Développer en back-end

Réalisé par M. Hamid Belyaz

Développer en back-end

Introduction

A. Découvrir le Framework PHP Laravel

1. Découvrir les notions fondamentales des Frameworks PHP
2. Préparer l'environnement de Laravel

B. Programmer avec Laravel

1. Connaître les fondements du modèle MVC Laravel
2. Maîtriser le Framework Laravel

C. Approfondir la programmation Laravel

1. Gérer la sécurité
2. Interagir avec la base de données
3. Manipuler l'ORM Eloquent
4. Prendre en charge les tests

D. Administrer un site à l'aide d'un CMS

1. Manipuler les éléments essentiels d'un CMS
2. Personnaliser graphiquement un site à l'aide d'un CMS
3. Manipuler les outils avancés d'un CMS

Conclusion

Développer en back-end

Intoduction

1. Les prérequis
   - ✓ Langage PHP
   - ✓ POO(encapsulation ,surcharge ,héritage …)
   - ✓ PDO(PDO, PDOStatement, PDOException)
   - ✓ MVC(Model, View Controller)
   - ✓ Serveurs(service, Port ,Protocol …)
   - ✓ Base de Donnée(SQL)

Développer en back-end

A. Découvrir le Framework PHP Laravel

1. Découvrir les notions fondamentales des Frameworks PHP

- ✓ Framewoks PHP
- ✓ Intérêt du Framework back end Laravel
- ✓ Architecture 3-tiers MVC
- ✓ Présentation des API PHP

Développer en back-end

A. Découvrir le Framework PHP Laravel

2. Préparer l'environnement de Laravel

- ✓ Architecture du Framework Laravel
- ✓ Installation complète de Laravel (Composer, commandes PHP Artisan)
- ✓ Configuration de l'environnement Laravel
- ✓ Création d'un premier projet
- ✓ Lancement du serveur Laravel

A. Découvrir le Framework PHP Laravel

2. Préparer l'environnement de Laravel

Organisation de Laravel

le dossier créé par Composer, de nombreux dossiers et fichiers ont été ajoutés. Heureusement, il n'est pas nécessaire de connaître tous les dossiers et tous les fichiers pour commencer à travailler avec Laravel.

Les dossiers

✓ **app**

Le dossier app est le dossier le plus important de votre projet. C'est celui qui contiendra votre application, c'est à dire, tout votre code PHP (fonctions, classes…).

A. Découvrir le Framework PHP Laravel

2. Préparer l'environnement de Laravel

Organisation de Laravel

✓ **bootstrap**

Le dossier bootstrap n'est pas utile. Il contient principalement des fichiers liés au lancement du framework ainsi qu'un dossier cache pour certaines optimisations.

✓ **config**

Le dossier config permet la configuration du framework. À l'intérieur, chaque fichier correspond à une fonctionnalité configurable. Par exemple, le fichier config/database.php contient un tableau PHP avec différentes valeurs de configuration pour l'URL de la base de données, l'utilisateur, le mot de passe…

Développer en back-end

A. Découvrir le Framework PHP Laravel

2. Préparer l'environnement de Laravel

Organisation de Laravel

✓ **database**

Le dossier database permet la gestion de la base données.

Le principal sous-dossier est le sous-dossier migrations. Les migrations sont des fichiers permettant de décrire votre base de données afin de permettre à Laravel de créer, modifier ou supprimer les tables et les colones automatiquement pour vous. Si vous avez déjà utilisé PHPMyAdmin, les migrations remplacent une partie l'utilisation de PHPMyAdmin.

Les sous-dossiers seeds et factories ne sont pas utiles pour le moment.

Développer en back-end

A. Découvrir le Framework PHP Laravel

2. Préparer l'environnement de Laravel

Organisation de Laravel

✓ **public**

Le dossier public contient tous les fichiers accessibles directement par vos visiteurs.

Par exemple, si vous avez des images publiques sur votre site, elles doivent être dans le dossier public (ou dans un sous-dossier du dossier public). Même chose pour vos fichiers CSS et JavaScript.

Laravel fournit de base quelques fichiers utiles comme un favicon, un fichier robots.txt…

Le fichier index.php est la porte d'entrée de votre application. C'est le seul fichier PHP accessible de l'extérieur et il sera responsable de lancer le framework et d'appeler votre code situé dans le dossier app. Vous n'aurez jamais à modifier ce fichier directement car, comme dit précédemment, notre code PHP se trouve dans le dossier app.

Développer en back-end

A. Découvrir le Framework PHP Laravel

2. Préparer l'environnement de Laravel

Organisation de Laravel

✓ **resources**

Le dossier resources contient de nombreuses choses diverses. Principalement pour les autres fichiers de notre application qui ne sont pas du code PHP.

Le dossier resources/js et resources/sass contient des fichiers pré-CSS et pré-JS avant leur compilation. Si vous n'avez jamais compilé de fichier SASS, ni utilisé Babel, Webpack ou autre pour compiler votre JavaScript, passez votre chemin. Nous aurons tout le temps d'étudier ces concepts par la suite.

Le dossier resources/lang contient les fichiers de traduction pour votre application. Ils ne vous seront utiles que si vous souhaitez créer un site multi-lingue.

Le dossier resources/views contient les vues de votre application. Les vues sont des fichiers majoritairement composés de HTML et sont chargés de la partie affichage de votre site. C'est l'un des dossiers les plus importants après le dossier app.

Développer en back-end

A. Découvrir le Framework PHP Laravel

2. Préparer l'environnement de Laravel

Organisation de Laravel

✓ **routes**

Si vous avez l'habitude d'une architecture PHP simple : https://mon-site.com/contact.php exécute le code situé dans contact.php et https://mon-site.com/compte.php exécute compte.php. Dans une architecture MVC comme celle de Laravel, toutes les requêtes des utilisateurs arrivent via le fichier public/index.php il est donc nécessaire de faire le lien entre l'URL entrée par le visiteur (« /contact », « /mon-compte »…) et le code qui sera exécuté. C'est dans le fichier routes/web.php que vous allez configurer ce lien.

Si vous souhaitez développer une API, le fichier routes/api.php sera l'endroit où mettre vos liens. Si vous souhaitez mettre en place des actions en ligne de commande pour votre application, ce sera le fichier routes/console.php. Et si vous souhaitez envoyer des messages avec des websockets à vos visiteurs, ce sera le fichier routes/channel.php. Mais ces trois fichiers ne sont pas indispensable contrairement au fichier routes/web.php.

Développer en back-end

A. Découvrir le Framework PHP Laravel

2. Préparer l'environnement de Laravel

Organisation de Laravel

✓ **storage**

Le dossier storage/app contient tous les fichiers générés par votre application, par exemple des factures PDF, les photos de profil de vos utilisateurs, etc.

Le dossier storage/framework contient des fichiers utilisés uniquement par le framework. Il est recommandé de ne pas ajouter ou supprimer de fichiers à ce dossier.

Enfin, le dossier storage/logs contient les fichiers de logs de votre application. Les fichiers de logs contiennent des informations sur l'activité de votre application. Par défaut, Laravel enregistrera dans un fichier storage/logs/laravel-YYYY-MM-DD.log tous les problèmes rencontrés par votre application : très utile pour comprendre pourquoi votre site ne fonctionne pas par exemple.

A. Découvrir le Framework PHP Laravel

2. Préparer l'environnement de Laravel

Organisation de Laravel

✓ **tests**

Les tests sont un moyen rapide et automatisé de vérifier que votre application fonctionne comme vous le souhaitez. Si votre application commence à grossir, il est important de comprendre comment fonctionne les tests en programmation.

✓ **vendor**

Le dossier vendor contient toutes les dépendances PHP téléchargées par Composer. Vous pouvez par exemple retrouver dans vendor/laravel/framework le code source de Laravel. Vous ne devez jamais changer les fichiers de ce dossier, car ces modifications seront écrasées par Composer à la prochaine mise à jour.

Développer en back-end

A. Découvrir le Framework PHP Laravel

2. Préparer l'environnement de Laravel

Organisation de Laravel

Les fichiers

À la racine de notre projet, Composer a également ajouté de nombreux fichiers. La plupart sont des fichiers de configuration pour des outils externes (Git, Composer, NPM, PHPUnit…) et nous ne les utiliserons pas avant d'avoir découvert ces outils.

✓ **.env**

Les fichiers .env contient les mots de passe de vos services ainsi que toutes les données sensibles de votre application (mot de passe de base de données, adresse de la base de données…). Ce fichier ne doit jamais être partagé. Afin de connaître les informations à renseigner, il existe un fichier .env.example qui contient uniquement des valeurs d'exemple.

✓ **.gitattributes et .gitignore**

Ces fichiers sont utilisés par le logiciel Git. Si vous n'utilisez pas Git, vous n'avez pas à vous en occuper.

✓ **artisan**

Le fichier artisan permet de lancer des commandes comme php artisan serve. Ces commandes vont nous permettre de faire beaucoup de choses avec Laravel .

A. Découvrir le Framework PHP Laravel

2. Préparer l'environnement de Laravel

Organisation de Laravel

✓composer.json et composer.lock

Le fichier composer.json contient toutes les dépendances requises par notre application. Le fichier composer.lock est un fichier généré automatiquement par Composer lors de la commande composer update. Vous ne devez jamais le modifier manuellement.

✓webpack.mix.js

Webpack est un outil permettant de transformer des fichiers SASS en fichier CSS ou encore de compiler du JavaScript.

A. Découvrir le Framework PHP Laravel

2. Préparer l'environnement de Laravel

Organisation de Laravel

✓ packages.json et yarn.lock

Le fichier packages.json est identique au fichier composer.json en PHP mais pour le JavaScript avec l'outil NPM. Nous ne l'utiliserons pas pour le moment.

✓ phpunit.xml

PHPUnit est un outil qui permet de lancer les tests unitaires et fonctionnels. Ce fichier sera utile lorsque vous utilisez des tests dans le dossire tests.

✓ server.php

Le fichier server.php est uniquement présent pour que la commande php artisan serve fonctionne. Vous ne devriez jamais avoir à y toucher.