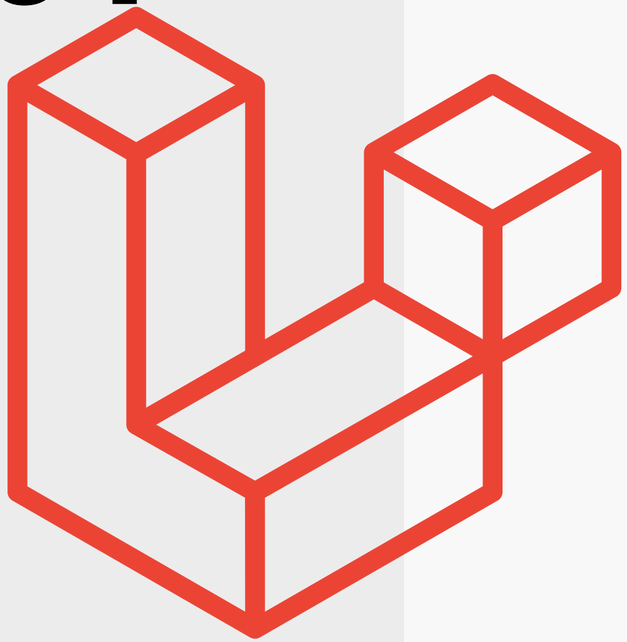
/01



Laravel

Pour un back solide

Qui c’est le prof ?

Ancien MMI / Licence Pro à l’IUT de Dijon

Développeur full-stack / devOps depuis 7 ans

PHP / Laravel / JavaScript / Vue.js, ...

/03

Laravel c’est quoi ?

Framework PHP open source

authentification

base de données

formulaires

notifications

...

/04

Un peu de réseaux sociaux

Taylor Otwell

Créateur de Laravel
@taylorotwell

Nuno Maduro

Equipe Laravel
@enunomaduro

Freek Van der Herten

Développeur chez Spatie
@freekmurze

Avant de commencer

Git / Github
Composer
Docker / Docker compose

/06

GIT GITHUB

Main

Version principale du code

Branche

Copie de code pour ajouter une feature ou corriger un bug

Commit

Modification d’un ou plusieurs fichiers

Push

Envoie du code sur le serveur Git

Merge

Fusion de branche

Github

Plateforme indépendante

=> Créer un compte Github

=> Créer un repository

Comprendre “projet” ici

Composer

Dépendance

Ensemble de fonctions, objets, ... relatifs à un domaine particulier. Ex : stockage de fichier

CLI

Outil en ligne de commande (Command Line)

Simplicité

Plus besoin de copier manuellement des tas de fichiers

Docker
Docker Compose

Image

Description des éléments nécessaires pour faire fonctionner votre application

Conteneur

Processus système intégrant un OS et tout ce qu'il faut pour faire tourner votre code

Linux ❤

Les images reposent généralement sur Linux. Et ce sera le cas dans ce cours

Docker Compose

Orchestration de différent conteneurs pour mettre en place un environnement complet (serveur web, base de données, ...)

/09

C’est parti 

1) Cloner le projet

bit.ly/3LaFTwB

2) Suivre le README.md

Découverte

Pour chaque dossier son usage.

App

Logique métier. Contrôleurs, modèles, ...

Config

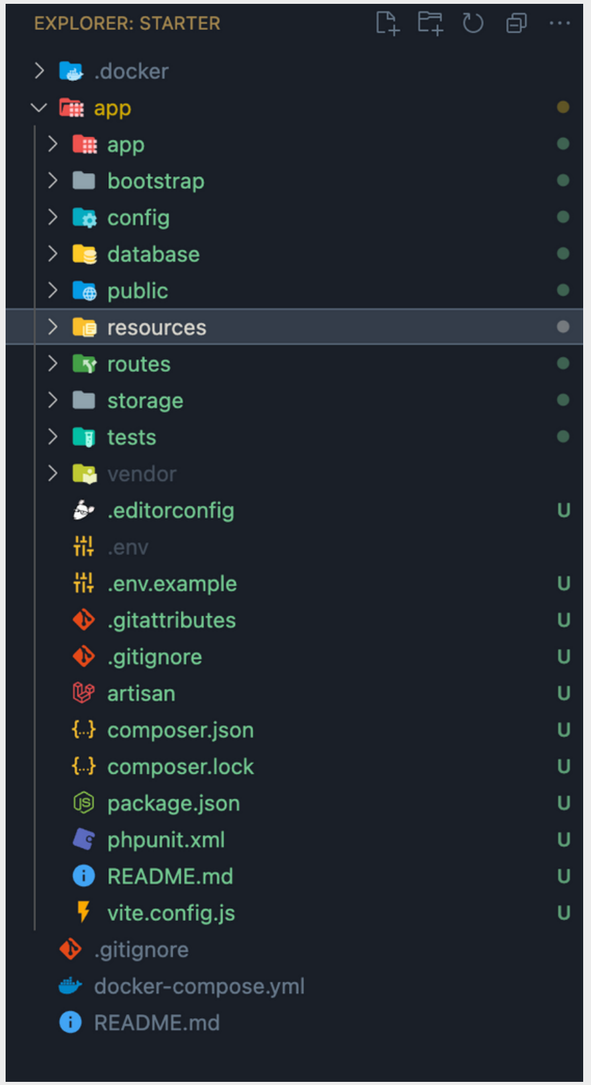
Paramètre de l'application en fonction du context

Database

Mise à jour des tables

Ressources

JS, CSS, images, ...



/11

ENVIRONNEMENTS

Environnement

.env

Définit

Configurations

config/app.php, ...

Routing

laravel.com/docs/10.x/routing

Définition

Une route = une URL

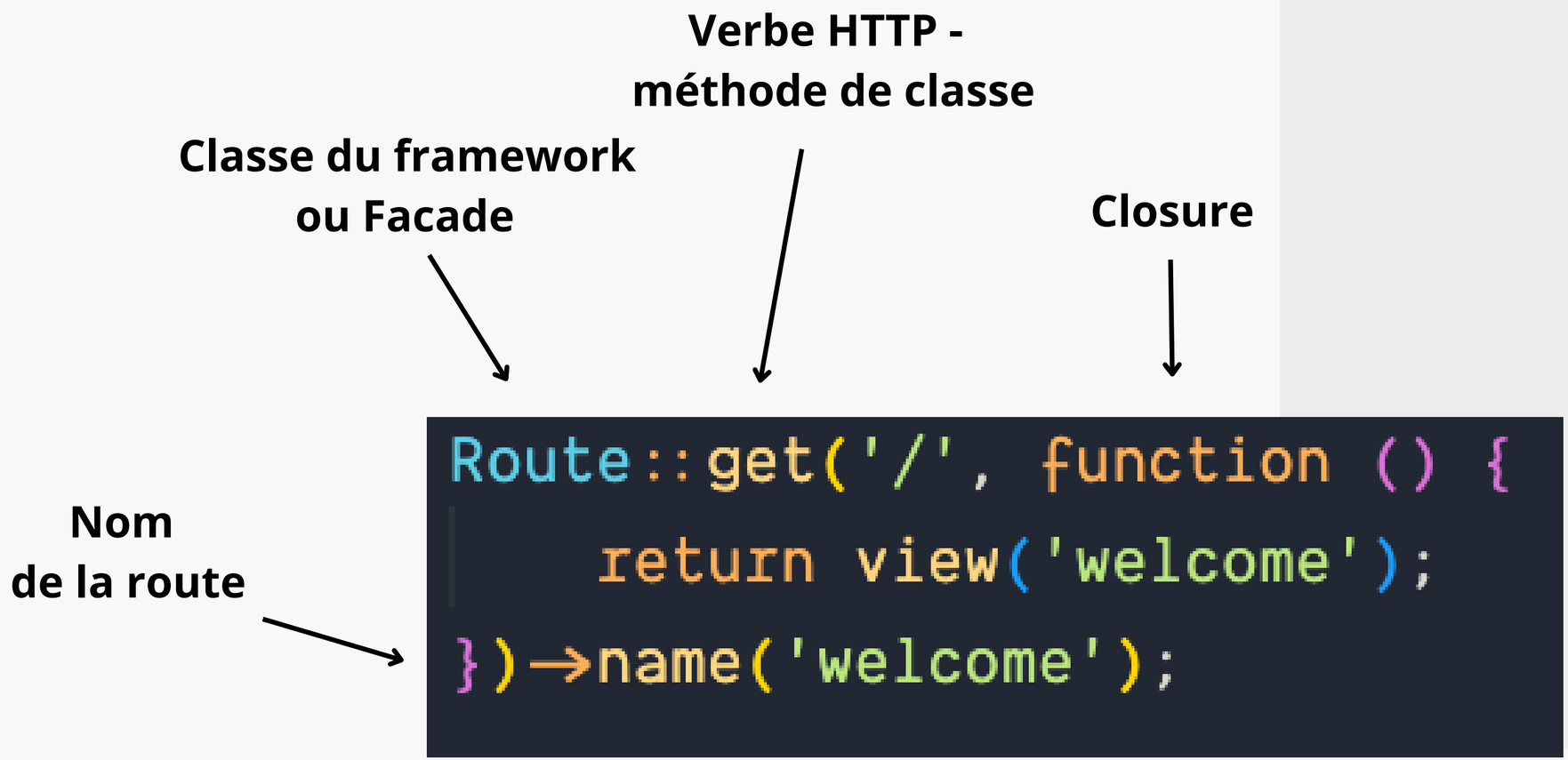
Verbes HTTP

developper.mozilla.org

Contrôleur, ou pas

Une route peux utiliser une Closure ou un Contrôleur

Routing


Verbe HTTP -
méthode de classe
Classe du framework
ou Facade
Closure
Nom
de la route
Route::get('/', function () {
return view('welcome');
})->name('welcome');

Blade

laravel.com/docs/10.x/blade

Moteur de template

Génère dynamiquement du HTML avec des données fournies par le contrôleur

Directives

Raccourcis vers des fonctionnalités PHP

Components

Comme en Vue (❤️), React, Svelte, ... C'est une portion de code que l'on peux réutiliser

Layouts

Présentation globale d'une ou plusieurs pages

Formulaires

laravel.com/docs/10.x/blade#forms

CSRF

Token présent pour s’assurer que le formulaire est autorisé à envoyer des données car sur le même domaine que le serveur cible

Directive @csrf

Methode / Verbe

Nativement, seul GET et POST sont supporté mais avec la directive @method il est possible d’utiliser d’autres verbes

Erreurs

Pour éviter de frustrer un utilisateur, on lui affiche clairement les erreurs de saisi dans le formulaire avec la directive @error

Formulaires

Directive CSRF

Définition de la route à utiliser

```blade
<form method="POST" action="{{ route('login') }}" class="flex flex-col space-y-4 w-1/3">
    @csrf

    <label for="email">E-mail</label>
    <input type="text" id="email" name="email"/>

    <label for="password">Password</label>
    <input type="text" id="password" name="password"/>

    <button
        type="button"
        class="rounded-md bg-indigo-600 px-3.5 py-2.5 text-sm font-semibold text-white shadow-sm"
    >
        Login
    </button>
</form>
```

laravel.com/docs/10.x/blade#forms

Controleurs

laravel.com/docs/10.x/controllers

Création par le CLI

php artisan make:controller NAME

Classe

Plusieurs méthodes pour différentes routes

Namespace

```php
use App\Http\Controllers\UserController;

Route::get('/user/{id}', [UserController::class, 'show']);
```

Appel du contrôleur

Validation

laravel.com/docs/10.x/validation

Assurance des données utilisateur

Présence, type, format, ... pour ne pas casser le fonctionnement

Retour d’erreurs

Simplifie le rendu des erreurs pour l’utilisateur

```php
public function login(Request $request)
{
    $validated = $request->validate([
        'email' => 'required|string',
        'password' => 'required|string|size:16',
    ]);

    // ...

    return redirect('home');
}
```

Appel du validateur

Régles / champ

Utilitaires

dd()

Permet de stopper PHP et de présenter dans le navigateur le contenu d'une variable, un texte, ...

route()

Permet de retrouve l'URL d'une route par son nom

config()

Permet de retrouver la valeur d'une configuration par sa clé

laravel.com/docs/10.x/helpers

/20

Les mains dans le cambouis

1) Création d’une branche GIT

Nommée “discovery”

2) On se place dessus

3) On suit le premier exercice

4) On pousse le code sur Github

Migrations

laravel.com/docs/10.x/migrations

Définition

Représentation d’une table en base de données, d’une modification du schema d’une table, ...

Emplacement

database/migrations

Chronologique et réversible

La date en début du nom de fichier indique l’ordre de lancement

Création

php artisan make:migration create_password_table

Migrations Commandes

laravel.com/docs/10.x/migrations

Création

php artisan make:migration create_password_table

Lancement

php artisan migrate

Retour arrière

php artisan migrate:rollback

On recommence tout

php artisan migrate:fresh

Modèles

laravel.com/docs/10.x/eloquent

Définition

Un modèle = une classe = une table en base de données

Emplacement

App\Models

Convention

Le nom d’un modèle est toujours au singulier

Création

php artisan make:model NAME

Eloquent

Définition

Eloquent est un ORM (Object Relational Mapping) permettant de simplifier les requêtes vers une base de données

laravel.com/docs/10.x/eloquent

PHP brut

```php
$pdo = new PDO($dsn, $username, $password);

$query = "SELECT name FROM users WHERE email = :email";
$stmt = $pdo->prepare($query);

$email = "donovan.broquin@iut-dijon.u-bourgogne.fr";
$stmt->bindParam(":email", $email, PDO::PARAM_STR);

$stmt->execute();


$result = $stmt->fetch(PDO::FETCH_ASSOC);
```

Laravel

```php
User::select('name')->where('email', 'donovan.broquin@iut-dijon.u-bourgogne.fr')->first();
```

CRUD

Create

```php
User::create(['name' => 'Donovan']);
```

Read

```php
User::where('name', 'Donovan')->first();
```

Update

```php
$user->update(['name' => 'Donovan']);
```

Delete

```php
$user->delete();
```

laravel.com/docs/10.x/eloquent

Relations

laravel.com/docs/10.x/eloquent-relationships

Définition

Lien entre differentes tables et donc modèles
Se définit dans le modèle sous forme de fonction

Convention

On suit la convention <nom_table_au_singulier>_id
pour nommer la colonne d'une relation

Dans le contrôleur

Méthode nommée avec un retour de type hasMany,
...

```php
public function password(): HasMany
{
    return $this->hasMany(Password::class);
}
```

Authentification

On ne réinvente pas la roue

Laravel dispose de plusieurs moyens de gérer l'authentification qui seront tous plus efficace que la votre

Breeze

L'authentification la plus simple
laravel.com/docs/10.x/starter-kits

Complet

Création de compte, connexion, déconnexion, session, réinitialisation de mot passe.

laravel.com/docs/10.x/authentication

dd()

Aperçu immédiat mais stop le script

Logs

Présents dans storage/logs/laravel.log
S’alimente en continu

Telescope

Package officiel permettant de voir tout ce qui se passe dans l’application

Debug

/29

La pratique

1) Merge la branche discovery dans main

2) En partant de main, on crée une nouvelle branche nommée “database”

3) On suit le deuxième exercice

4) On pousse le code sur Github

bit.ly/45SJJSU

/Fin

bit.ly/45SJJSU

Ressources du cours