



# Laravel

Maacha Otmane



## Constitution de Laravel.

- Laravel, créé par Taylor Otwell, initie une nouvelle façon de concevoir un framework en utilisant ce qui existe de mieux pour chaque fonctionnalité.
- Laravel est un framework web open-source écrit en PHP respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet. Laravel est distribué sous licence MIT, avec ses sources hébergées sur GitHub.



Vous allez trouver dans Laravel :

- un système de routage perfectionné (RESTful et ressources),
- un créateur de requêtes SQL et un ORM performants,
- un moteur de template efficace,
- un système d'authentification pour les connexions,
- un système de validation,
- un système de pagination,
- un système de migration pour les bases de données,
- un système d'envoi d'emails,
- un système de cache,
- une gestion des sessions...

# Installation et organisation



## 1. Composer

Laravel utilise des composants d'autres sources. Plutôt que de les incorporer directement donc il utilise un gestionnaire de dépendance : Composer. D'ailleurs pour le coup les composants de Laravel sont aussi traités comme des dépendances. **Mais c'est quoi un gestionnaire de dépendance ?**

Imaginez que vous créez une application PHP et que vous utilisez des composants issus de différentes sources : Carbon pour les dates, Redis pour les données... Vous pouvez utiliser la méthode laborieuse en allant chercher tout ça de façon manuelle, mais vous allez être confronté à des difficultés :

- télécharger tous les composants dont vous avez besoin et les placer dans votre structure de dossiers,
- traquer les éventuels conflits de nommage entre les librairies,
- mettre à jour manuellement les librairies quand c'est nécessaire,

Tout ça est évidemment faisable mais avouez que s'il était possible d'automatiser les procédures ce serait vraiment génial. C'est justement ce que fait un gestionnaire de dépendances !



## 1-1. Installation

Laravel utilise Composer comme gestionnaire de dépendances. Il vous faut donc commencer par l'installer sur votre ordinateur.

Pour l'installation dans Windows il suffit de télécharger un setup qui fait tout très proprement et renseigne aussi la variable d'environnement PATH, ce qui permet ensuite d'utiliser Composer à partir de n'importe quel emplacement. Par contre l'installateur vous demandera où se trouve php.exe et vous devrez répondre. En effet Composer est un fichier PHP et a besoin d'être exécuté.

# Créer une application Laravel



## 1.Installation avec Composer

**composer create-project laravel/laravel name-projet**

## 2.Installation avec Laravel Installer

Une autre solution pour installer Laravel consiste à utiliser l'installateur. Il faut commencer par installer globalement l'installateur avec Composer :

**composer global require laravel/installer**

Pour créer une application il suffit de taper :

**laravel new name-projet**

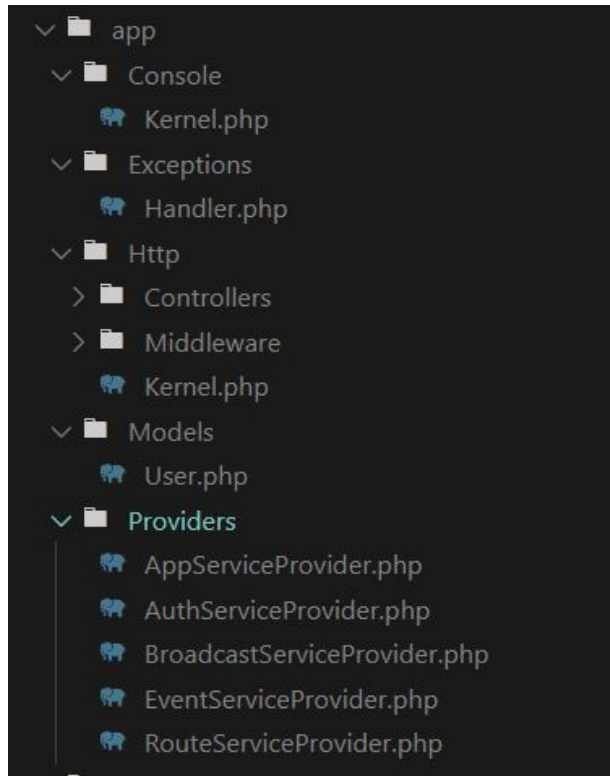
# Organisation de Laravel



## Dossier app

Ce dossier contient les éléments essentiels de l'application :

- **Console/Commands** : ici on mettra toutes les commandes en mode console, il y a au départ une commande `Inspire` qui sert d'exemple,
- **Http** : ici on va trouver tout ce qui concerne la communication : contrôleurs, middlewares,
- **Providers** : ici on va mettre tous les providers. Les providers servent à initialiser les composants.



# Autres dossiers

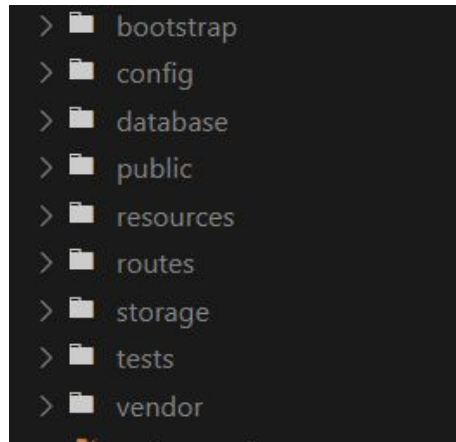
Voici une description du contenu des autres dossiers :

- **bootstrap** : scripts d'initialisation de Laravel pour le chargement automatique des classes, la fixation de l'environnement et des chemins, et pour le démarrage de l'application,
- **config** : toutes les configurations : application, authentification, cache, base de données, espaces de noms, emails, systèmes de fichier, session...
- **database** : ici on aura les migrations et les populations,
- **public** : tout ce qui doit apparaître dans le dossier public du site : images, CSS, scripts...
- **resources** : ici on aura les vues, les fichiers de langage et les assets (par exemple les fichiers LESS ou Sass),
- **storage** : pour stocker les données temporaires de l'application : vues compilées, caches, clés de session...
- **tests** : pour les fichiers de tests unitaires.
- **vendor** : tous les composants de Laravel et de ses dépendances,

## Fichiers de la racine

Il y a un certain nombre de fichiers dans la racine dont voici les principaux :

- **artisan** : outil en ligne de Laravel pour des tâches de gestion,
- **composer.json** : fichier de référence de Composer,
- **phpunit.xml** : fichier de configuration de phpunit (pour les tests unitaires),
- **.env** : fichier pour spécifier l'environnement d'exécution.





# Artisan



Artisan est l'interface de ligne de commande incluse avec Laravel. Il fournit un certain nombre de commandes utiles qui peuvent vous aider lors de la création de votre application. Pour afficher une liste de toutes les commandes Artisan disponibles, vous pouvez utiliser la commande list:

`php artisan list`

Chaque commande comprend également un écran d'aide " " qui affiche et décrit les arguments et options disponibles de la commande. Pour afficher un écran d'aide, faites précéder le nom de la commande de help:

`php artisan help migrate.`

# Migration



Une migration permet de créer et de mettre à jour un schéma de base de données. Autrement dit vous pouvez créer des tables, des colonnes dans ces tables, en supprimer, créer des index... Tout ce qui concerne la maintenance de vos tables peut être prise en charge par cet outil.

# Eloquent



Laravel propose un ORM (acronyme de object-relational mapping ou en bon Français un mappage objet-relationnel) très performant. De quoi s'agit-il ? Tout simplement que tous les éléments de la base de données on une représentation sous forme d'objets manipulables. Quel intérêt ? Tout simplement de simplifier grandement les opérations sur la base