

Chapitre 1

Installation et configuration

1.1 Présentation de Laravel

Laravel est un framework de développement web open source écrit en PHP, créé par Taylor Otwell en 2011. Il offre un grand nombre de fonctionnalités et de bibliothèques qui permettent aux développeurs de gagner du temps et de se concentrer sur la logique métier de leurs applications au lieu des tâches redondantes. Le framework laravel est conçu pour être facile à apprendre et à utiliser, avec une documentation complète¹ et une communauté active². Laravel est utilisé par de nombreuses entreprises pour développer des applications web robustes et évolutives, y compris des applications e-commerce, des applications de réservation, des plateformes de médias sociaux et des applications de gestion de contenu. Les fonctionnalités clés de Laravel incluent le routage, les contrôleurs, les vues, les migrations de base de données, l'authentification, l'autorisation, la validation de formulaires ... etc. Laravel utilise le modèle d'architecture MVC pour séparer les responsabilités de l'application et faciliter la gestion.

1. <https://laravel.com/docs/10.x/readme>

2. <https://laravel.io/>

1.2 L'architecture MVC

MVC est un acronyme pour "Modèle-Vue-Contrôleur". C'est un modèle d'architecture logicielle couramment utilisé dans le développement d'applications web et d'applications basées sur des interfaces utilisateur. L'architecture MVC sépare les responsabilités de l'application en trois composants distincts :

1.2.1 Les Modèles

C'est la partie de l'application qui représente les données. Le modèle est responsable de la manipulation des données, de l'accès aux bases de données et de la mise à jour de l'état de l'application.

1.2.2 Les Vues

C'est la partie de l'application qui représente l'interface utilisateur. Les vues sont responsables de l'affichage des données aux clients et de l'interaction avec eux. Elles peuvent contenir des éléments tels que des formulaires, des boutons, des graphiques, des tableaux et des images.

1.2.3 Les Contrôleurs

C'est la partie de l'application qui gère les interactions entre les modèles et les vues. Les contrôleurs reçoivent souvent les actions des utilisateurs, récupèrent les données nécessaires auprès des modèles et les envoient à la vue appropriée pour affichage.

L'architecture MVC aide à maintenir le code propre et bien organisé, en séparant les différentes responsabilités de l'application et en favorisant une meilleure gestion des changements et des mises à jour. Cela permet également une meilleure collaboration entre les membres d'une équipe de développement, en répartissant clairement les responsabilités de chacun.

1.3 Préparation de l'environnement

Avant de créer un projet Laravel, il y a plusieurs prérequis qu'il faut installer sur votre machine locale.

1.3.1 Un interpréteur PHP

Laravel nécessite une version récente de PHP, de préférence PHP 7.3 ou supérieure, installée sur votre machine. Selon la version de Laravel que vous voudriez installer, une version supérieure de PHP pourrait être requise. Par exemple, pour installer Laravel 10 (la dernière version au moment de la rédaction de ce cours), il est nécessaire d'avoir PHP 8.1 ou 8.2 installé sur votre machine. Vous pouvez vérifier la version de PHP installée en exécutant la commande `php -v` dans votre terminal (à condition que le répertoire d'installation de PHP soit présent sur la variable d'environnement `PATH`). Pour connaître la version de PHP requise pour l'installation de chaque version de Laravel, il faut consulter le lien en bas de page³.

1.3.2 Un SGBD

Laravel prend en charge plusieurs systèmes de gestion de bases de données, notamment MySQL, Oracle, PostgreSQL, SQL Server et SQLite. Dans notre cas nous allons utiliser MySQL car c'est le SGBD le plus populaire à associer avec PHP. Nous montrons dans la section 1.6 comment configurer un projet Laravel pour utiliser une base de données MySQL.

1.3.3 Composer

Les projets Laravel peuvent être créés à l'aide de Composer, un gestionnaire de dépendances pour PHP, pour installer les dépendances et les packages nécessaires. Il est donc nécessaire d'installer Composer sur votre machine. Vous pouvez télécharger et installer Composer facilement à partir de son site officiel⁴.

3. <https://laravel.com/docs/10.x/releases#support-policy>

4. <https://getcomposer.org/>

Il est aussi possible d'utiliser la plate-forme de virtualisation "Docker" pour lancer un projet Laravel, néanmoins, ce cas d'utilisation est un peu avancé pour ce cours, nous allons donc nous concentrer sur Composer.

1.3.4 Serveur web

Pour exécuter une application Laravel, il vous faut un serveur web tel que Apache. Les package laravel incluent un serveur web léger intégré, mais il reste recommandé d'utiliser un logiciel spécialisé pour une expérience plus adéquate et similaire à un service d'hébergement réel.

1.3.5 Node.js

Finalement, il est recommandé d'installer Node.js pour les projets Laravel vu qu'il utilise des outils de développement front-end tels que Laravel Mix, qui nécessitent l'outil npm (livré avec Node.js sur Windows) pour fonctionner.

1.3.6 Installation des prérequis

Nous vous montrons dans cette sections les étapes à suivre pour créer votre premier projet Laravel, nous commençons d'abord par l'installation de WampServer, nous passons après à l'installation de Composer, et finalement nous créons et lançons un premier projet Laravel.

Installer WampServer

WampServer est une plate-forme de développement Web sous Windows qui inclue le serveur Apache2, un interpréteur des scripts PHP et d'une base de données MySQL (et aussi MariaDB qui pourrait être une bonne alternative à MySQL). Il possède également une interface web appelé PHPMyAdmin pour gérer plus facilement vos bases de données via une interface graphique.

Si vous utiliser un autre système d'exploitation, ou que vous ne pouvez pas installer Wamp-

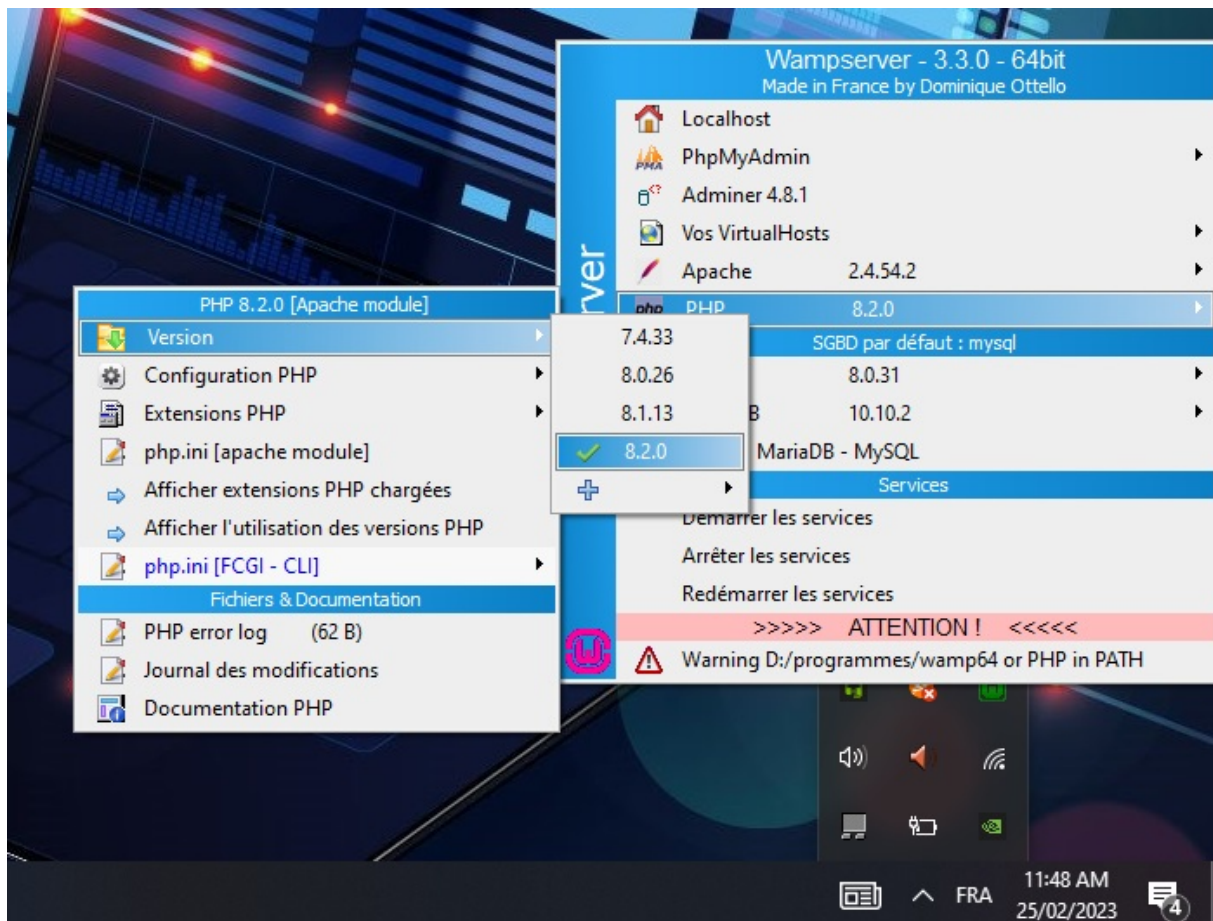


FIGURE 1.1 – Changer la version de PHP

Server, plusieurs logiciels alternatives existent. Nous citons particulièrement : XAMPP⁵ (multi plate-formes), MAMP⁶ (conçu pour MacOS), LAMP⁷ (conçu pour Linux), Easy-PHP⁸ (conçu pour Windows), Laragon⁹ (utilise Nginx au lieu de Apache) ... etc. Nous nous concentrons dans ce cours uniquement sur Wamp, mais en pratique, il n'y a pas de grandes différence entre ces différentes plate-formes.

L'installation de WampServer est une tâche simple, il suffit de le télécharger sur le site officiel¹⁰, puis lancer l'installation (en double cliquant le fichier). Une icône apparaîtra sur le bureau de votre machine à la fin de l'installation, lorsque vous exécutez WampServer après l'installation, il faut vérifier qu'une icône verte (visible sur la figure 1.1) apparaisse

5. <https://www.apachefriends.org/>

6. <https://www.mamp.info/en/>

7. <https://doc.ubuntu-fr.org/lamp>

8. <https://www.easyphp.org/>

9. <https://laragon.org/>

10. <https://www.wampserver.com/>

sur la barre des tâches, si l'icône est rouge, cela signifie que tous les services (Apache, MySQL et MariaDB) n'ont pas pu se lancer correctement. Si l'icône est orange, cela signifie qu'au moins un des services n'est pas lancé.

WampServer vient équipé de plusieurs versions de PHP, la version 3.3.0 de WampServer (la dernière version stable au moment de la rédaction de ce cours) arrive avec les versions 7.4.33, 8.0.26, 8.1.13 et 8.2.0 de PHP. La figure 1.1 vous montre comment choisir la version adéquate à votre travail. Il est recommandé d'utiliser une version 8.1 ou 8.2 de PHP pour que la dernière version de Laravel puisse être installée correctement, sinon une version antérieure sera automatiquement choisie par le gestionnaire de dépendances (Composer) plus tard.

Installer Composer

Après avoir installé WampServer (ou toute autre alternative), nous passons maintenant à l'installation de Composer, son installation est également très simple, le seul détail important durant l'installation est le choix de la version de PHP à utiliser. Il va falloir dans ce cas parcourir le répertoire d'installation de WampServer, puis choisir la même version de PHP que vous avez choisie sur WampServer (dans mon cas 8.2.0) comme le montre la figure 1.2

Notez qu'il est très important de choisir la même version de PHP que votre environnement d'exécution pour Composer, autrement, il est possible d'avoir des problèmes de compatibilité entre les bibliothèques installées par Composer, et la version PHP utilisée pour l'exécution.

1.4 Création d'un projet Laravel

Une fois que les prérequis sont préparées, nous procédons à l'installation de Laravel, pour ce faire, nous allons d'abord ouvrir une fenêtre de commandes ("Win + r" puis exécuter la commande "cmd"), puis nous déplacer vers le répertoire sur lequel nous voulons créer le projet (avec la commande "cd") puis exécuter la commande suivante :

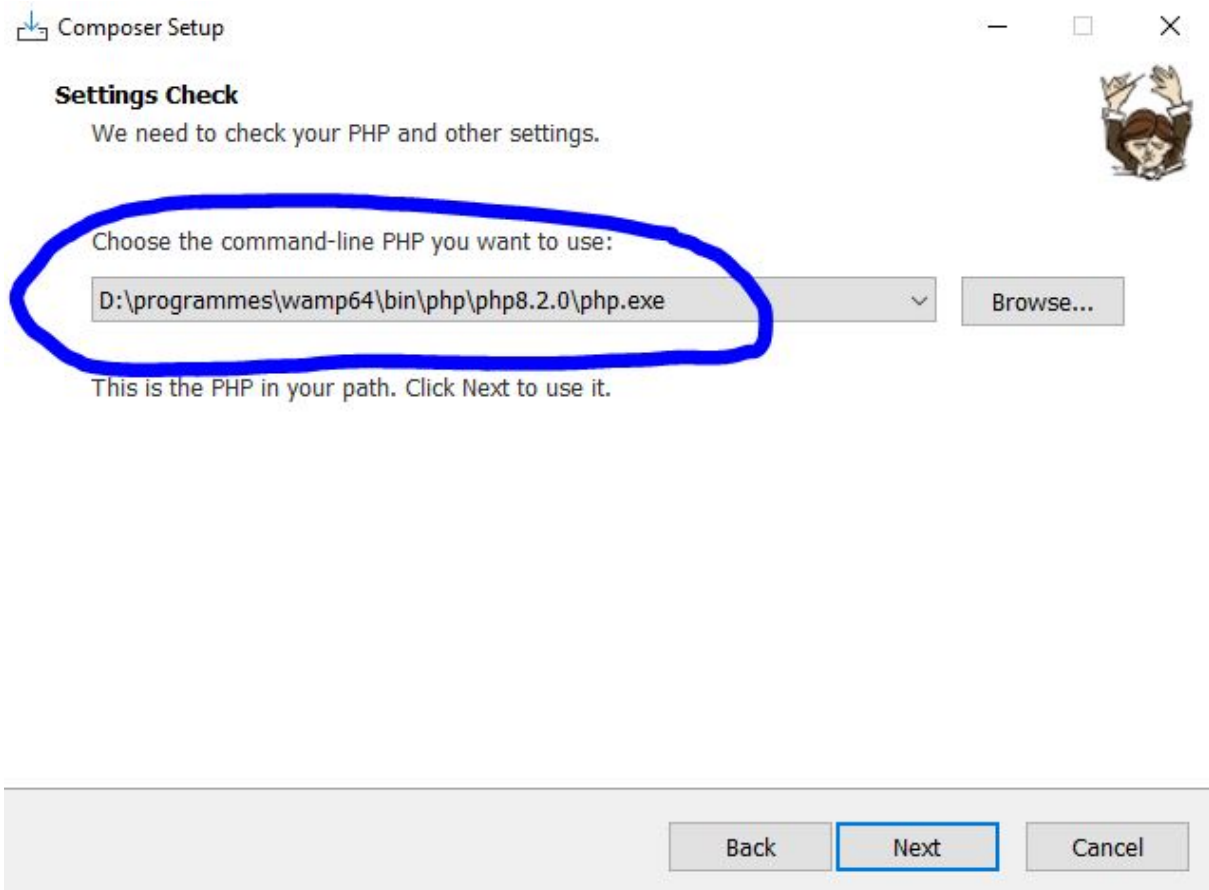


FIGURE 1.2 – Parcourir la version de PHP durant l’installation de Composer

```
composer create-project laravel/laravel <nom-de-projet>
```

puis attendre que composer télécharge et installe les bibliothèques requise. Une autre façon de créer un projet Laravel est d’installer globalement la commande `laravel` à l’aide de composer, puis l’utiliser afin de créer votre projet. Les commandes à écrire dans ce cas sont :

```
composer global require laravel/installer
laravel new <nom-de-projet>
```

Pour tester par la suite que votre projet a bien été installer, il faut déplacer sur le répertoire de projet, puis utiliser la CLI (Command-Line Interface) de `artisan` pour lancer le projet à l’aide des deux lignes suivante :

```
cd <nom-de-projet>
php artisan serve
```

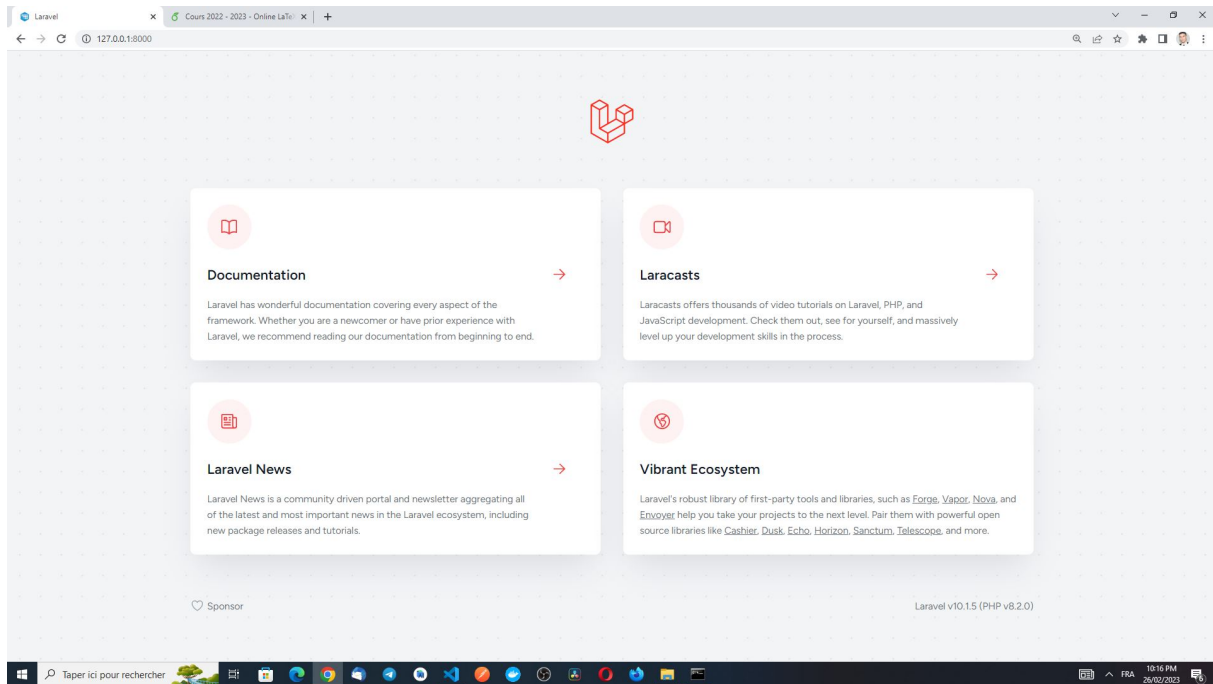


FIGURE 1.3 – Page d'accueil par défaut du projet Laravel

Pour bien vérifier que le projet a été bien lancé, vous devez ouvrir l'URL affiché après la dernière commande (`http://127.0.0.1:8000`) dans n'importe quel navigateur. Une page d'accueil similaire à celle dans la figure 1.3 doit s'afficher (la page pourrait changer pour des versions différentes de Laravel, mais cela n'a pas de grande importance).

Il est recommandé de créer votre projet dans le répertoire `www` de WampServer, car cela nous permet de lancer le projet en utilisant Apache au lieu de compter sur le serveur fourni qui sert principalement à faire des petits tests mais pas plus.

1.5 La structure du projet

La structure de base d'un projet Laravel est divisée en plusieurs répertoires et fichiers. Il est important de comprendre la structure du projet pour pouvoir utiliser ce framework d'une façon efficace. Les répertoires et les fichiers qui nous intéressent ici sont les suivants :

- Le répertoire "`app/Models`" contient les modèles (les classes) de notre projet.
- Le répertoire "`resources/views`" contient les différentes interfaces graphiques (views) de l'application.

- Le répertoire "app/Http/Controllers" contient les contrôleurs.
- Le répertoire "routes" contient les fichiers de routage du projet qui permettent de définir les routes qui seront utilisées pour accéder aux différentes fonctionnalités implémentées dans les contrôleurs.
- Le répertoire "database/migrations" qui contient les fichiers de migration pour la base de données.
- Le fichier ".env" utilisé pour stocker des informations de configuration spécifiques à l'environnement, telles que les informations de connexion à la base de données, les clés d'API, les variables d'environnement ... etc.

Nous allons expliquer dans la section suivante les variables du fichier ".env", les autres seront vus dans les prochains chapitres.

1.6 Configuration de l'environnement

Le fichier ".env" toutes les variables d'environnement pour votre application. Pour configurer le fichier ".env" il faut l'ouvrir dans n'importe quel éditeur. Les variables les plus couramment modifiées sont les suivantes :

- APP_NAME : le nom de votre application.
- APP_ENV : l'environnement de votre application (production ou développement).
- DB_CONNECTION : le pilote de base de données que vous utilisez (mysql, postgresql, sqlite ... etc).
- DB_HOST : l'hôte de votre base de données.
- DB_PORT : le port de votre base de données.
- DB_DATABASE : le nom de la base de données que vous utilisez.
- DB_USERNAME : le nom d'utilisateur de la base de données.
- DB_PASSWORD : le mot de passe de la base de données.

Une fois que vous avez configuré le fichier ".env", vous pouvez utiliser les variables d'environnement dans votre application Laravel en utilisant la fonction "env()". Par exemple, pour accéder à la variable d'environnement APP_NAME, vous pouvez utiliser la fonction "env('APP_NAME')".