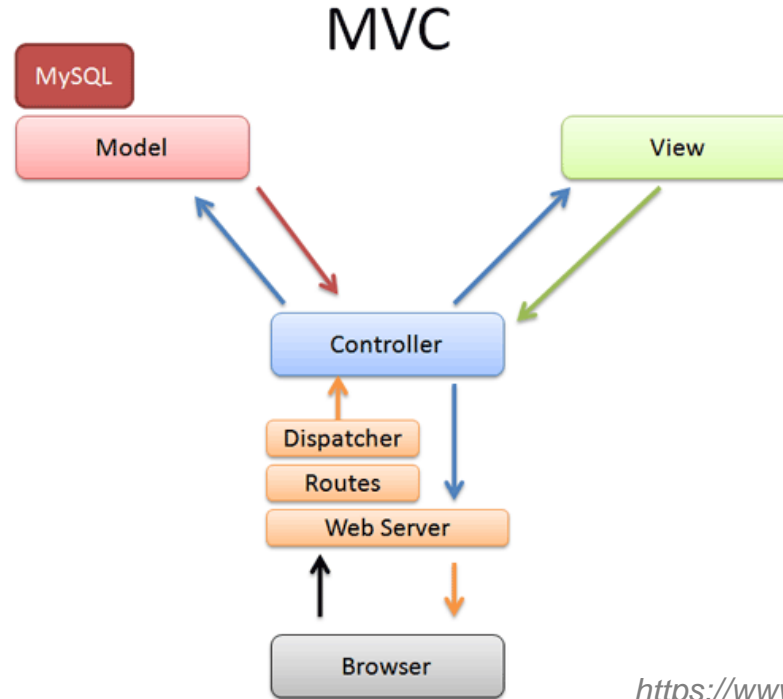
- ▶ Une note à la fin des séances
- ▶ Partage d'écran avec démo et explications de code de 10 min puis 5 min de questions
- ▶ Attentes
  - Code (pas de duplication, indenté, commenté, versionné, refactorisé...)
  - Tests

### Framework

- ▶ Ensemble d'outils et de librairies
- ▶ Existe pour toute sorte de langage (PHP, javascript, ...)
- ▶ Concurrent de Laravel : Symfony
- ▶ Ne remplace pas la connaissance d'un langage

### Constitution

- ▶ Système de routage
- ▶ Moteur de template (Blade)
- ▶ Object-Relational Mapping – ORM (Eloquent)
- ▶ Système d'authentification et de gestion de sessions pour les connexions



<https://www.awesomeinc.org/tutorials/rails-blog/>

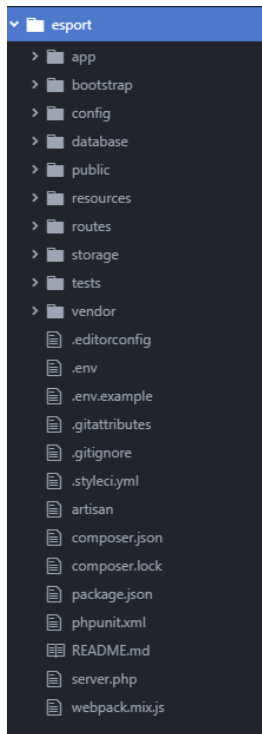
### Composer

- ▶ Gestionnaire de dépendances
- ▶ Configuration par un fichier JSON
- ▶ Installer : <https://getcomposer.org/download/>
- ▶ Vérification (terminal)

*composer --version*

### Laravel

- ▶ Se placer dans le dossier à la racine de votre serveur (www)
- ▶ Création du projet esport avec la dernière version stable de Laravel (terminal) :  
*composer create-project --prefer-dist laravel/laravel esport*
- ▶ Vérification (navigateur) :  
<http://localhost/esport/public>
- ▶ Attention aux droits d'écriture et lecture



### Principaux dossiers et fichiers

- ▶ app : cœur de l'application (controller, model, helper, ...)
- ▶ config : les fichiers de configuration
- ▶ database : migrations, seeders
- ▶ public : images. Seul dossier accessible depuis le serveur
- ▶ resources : vues, css / scss et js
- ▶ routes : routes de l'application dans web.php
- ▶ storage : session, log
- ▶ .env : environnement de l'application
- ▶ composer.json : dépendances de l'application



## Route

- ▶ Fichier routes/web.php
- ▶ Remplacer l'appel de la vue par « Hello world »
- ▶ Options

## Vue

- ▶ Blade : moteur de template
- ▶ maVue.blade.php
- ▶ `<?php echo $numero ?>` → `{{ $numero }}`
- ▶ `@if`, `@elseif`, `@endif`, ... (<https://walkerspider.com/cours/laravel/blade>)
- ▶ `@extend` : extension d'une vue par une autre
- ▶ `@section` : permet de définir du contenu
- ▶ `@yield` : section que pourront utiliser les vues enfants

## Contrôleur

- ▶ Créer un contrôleur :  
*php artisan make:controller MonContrôleur*
- ▶ Appeler la vue depuis le contrôleur
- ▶ Relier une route à un contrôleur
- ▶ Récupérer des paramètres depuis la route et les passer à la vue
- ▶ Afficher ces paramètres

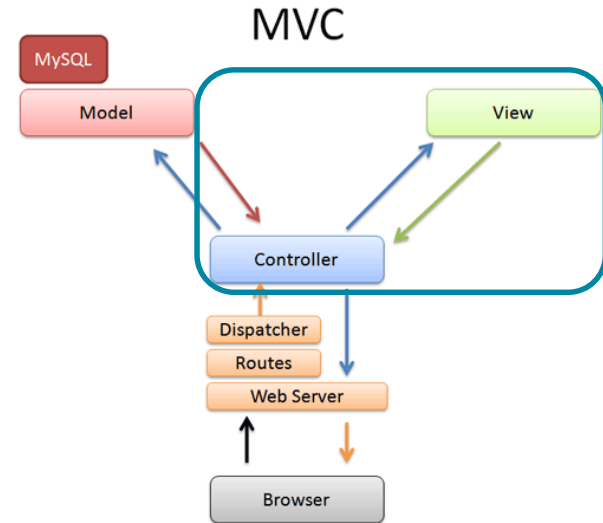
## E-sport

Gestion de tournois de jeux en ligne

- ▶ MLD dispo sur MLS
- ▶ Connexion (identification des rôles admin, joueur)
- ▶ CRUD de tournois nationaux/internationaux, de jeux (avec catégorie), d'équipes
- ▶ Formulaire d'inscription
- ▶ Participation des équipes à un tournoi
- ▶ Gestion du classement

## Route

- ▶ Créez une vue template
  - un titre,
  - un bandeau,
  - un menu (pour rôle admin)
- ▶ Créez une vue enfant avec un texte d'accueil
- ▶ Créez le contrôleur qui va appeler cette vue
- ▶ Créez la route
- ▶ Vérifiez le tout



- ▶ Le modèle est chargé de gérer les données
- ▶ Les migrations permettent de créer et mettre à jour la structure des tables
- ▶ Les seeders permettent de remplir les tables (ajout de données)
- ▶ Les factories permettent de créer des enregistrements en quantité et d'établir facilement diverses relations entre les tables

### ► Création BD

- Création de la base de données (phpMyAdmin)
- Nom : esport
- Type : utf8\_general\_ci

### ► Configuration

- database.php
- .env (DB\_DATABASE)

### Modèle

- ▶ Création

*php artisan make:model Game*

- ▶ Configuration

## Migration

### ► Création

*php artisan make:migration game\_table*

ou

*php artisan make:migration game\_table --create=users*

(Avec l'option create qui crée la table game)

### ► Configuration

### ► Exécution

*php artisan migrate*

### ► Clé étrangère



## Population (seeder)

- ▶ Création

*php artisan make:seeder GameTableSeeder*

- ▶ Configuration

- ▶ Exécution

*php artisan db:seed*

- ▶ Effacer les enregistrements pour repartir d'une base vide

*php artisan migrate:fresh*

## Factories

- ▶ Création

*php artisan make:factory GameFactory --model=Game*

- ▶ Configuration

- ▶ Exécution

*php artisan migrate:fresh --seed*

- ▶ Voir pour les relations 1-n et n-n

## Formulaire

- ▶ Créer un formulaire de saisie de tournoi qui étend votre template
  - Ajouter le token CSRF (voir ce que c'est)
  - Ajouter un menu déroulant avec la liste des jeux
  - Ajouter des validations sur le format du nom du tournoi, la date de tournoi, le choix d'un jeu
  - Insérer le nouveau jeu dans la BD

## Les tests

### ► Tests manuels

- Installer Barryvdh - Laravel Debugbar

### ► Tests unitaires

- PHP Unit
- Aide

*php vendor\phpunit\phpunit\phpunit -h*

- Lancer les tests

*php artisan test*

## Authentification

► Laravel propose un système d'authentification

► Suivre le cours :

<https://laravel.sillo.org/cours-laravel-8-la-securite-lauthentification/>

- ▶ <https://walkerspider.com/cours/laravel>
- ▶ <https://laravel.sillo.org/laravel-8/>
- ▶ <https://www.grafikart.fr/tutoriels/laravel>