Laravel

Diapos : <https://docs.google.com/presentation/d/11aq1-r0ehzyyNpR3ja-ChPwPHNE6iJ-K9woT7M5lIi4/edit>

Créer le projet depuis le dépot github

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

Créer la base de données mysql

CREATE DATABASE laravel\_bdd;

Ouvrir le fichier .env et modifier les paramètres de connexion à la BDD

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=homestead

DB\_USERNAME=homestead

DB\_PASSWORD=secret

Aller dans le dossier suivant:

/database/migrations/

C’est ici que Laravel créé les scripts de création des tables

Aller dans le fichier

2018\_08\_25\_000002\_create\_formateurs\_table.php

Blueprint est une classe qui permet de créer un nouveau schéma

Spécificités Laravel sur le nommage des tables et des clés primaires:

Tous les noms des tables doivent avoir le suffixe “s”

Toutes les clés primaires sont préfixés avec le terme “id”

En ligne de comande : taper “php artisan” pour afficher toutes les commandes propres à artisan

Migrer les tables dans la base de données mysql

php artisan migrate

Msg d’erreur => Modifier le fichier “2018\_08\_25\_000003\_create\_apprenants\_table.php” et ajoute run deuxième schéma propre à la création de sclés étrangères

“php artisan:cach” clear pour vider le cache

Réflex en cas d’erreur, arrêter le server et redémarrer avec “php artisan serve”

Créer un nouveau controller pour le modèle “Formateurs” : “php artisan make:model Formateurs -cr” (-cr pour créer un controlleur)

=> cette commande créé automatiquement les fichiers dans l’arborescence de fichiers

Le controlleur dans app/Http/Controllers/FormateursController

Le modèle dans app/Formateurs.php

Définir les propriétés “Protected” de la la classe pour le modèle

Pagination : méthode qui permet de paginer les résultats dans le résultat retourné par la requête.

Appeller des variables php dans une vue avec Laravel : {{$data->links()}}

Les vues avec “Blade” : ce sont des fichiers qui portent l’extension “.blade.php”

Elles se trouvent dans le dossier views.

layouts/app.blade.php correspond à la vue par défaut à afficher

Les autres fichiers correspondent aux différentes vues

Aller dans le dossier “routes” et ajouter la ligne suivante au fichier “web.php” :

*Route*::*get*('/formateurs', 'FormateursController@index')->*name*('formateurs');

Modifier le fichier “formateurs.php” :

*class* formateurs extends *Model*

{

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

protected $table='formateurs';*//fait référence à la table de la base de données*

protected $primarykey='id';

}

Modifier le fichier “FormateursController.php” :

public *function* *index*()

{

$data = *formateurs*::*all*();

return *view*('formateur')->*withData*($data); *// formateur fait référence au fichier formateur.blade.php qui est la vue*

*// echo "<p>test</p>";*

}

A chaque fois que le schéma est modifié, il faut faire la migration des modifications dans la base de données:

php artisan migrate:refresh

/!\ si un message d’erreur s’affiche, relancer la commande

Générer des fixtures:

Dans /Seeds créer deux fichiers “ApprenantsTableSeeder.php” et “FormateursTableSeeder.php” et y copier le code fourni

<?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 < 1000; $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();

}

}

}

?>

Pour éxecuter le script:

Inclure la nouvelle classe créée , la charger dans l’application

composer dump-autoload

Lancer la population des données dans la base de données

php artisan db:seed --class=FormateursTableSeeder