

1 - Introduction

Laravel est un framework PHP élégant, expressif et flexible avec un accent extrême sur le code propre et la vitesse qui se décrit comme "Le framework PHP pour les artisans du Web". Il s'agit d'un framework PHP gratuit et Open source créé par Taylor Otwell, basé sur l'architecture Model View Controller (MVC).

Créer une application Web à partir de zéro peut être intimidant, surtout si vous êtes débutant. Une simple application Web contient également divers éléments, petits et grands, et créer ces éléments à chaque fois que vous créez une application Web peut être ennuyeux et répétitif et il ne sert à rien de réinventer les roues. C'est alors que Laravel vient à votre secours.

Le framework Laravel fournit diverses bibliothèques PHP et fonctions d'assistance et peut vous aider à vous concentrer sur des éléments plus importants tout en fournissant des fonctions et une logique communes pour accélérer le temps de développement et faciliter le développement.

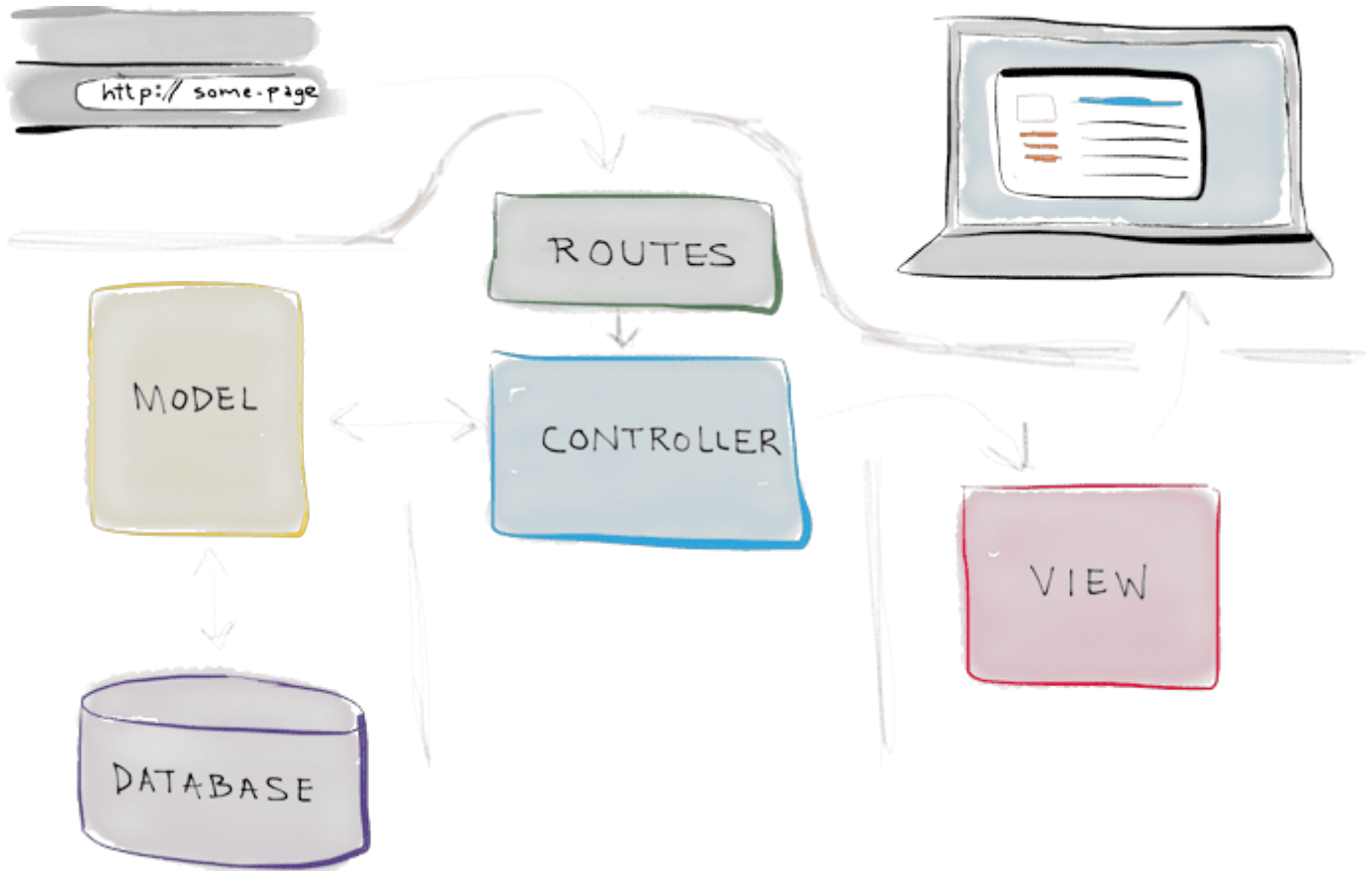
Référence: <https://www.parthpatel.net/laravel-tutorial-for-beginner/>

2 - Ce que vous devez savoir pour utiliser Laravel.

- HTML/CSS
 - PHP intermédiaire - concepts tels que PHP OOP, abstraction, etc.
 - Compréhension de base du framework MVC
 - Base de données relationnelle
-

3 - Modèle-Vue-Contrôleur

Les applications Laravel suivent le modèle de conception d'architecture Modèle-Vue-Contrôleur.



- Les **modèles** représentent les entités de la base de données et vous aident à interroger la base de données et à renvoyer les données
- Les **vues** sont les pages qui seront affichées lors de l'accès à l'application. View Component est utilisé pour l'interface utilisateur de l'application.
- Les **contrôleurs** gèrent les demandes des utilisateurs, obtiennent les données requises des modèles et les transmettent aux vues. Les contrôleurs servent d'intermédiaire entre les composants de modèle et de vue pour traiter la logique métier et la demande entrante.

4 - Exploration de la structure des répertoires.

Lorsque vous avez installé composer et créé votre première application Web Laravel, vous avez peut-être remarqué le dossier de l'application avec différents fichiers et dossiers.

- **App** : Ce répertoire est la viande de l'application et contient le code de base.
 - **Console** : ce répertoire contient toutes les commandes Artisan personnalisées créées à l'aide de make:command
 - **Exceptions** : ce répertoire contient le gestionnaire d'exceptions de l'application et est un bon endroit pour ajouter des classes d'exceptions personnalisées pour gérer différentes exceptions levées par votre application
 - **Http** : Ce répertoire contient tous vos contrôleurs, middleware et requêtes
 - **Models** : Il s'agit d'un nouveau directeur ajouté depuis Laravel 8 pour contenir les fichiers de modèle. Auparavant, les modèles étaient stockés dans le dossier App, mais maintenant, ils peuvent également être stockés dans les dossiers App/Models.
 - **Providers** : Ce répertoire contient tous vos fournisseurs de services pour l'application.
- **Bootstrap** : ce répertoire contient le bootstrap du framework ainsi que les fichiers de

configuration. Il contient également le répertoire Cache qui contient les fichiers de cache générés par le framework

- **Config** : ce répertoire contient tous les fichiers de configuration de votre application.
 - **Database** : ce répertoire contient toutes les migrations de base de données et les graines. Vous pouvez également stocker le fichier de base de données SQLite ici
 - **Public** : ce répertoire contient des ressources telles que des images, des fichiers js et des CSS.
 - **Resources** : ce répertoire contient tous les fichiers de vue et les fichiers CSS ou LESS ou SASS. Il contient également le répertoire lang pour stocker les fichiers de langue.
 - **Routes** : ce répertoire contient toutes les définitions de routes pour l'application. php est le fichier qui reçoit toutes les requêtes adressées à votre application et vous pouvez ici rediriger les requêtes vers leurs méthodes de contrôleur respectives.
 - **Storage** : ce répertoire contient des modèles de lame, des fichiers de session, des fichiers de cache et autres.
 - **Tests** : Ce répertoire contient tous les fichiers de test
 - **Vendor** : ce répertoire contient toutes les dépendances du compositeur
-