
Introduction à Laravel

Cette formation Laravel vous apprendra à mettre en œuvre le routage, le moteur de template, le constructeur de formulaire, le système d'authentification, le créateur de requête SQL et un ORM performants (Eloquent), ainsi que d'autres éléments encore tel que l'envoi de mail.

Routage (Routing) : Laravel propose un système de routage qui permet de définir facilement les points d'accès de votre application. Vous pouvez spécifier les URI (Uniform Resource Identifier) et les actions associées à ces URI.

Moteur de template : Laravel utilise le moteur de template Blade, qui offre une syntaxe simple et expressive pour la création de vues. Il facilite l'inclusion de données dynamiques dans vos pages.

Constructeur de formulaire : Laravel propose des outils pour créer et gérer des formulaires de manière simple et élégante. Cela inclut la validation côté serveur et la rétroaction facile pour l'utilisateur.

Système d'authentification : Laravel intègre un système d'authentification complet, avec des fonctionnalités telles que l'inscription, la connexion, la réinitialisation de mot de passe, etc. Il simplifie grandement la mise en œuvre de l'authentification dans votre application.

Créateur de requête SQL et ORM performants (Eloquent) : Eloquent est l'ORM (Object-Relational Mapping) de Laravel. Il permet d'interagir avec la base de données en utilisant des objets plutôt que des requêtes SQL brutes. Cela simplifie la manipulation des données et améliore la lisibilité du code.

Envoi de mails : Laravel offre une API simple pour envoyer des e-mails. Vous pouvez facilement configurer les paramètres SMTP et envoyer des e-mails à partir de votre application.

Pourquoi un nouveau framework PHP ?

Laravel a été introduit en tant que nouveau framework PHP pour plusieurs raisons, et son adoption a été largement soutenue en raison des limitations perçues dans les solutions maison et d'autres frameworks PHP existants.

Quelques raisons pour lesquelles Laravel a gagné en popularité et pourquoi de nombreux développeurs ont choisi de l'adopter :

Élégance de la syntaxe : Laravel offre une syntaxe propre et expressive, grâce notamment à son moteur de template Blade. Cela rend le code plus lisible et facilite le développement, tout en minimisant la quantité de code nécessaire pour réaliser des tâches courantes.

Routage puissant : Le système de routage de Laravel est simple à utiliser et puissant. Il permet de définir facilement des points d'entrée pour l'application, facilitant ainsi la gestion des URI et des actions associées.

ORM performant (Eloquent) : Eloquent simplifie considérablement l'interaction avec la base de données. Il permet aux développeurs de travailler avec des objets au lieu de requêtes SQL brutes, ce qui rend le code plus lisible et facilite la maintenance.

Système d'authentification intégré : Laravel propose un système d'authentification complet, prêt à l'emploi. Cela permet de gagner du temps lors du développement d'applications nécessitant une gestion des utilisateurs et des sessions.

Outils de test intégrés : Laravel est livré avec des outils de test intégrés, facilitant la création et l'exécution de tests unitaires et fonctionnels. Cela favorise le développement piloté par les tests (TDD) et garantit la stabilité de l'application.

Mises à jour régulières : Laravel est activement maintenu et mis à jour par la communauté. Les mises à jour régulières ajoutent de nouvelles fonctionnalités, améliorent la sécurité et garantissent la compatibilité avec les dernières versions de PHP.

Documentation complète : Laravel propose une documentation détaillée et bien organisée, facilitant l'apprentissage et la compréhension du framework. Cela permet aux développeurs de s'initier rapidement et de résoudre efficacement les problèmes.

En ce qui concerne les limitations des solutions maison, celles-ci peuvent inclure un manque de modularité, une difficulté à maintenir un code propre et évolutif, ainsi que des défis liés à la sécurité. Les frameworks comme Laravel visent à résoudre ces problèmes en offrant une structure bien définie, des conventions de codage et des fonctionnalités intégrées qui permettent aux développeurs de se concentrer davantage sur la logique métier plutôt que sur les détails techniques.

Les apports de Laravel

Laravel apporte plusieurs avantages significatifs au développement web PHP, contribuant à sa popularité croissante parmi les développeurs.

Quelques-uns des apports majeurs de Laravel :

Expressivité de la syntaxe : Laravel utilise une syntaxe propre et expressive qui facilite la lecture et la rédaction du code. Cela contribue à accroître la productivité des développeurs et à rendre le développement plus agréable.

Outils de migration et de schéma de base de données : Laravel propose des outils de migration qui simplifient la gestion des schémas de base de données. Les migrations permettent de versionner facilement la structure de la base de données et de la mettre à jour de manière cohérente.

Système de gestion de la configuration : Laravel offre un système de configuration qui permet de gérer facilement les paramètres de l'application. Cela facilite la personnalisation de l'application en fonction des besoins spécifiques du projet.

Intégration de Composer : Laravel utilise Composer, un gestionnaire de dépendances pour PHP, facilitant l'ajout et la gestion des bibliothèques tierces dans le projet.

Structure d'une application Laravel

Laravel encourage une structure de projet bien organisée pour faciliter le développement, la maintenance et la compréhension du code.

ci-dessous une vue d'ensemble de la structure d'une application Laravel typique :

Dossiers de base :

app : Contient le code source de l'application, y compris les modèles (models), les contrôleurs (controllers), les middleware, etc.

bootstrap : Contient les fichiers nécessaires au processus de démarrage de l'application.

config : Contient les fichiers de configuration de l'application.

database : Contient les migrations (database/migrations), les seeds (database/seeds), et le dossier factories pour la création de fausses données.

public : Contient les fichiers publics, tels que les fichiers CSS, JavaScript, les images, etc. Le point d'entrée de l'application, index.php, se trouve ici.

resources : Contient les vues (views), les fichiers de langage, les fichiers de style (resources/sass), les fichiers JavaScript (resources/js), etc.

routes : Contient les fichiers de définition des routes de l'application.

storage : Contient les fichiers générés par l'application, tels que les fichiers de logs, les fichiers temporaires, etc.

tests : Contient les tests de l'application.

Fichiers spécifiques :

.env : Fichier de configuration environnementale où sont définies les variables d'environnement telles que les informations de base de données, les clés secrètes, etc.

artisan : Le script en ligne de commande pour l'artisan, le générateur de code de Laravel.

composer.json : Fichier de configuration pour Composer, le gestionnaire de dépendances pour PHP.

Dossier vendor :

Contient les dépendances de Composer.

Fichiers racine :

index.php : Point d'entrée de l'application, il charge l'autoloader et démarre le framework.

server.php : Utilisé pour le serveur web intégré de PHP lors du développement.

webpack.mix.js : Fichier de configuration pour Laravel Mix, qui simplifie l'intégration de Webpack pour la compilation des ressources front-end.

Cette structure favorise la maintenance, la collaboration et la réutilisation du code.

Environnement de développement

Pour développer des applications avec Laravel, vous aurez besoin d'un environnement de développement configuré correctement.

Les étapes de base pour configurer un environnement de développement Laravel :

Prérequis :

PHP : Laravel nécessite PHP. Assurez-vous d'installer une version récente de PHP (de préférence PHP 7.x ou supérieure). Vous pouvez utiliser des outils comme XAMPP, WampServer (sur Windows), ou MAMP (sur Mac) pour installer PHP et Apache/MySQL.

Composer : Laravel utilise Composer comme gestionnaire de dépendances. Installez Composer en suivant les instructions sur le site officiel : <https://getcomposer.org/download/>

Node.js et NPM : Pour gérer les dépendances front-end, vous aurez besoin de Node.js et NPM (Node Package Manager). Vous pouvez les télécharger depuis le site officiel : <https://nodejs.org/en>

Installation de Laravel :

Utilisez Composer pour installer Laravel en exécutant la commande suivante dans votre terminal :

```
composer create-project --prefer-dist laravel/laravel projet_uahb
```

Cela créera un nouveau dossier contenant l'installation fraîche de Laravel.

Configuration de la base de données :

Copiez le fichier .env.example et renommez-le en .env. Configurez les paramètres de base de données dans ce fichier, notamment DB_CONNECTION, DB_HOST, DB_PORT, DB_DATABASE, DB_USERNAME, et DB_PASSWORD.

Génération de la clé d'application :

Exécutez la commande suivante pour générer la clé d'application Laravel :

```
php artisan key:generate
```

Migration de la base de données :

Exécutez les migrations pour créer les tables de base de données définies dans le dossier database/migrations :

```
php artisan migrate
```

Exécution du serveur de développement :

Utilisez la commande suivante pour démarrer le serveur de développement de Laravel :

```
php artisan serve
```

Cela lancera l'application sur <http://localhost:8000> par défaut.

Compilation des ressources front-end :

Si votre application utilise des ressources front-end telles que des fichiers CSS ou JavaScript, vous devrez les compiler en utilisant Laravel Mix. Exécutez la commande suivante pour compiler les ressources :

```
npm install
```

```
npm run dev
```

Ou pour une compilation en continu :

```
npm run watch
```

On peut espérer avoir un environnement de développement Laravel fonctionnel.