

laravel

DÉROULEMENT de la MASTERCLASS

1. Mise en place de l'environnement
2. Présentation de laravel - points forts
3. Effectuer les migrations
4. Mise en place des routes
5. Mise en place du modèle et contrôleur
6. La pagination
7. Les vues (Blades)
8. Générer les fixtures

Mise en place du projet Laravel

-- installation --

```
git clone https://github.com/julienchenel/masterclass\_laravel.git myproject
```

```
&& cd myproject
```

```
&& composer update
```

```
&& cp .env.example .env
```

```
&& php artisan key:generate
```

```
&& php artisan serve
```

1. Mise en place du projet Laravel

-- création de la base de donnée --

Ouvrez un nouveau terminal et tapez

mysql -u root -p // remplacer *root* par votre login mysql

Entrer votre mot de passe puis

CREATE DATABASE *laravel_bdd* // remplacer *laravel_bdd* par le nom de votre bdd

1. Mise en place du projet Laravel

-- définition des variables d'environnement --

```
sudo nano .env
```

1. Mise en place du projet Laravel

-- définition des variables d'environnement --

APP_NAME=Laravel

APP_KEY=base64:0x3ewQGFw6HR/Wf+tPQTBF0tOd6bvOQXhjt0G

DB_CONNECTION=pgsql // mysql si vous êtes sur mysql

DB_HOST=127.0.0.1

DB_PORT=5432 // 3306 pour mysql // 27017 pour mongodb

DB_DATABASE=laravel_bdd // le nom de votre base de données

DB_USERNAME=admin // votre login

DB_PASSWORD=admin // et votre mot de passe

2. Présentation de Laravel

-- Points forts de Laravel --

- bien documenté...

<https://laravel.com/docs/5.6> et <https://laracasts.com/>

- bien commenté...

```
<!-- config/app.php -->

<?php

return [

    /*
    |-----
    | Application Name
    |-----
    |
    | This value is the name of your application. This value is used when the
    | framework needs to place the application's name in a notification or
    | any other location as required by the application or its packages.
    |
    */

    'name' => env('key: 'APP_NAME', default: 'Laravel'),
```

2. Présentation de Laravel

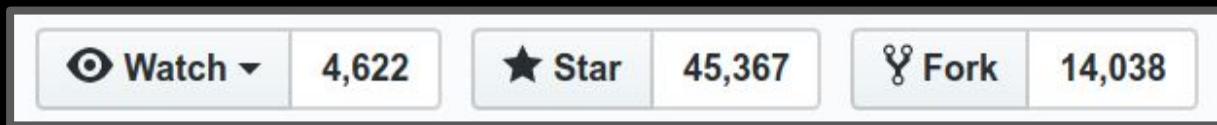
-- Points forts de Laravel --

- Mise à jour régulière (version majeure tous les 6 mois)

2. Présentation de Laravel

-- Points forts de Laravel --

- grande communauté → <https://github.com/laravel/laravel>



- courbe d'apprentissage rapide

Nous avons notre environnement (projet laravel + bdd + ...).

Il nous faut maintenant un schéma de base de donnée...

3. Effectuer les migrations

php artisan migrate

4. Mise en place d'une route

➔ Définition des URLs

```
Route::get('ma_route', 'ExempleController@index');
```

A vous de générer la route vers la vue “formateur” !

5. Création du modèle

-- [avec le contrôleur associé] --

➡ Réceptionner une requête et de définir la réponse appropriée.

php artisan make:model *exemples* -cr

A vous de générer le modèle et le contrôleur !

N.B :

Qu'il est bon de faire un **git status** après chaque commande pour voir ce qui se passe, quels fichiers ont été créés

le modèle

```
class exemples extends Model
```

```
{
```

```
    protected $fillable = ['nom', 'prenom', 'mail', 'adresse', 'telephone'];
```

```
    protected $table='exemples';
```

```
    protected $primaryKey='id';
```

```
}
```

Le contrôleur

```
class ExempleController extends Controller
{
    public function index()
    {
        $data = ma_table::all();
        return view('ma_vue')->withData($data);
    }
}
```

6. Gérer la pagination

```
class ExempleController extends Controller
{
    public function index()
    {
        $data = DB::table('formateurs')->paginate(15);
        return view('formateur')->withData($data);
    }
}
```

Ajoutez ceci dans la vue : {{ \$data->links() }}

7. Les vues “Blades”

Les **blades** sont construit de façon à rendre le code plus facile à écrire.

Le blade par défaut commencera comme n'importe quel `index.html`. Si je besoin de changer une donnée dans le code, je n'ai pas besoin de changer tous les blades, je n'ai qu'à changer le blade par défaut.

default.blade.php

```
<!DOCTYPE html>
<html lang="fr">
<head>
...
<title>@yield('title')</title>
...
<link href="..." type="text/css" rel="stylesheet" />
...
@yield('custom_css')
...
</head>
<body>
...
@yield('content')
...
<script src="..."></script>
...
@yield('scripts')
...
</body>
</html>
```

page1.blade.php

@section('title')

Titre de la page 1

@endsection

@section('content')

<div></div>

@endsection

page2.blade.php

@section('title')

Titre de la page 2

@endsection

@section('custom_css')

<link href=.../>

@endsection

@section('content')

<div></div>

@endsection

8. Générer les fixtures

-- création des classes --

basée sur des classes pour **peupler votre base de données** avec des **données de test**.

```
▸ app
▸ bootstrap
▸ config
▸ database
  ▸ factories
  ▸ migrations
    2014_10_12_000000_create_users_table.php
    2014_10_12_100000_create_password_resets_table.php
    2018_08_25_000002_create_formateurs_table.php
    2018_08_25_000003_create_apprenants_table.php
  ▸ seeds
    ApprenantsTableSeed.php
    DatabaseSeeder.php
    FormateursTableSeeder.php
```

**Créer vos classes
de fixtures ici**

```
<?php
```

```
use Illuminate\Database\Seeder;
```

```
use App\Formateurs;
```

```
class FormateursTableSeeder extends Seeder {
```

```
    public function run()
```

```
{
```

```
    $faker = Faker\Factory::create('fr_FR');
```

```
    for ($i = 0; $i < 100; $i++) {
```

```
        $formateur = new Formateurs;
```

```
        $formateur->nom = $faker->lastName;
```

```
        /*
```

```
        * ...
```

```
        * lastName est une méthode qui va générer aléatoirement des noms de famille.
```

```
        * Continuer à générer les données pour les autres champs. A vous de jouer !
```

```
        * Les autres méthodes sont disponibles ici : https://github.com/fzaninotto/Faker
```

```
        */
```

```
        $formateur->save();
```

```
<?php
use Illuminate\Database\Seeder;
use App\Apprenants;
use App\Formateurs;

class ApprenantsTableSeeder extends Seeder
{
    public function run()
    {
        $faker = Faker\Factory::create('fr_FR');
        for ($i = 0; $i < 100; $i++) {
            $apprenant = new Apprenants;
            $apprenant->nom = $faker->lastName;
            $apprenant->prenom = $faker->firstName;
            $apprenant->mail = $faker->unique()->email;
            $apprenant->adresse = $faker->streetAddress;
            $apprenant->telephone = $faker->phoneNumber;
            $apprenant->formateurs_id = $faker->numberBetween(1, 100);
            $apprenant->save();
        }
    }
}
```

```
<?php
```

```
use Illuminate\Database\Seeder;
```

```
use App\Apprenants;
```

```
use App\Formateurs;
```

```
class ApprenantsTableSeeder extends Seeder
```

```
{
```

```
    public function run()
```

```
    {
```

```
        $faker = Faker\Factory::create('fr_FR');
```

```
        for ($i = 0; $i < 100; $i++) {
```

```
            $apprenant = new Apprenants;
```

```
            $apprenant->nom = $faker->lastName;
```

```
            $apprenant->prenom = $faker->firstName;
```

```
            $apprenant->mail = $faker->unique()->email;
```

```
            $apprenant->adresse = $faker->streetAddress;
```

```
            $apprenant->telephone = $faker->phoneNumber;
```

```
            $apprenant->formateurs_id = $faker->numberBetween(10, 20);
```

```
            $apprenant->save();
```

```
        };
```

```
    }
```

```
}
```

Une fois la classe des fixtures terminée..

Exécutez d'abord :

```
composer dump-autoload
```

 Ensuite:

Puis :

```
php artisan db:seed --class=FormateursTableSeeder
```

Terminé

Vous pouvez reprendre une activité normale.

Mémoire de stage, Opcast, etc :)