



COURS FRAMEWORK WEB

CHAPITRE 2: INSTALLATION DU FRAMEWORK LARAVEL

Enseignantes responsables:
Nahla Haddar & Amal Bouaziz

Audiences: D-LSI-ADBD
Année-universitaire: 2023-2024

Un framework ?

- On peut imaginer un framework comme un ensemble d'outils à disposition.
- L'utilité d'un framework est d'éviter de passer du temps à développer ce qui a déjà été fait par d'autres souvent plus compétents et qui a en plus été utilisé et validé par de nombreux utilisateurs.
- Un framework fait gagner du temps et donne l'assurance de disposer de composants bien codés et fiables



Laravel

- Laravel est un framework d'application Web PHP gratuit et open source,
 - Fondé en 2011 par Taylor Otwell,
 - Il est orienté objet.
 - Il utilise le pattern MVC
 - La version actuelle est V9
 - **Pourquoi Laravel?**
 - *Bien documenté*
 - *Offre une structure claire au langage PHP.*
 - *Complet et très puissant*
 - *Offre environ 9000 packages*
 - *Classé n°1 parmi les framework PHP*
- <https://www.excellentwebworld.com/best-php-frameworks/>

3

Chapitre 3: Installation du Framework laravel

° ENVIRONNEMENT DE TRAVAIL

4

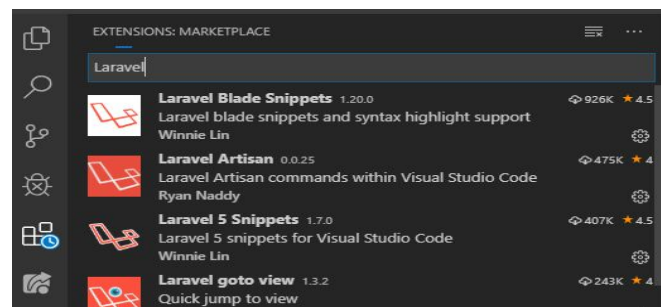
Les pré-requis techniques

- Environnement de travail nécessaire:
 - Un éditeur de code telque VSCode
 - **PHP V8**
 - Un serveur web Apache
 - Un serveur de BD MySQL
 - **Composer v.2:** <https://getcomposer.org/download/>
 - **Composer** est un outil de gestion des dépendances de packages en PHP.
 - Il vous permet de déclarer les bibliothèques dont dépend votre projet et il les gèrera (installer/mettre à jour) pour vous.

5

Configuration de l'IDE Visual Studio Code

- Le grand avantage de cet IDE, est de proposer de très nombreux plugins pour étendre ses possibilités, et il y en a pas mal pour Laravel.
- Pour trouver ceux qui concernent Laravel vous avez une zone de recherche:



6

Installation de l'environnement de travail

- Deux solutions sont possibles:
 - **1ère solution:**
 - Installer un environnement web en local tel que XAMPP, EASYPHP, CADDY,... Cela vous permettra d'avoir PHP, Apache et MySQL installés en local
 - Télécharger et installer Composer
 - **2ème solution:**
 - Installer **Laragon**: un environnement de développement web assez complet:
 - Offre un serveur Apache, serveur de BD, Composer, ...

7

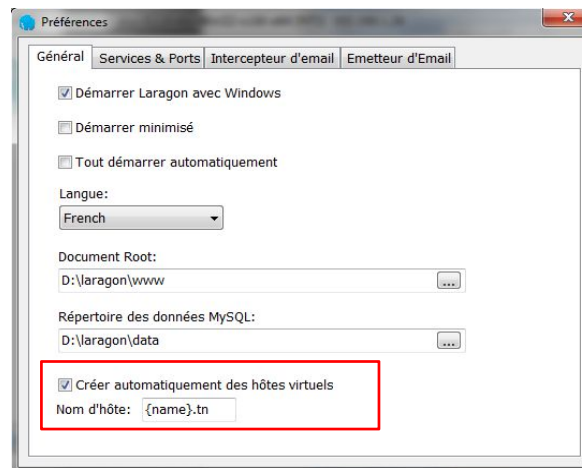
Installation de l'environnement de travail

- **Laragon et hôte virtuel:**
 - Laragon crée automatiquement des hôtes virtuels pour les dossiers qui se trouvent sur le serveur (**www**).
 - Le fichier de démarrage de Laravel est placé en **www/monsite/public/index.php**
 - Donc à partir de **localhost** il faut entrer **http://localhost/monsite/public**. Ce n'est ni pratique ni élégant.
 - Un hôte virtuel permet de définir une adresse simplifiée. Par exemple ici Laragon va définir automatiquement **monsite.test**.

8

Installation de l'environnement de travail

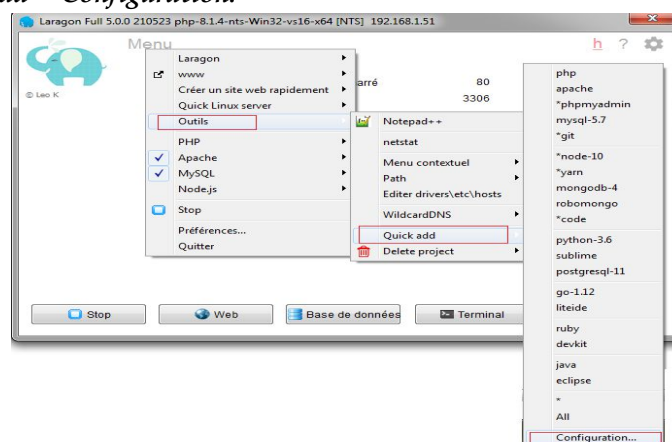
- **Laragon et hôte virtuel:**
 - Vous pouvez aller aux préférences de Laragon et personnaliser le domaine de votre hôte virtuel



9

Installer PhpMyAdmin avec Laragon

- **S'assurer que Laragon utilise la dernière version de phpMyAdmin:**
 1. Dans Laragon, sélectionnez *Menu* → *Outils* → *Quick-add* → *Configuration*.

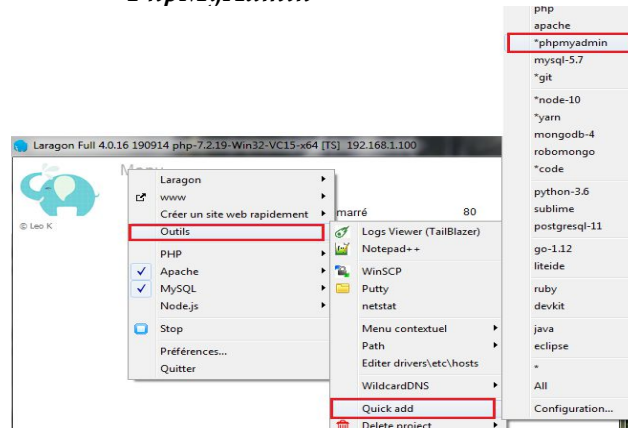


10

Installer PhpMyAdmin avec Laragon

- **Installation de PhpMyAdmin:**

1. Allez à **laragon\etc\apps** et supprimez le dossier **PhpMyAdmin** s'il existe
2. Dans Laragon, allez à **Menu → Outils → Quick add → PhpMyAdmin**

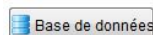


11

Accès à PhpMyAdmin

- Attendez quelques secondes et le navigateur sera lancé avec la page de configuration suivante:
 - **Utilisateur / Username = root**
 - **Mot de passe / Password = laissez vide**
 - puis cliquez sur « Go »
- Le navigateur doit vous afficher le home de PhpMyAdmin pour gérer vos bases de données.

Notez bien: Vous pouvez à tout moment accéder au home de PhpMyAdmin à travers le menu de laragon



12

Chapitre 3: Les bases du Framework laravel

° INSTALLATION DE LARAVEL

13

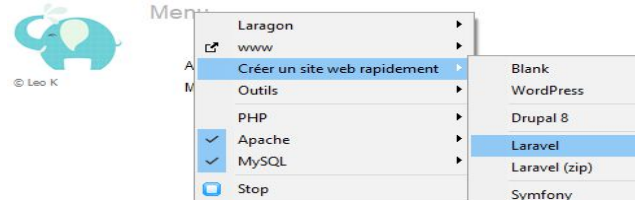
Installation de Laravel

- Installation => création d'un projet
 - ◆ Création d'un **dossier** sous **www** de votre serveur Apache portant le **nom du projet**
 - ◆ Le **dossier** aura la **squelette** du **framework Laravel** et contiendra toutes ses **librairies**.
- Trois solutions possibles pour créer un projet:
 - Avec Laragon
 - Avec Composer
 - Avec Laravel Installer
- Vous devez être connecté sur Internet !

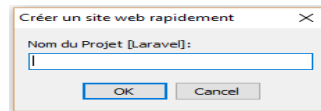
14

Installation avec Laragon

- Utilisez le Menu > Créer un site web rapidement > Laravel :



- On vous demande alors le nom du projet (donc du dossier) :



- Une base de données est automatiquement créée avec le même nom !

15

Installation avec Composer

- Par exemple, je veux créer une application dans un dossier **MonSite** à la racine de mon serveur, voici la syntaxe à exécuter sur un terminal:

- On se pointe sur la racine du serveur:

```
cd laragon\www
```

- On exécute la commande:

```
composer create-project laravel/laravel MonSite
```

- L'installation démarre
- Attendez quelques minutes pour que Composer fasse son travail jusqu'au bout.

16

Installation avec Laravel Installer

1. On se pointe sur la racine du serveur:

```
cd laragon\www
```

2. Il faut commencer par installer globalement l'installateur avec Composer :

```
composer global require laravel/installer
```

- Pour certaines versions de Windows, il faut ensuite informer la variable d'environnement **path** de l'emplacement du dossier:
 - %USERPROFILE%\AppData\Roaming\Composer\vendor\bin.
- 3. Pour créer une application il suffit de taper:

```
laravel new MonSite
```

17

Lancement du projet

- Pour lancer le projet sur le serveur:
 - **Avec Laragon:** allez à: Menu>www>MonSite
 - Ou écrivez directement dans le navigateur:

```
http://localhost/MonSite/public
ou
http://monsite.test
```

- Vous devez avoir la page d'accueil de Laravel:



Documentation

Laravel has wonderful, thorough documentation covering every aspect of the framework. Whether you are new to the framework or have previous experience with Laravel, we recommend reading all of the documentation from beginning to end.

Laracasts

Laracasts offers thousands of video tutorials on Laravel, PHP, and JavaScript development. Check them out, see for yourself, and massively level up your development skills in the process.

18

Chapitre 3: Organisation d'un projet laravel

PRINCIPE DE FONCTIONNEMENT

19

Organisation d'un projet Laravel 9

```

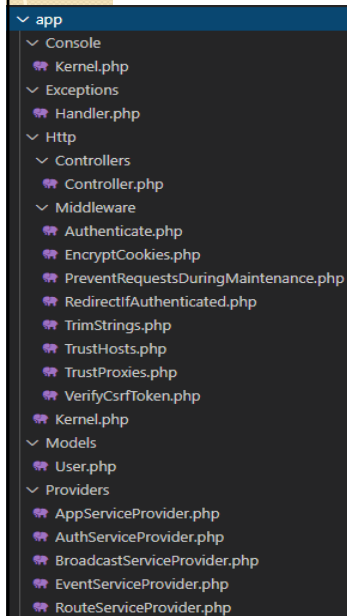
> app
> bootstrap
> config
> database
> lang
> public
> resources
> routes
> storage
> tests
> vendor
⚙ .editorconfig
⚙ .env
⚙ .gitattributes
⚙ .gitignore
! .styleci.yml
≡ artisan
{} composer.json
{} composer.lock
{} package.json
🔗 phpunit.xml
📖 README.md
JS webpack.mix.js

```

- **app**: Ce dossier contient les éléments essentiels de l'application
- **bootstrap** : scripts d'initialisation de Laravel,
- **config**: toutes les configurations concernant l'application, authentification, base de données, espaces de noms, emails, session...
- **database** : migrations et populations,
- **lang**: fichiers de langage
- **public**: C'est le point d'entrée de l'application - chaque requête HTTP passe forcément par ce dossier, contient le **contrôleur frontal index.php**
- **resources** : vues, et assets (par exemple les fichiers CSS et JS),
- **routes** : la gestion des urls d'entrée de l'application,
- **storage** : données temporaires de
- **tests** : fichiers de tests unitaires,
- **vendor** : tous les packages de Laravel et de ses dépendances (créé par composer).

20

Organisation d'un projet Laravel

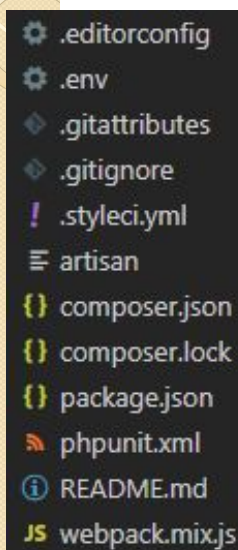


• Dossier app:

- **Console** : toutes les commandes en mode console,
- **Exceptions** : pour gérer les erreurs d'exécution,
- **Http** : tout ce qui concerne la communication : contrôleurs, middlewares (il y a 7 middlewares de base qui servent à filtrer les requêtes HTTP) et le kernel,
- **Providers** : tous les fournisseurs de services (providers), il y en a déjà 5 au départ. Les providers servent à initialiser les composants.
- **Models** : le dossier des modèles avec déjà un présent qui concerne les utilisateurs.

21

Organisation d'un projet Laravel



• Fichiers de la racine:

- **Fichier [.env]**: c'est le fichier de configuration chargé lors du lancement de l'application
- **Fichier [composer.json]**: enregistre tous les **packages** utilisés par votre projet.
 - Ils sont installés automatiquement dans le dossier `vendor` lors de l'initialisation du projet

22

Installation et mise à jour des packages avec Composer

- **Installer les packages manuellement:**
 - La commande `composer install`
- **Mettre à jour les packages:**
 - La commande `composer update`
- **Ajouter un nouveau package:**
 - La commande `composer require monpackage`
- Pour une documentation complète sur les commandes Composer:
 - <https://getcomposer.org/doc/03-cli.md>

23

Chapitre 3: Installation du Framework laravel

◦ PRINCIPE DE FONCTIONNEMENT

24

Réception de la requête

htaccess et index.php

- Pour Laravel on veut que toutes les requêtes aboutissent obligatoirement sur le fichier **index.php** situé dans le dossier **public**.
- Alors au lieu d'écrire ce genre d'URL:

```
http://monsite.fr/index.php/mapage
```

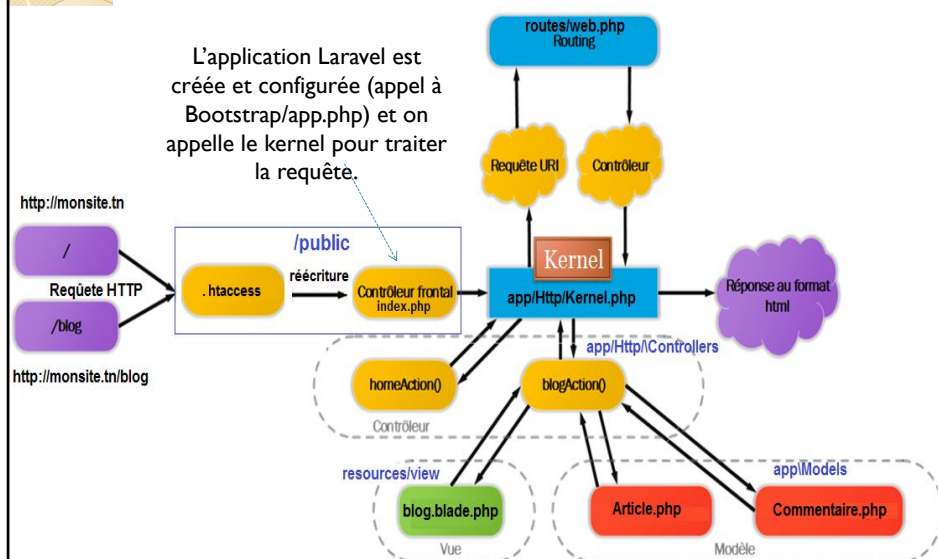
- Il y a un fichier **.htaccess** dans le dossier **public** de Laravel avec une règle de réécriture de telle sorte qu'on peut avoir une url simplifiée :

```
http://monsite.fr/mapage
```



25

Fonctionnement selon le Modèle MVC



26