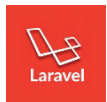


Création d'un gestionnaire de membres de club sportif

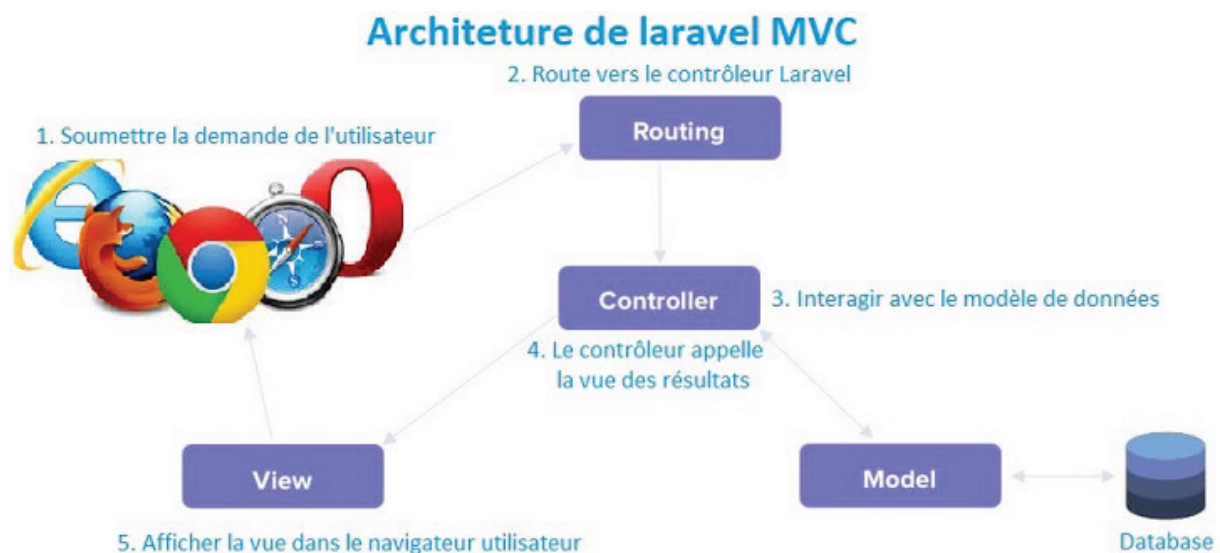
Introduction



Laravel est un framework web open-source écrit en PHP respectant le principe modèle-vue-contrôleur (MVC) et entièrement développé en programmation orientée objet.

Laravel crée une application web en réutilisant les composants existants de différents frameworks.

Ainsi, l'application web conçue est plus organisée et plus pratique, possède **un large éventail de fonctionnalités** qui aident à accélérer la vitesse de développement Web.



En résumé, lorsqu'un client envoie une requête à l'application, celle-ci est analysée par le contrôleur, qui demande au modèle approprié d'effectuer les traitements, puis renvoie la vue adaptée au navigateur, si le modèle ne l'a pas déjà fait.

L'avantage d'utiliser Laravel comme framework est qu'il offre **une sécurité de haut niveau**. Cela signifie que votre code de développement Web est plus sûr et sécurisé.

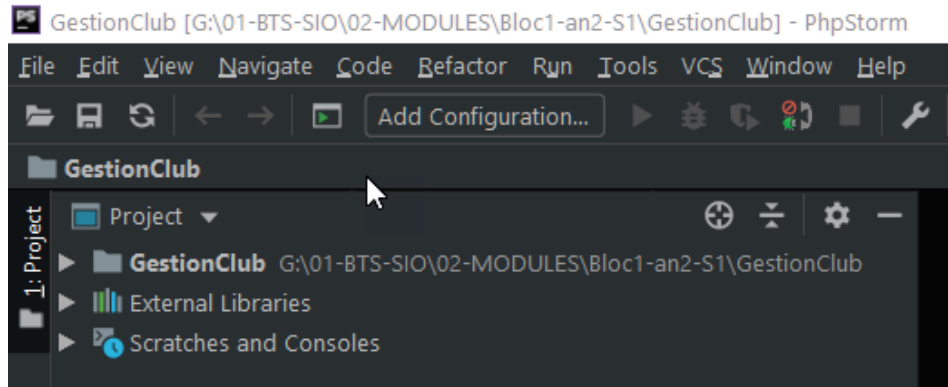
Une autre raison qui fait du framework Laravel un choix privilégié pour les développeurs pour votre développement Web est qu'il facilite les **tests unitaires**. Cela garantit que les nouvelles modifications apportées par le développeur n'interrompent pas l'application Web de manière inattendue.

Le **test unitaire** est un type de test dans lequel chaque module ou composant de votre application Web est testé afin qu'aucune partie de votre site Web ne reste cassée.

Il garantit qu'il n'y a pas de bogues ou d'exceptions dans votre application Web avant qu'elle n'entre dans l'environnement en direct, de sorte que vous ayez des applications hautement performantes et sans bogues pour vos utilisateurs finaux (nous reverrons ce point plus en détail ultérieurement).

Le projet

Créez un nouveau projet PHPstorm



Maintenant, nous devons installer **Laravel** pour notre projet.

Ouvrez « Terminal » (alt+F12), il devrait aller dans le répertoire racine du projet.

Exécutez les commandes suivantes :

Téléchargez l'installateur Laravel →

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

Cette commande installera automatiquement la dernière version de l'installateur Laravel sur votre machine.

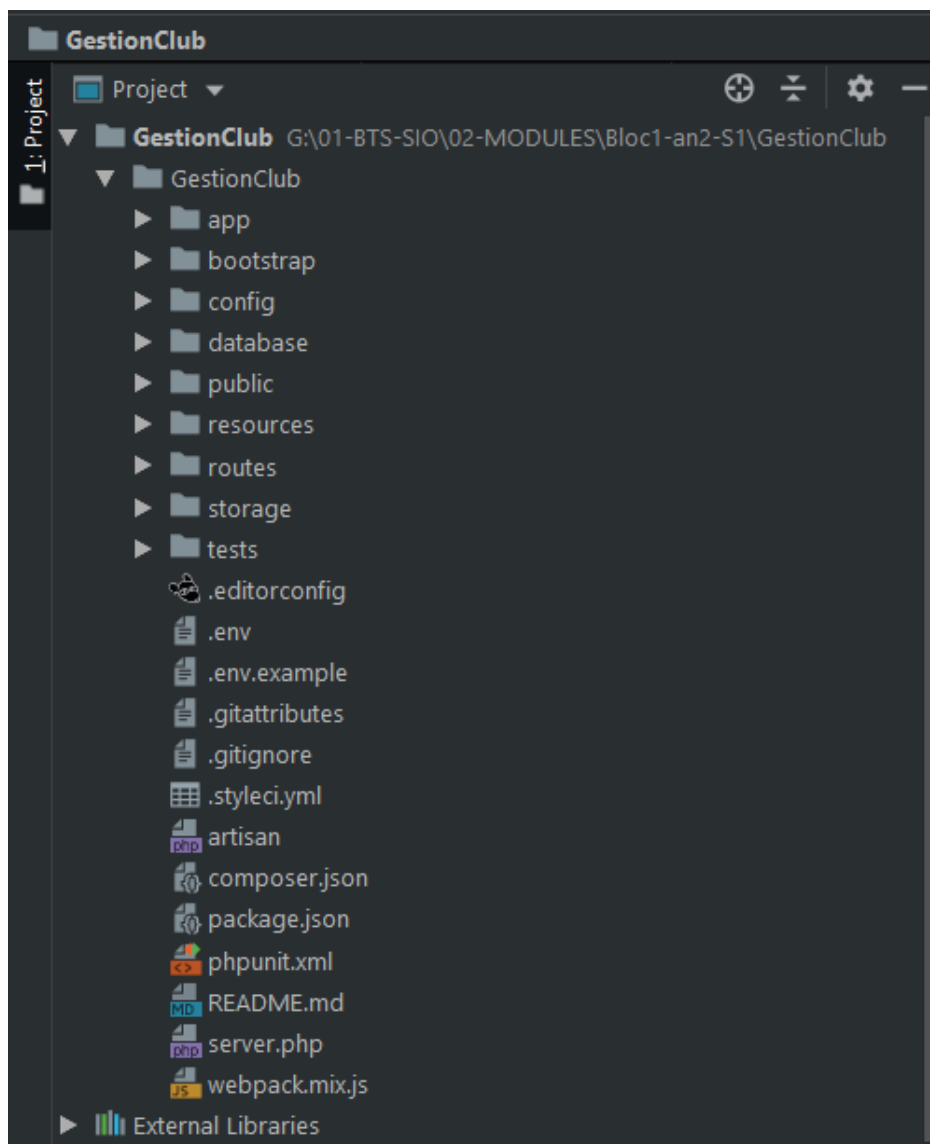


Ensuite, nous pouvons créer une nouvelle application Laravel dans le répertoire du projet :

```
laravel new GestionClub
```

⚠ Lisez bien les messages dans le terminal et au besoin, activez les extensions demandées.

👁 Une fois que c'est fait, voici ce que vous devriez obtenir :



Vérifiez que composer est bien installé pour le projet (Tools/Composer/)

Répertoires à la racine

app

Ce répertoire contient l'intégralité du code source de notre projet. Il comprend les événements, les commandes, les exceptions, etc... Mais, pour l'instant, nous ne nous intéressons qu'aux modèles et aux contrôleurs.

config

Comme son nom l'indique, il stocke tous les fichiers de configuration de notre projet.

database

Le site `database` est l'endroit où nous mettons tous les fichiers de semences et de migration. Ils déterminent la structure de la base de données.

public

Ce répertoire contient le fichier `index.php`, qui est le point d'entrée de toutes les requêtes. Nous devons également placer tous les fichiers statiques (CSS et JS) dans ce répertoire qui seront générés à partir des fichiers CSS et JS du répertoire `resources`.

routes

Le site `route` contient toutes les déclarations d'URL pour notre projet. Par défaut, il y a quatre fichiers `route` : `web.php`, `api.php`, `console.php` et `channels.php`. Pour ce tutoriel, nous avons seulement besoin de comprendre `web.php`.

resources

Ce dossier stocke toutes les vues et les fichiers non compilés tels que LESS, SASS ou JavaScript.

Il y a également un dossier pour **bootstrap** si jamais on veut l'utiliser pour le site internet

Répertoires du dossier app

Http/Controllers

C'est ici que nous plaçons tous les contrôleurs de notre projet. Toute la logique permettant de traiter les demandes entrant dans votre application sera placée dans ce répertoire. Pour rappel, les contrôleurs sont les intermédiaires entre nos vues et nos modèles.

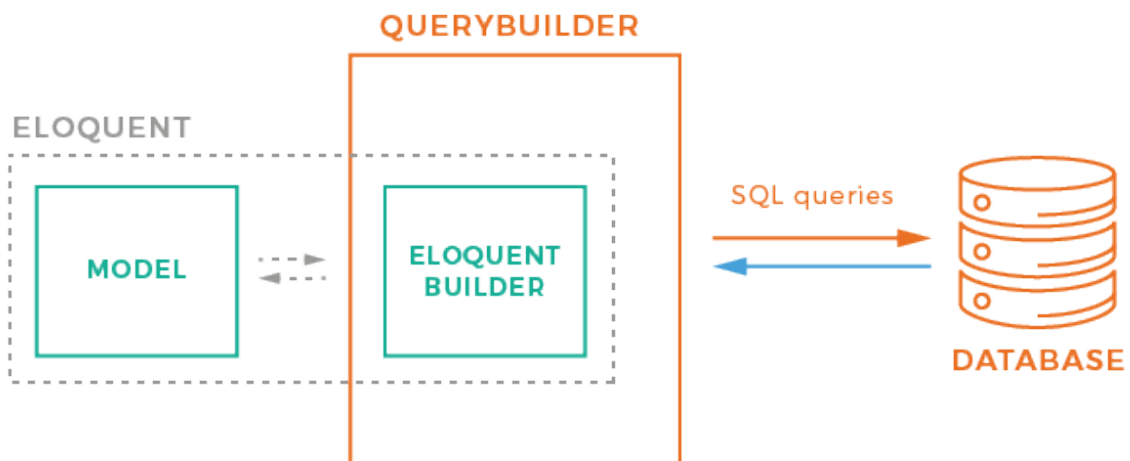
Models

Laravel 9 possède désormais un dossier individuel pour les modèles. Le dossier `Models` contient toutes vos classes de modèles Eloquent.

L'ORM Eloquent¹ inclus avec Laravel fournit une belle et simple implémentation ActiveRecord pour travailler avec votre base de données.

Chaque table de votre base de données a un "modèle" correspondant qui est utilisé pour interagir avec cette table.

Les modèles vous permettent d'interroger les données de vos tables, ainsi que d'insérer de nouveaux enregistrements dans la table.



¹ C'est quoi un ORM en informatique ?

ORM signifie Object-Relational Mapping. Un ORM est un ensemble de classes permettant de manipuler les tables d'une base de données relationnelle comme s'il s'agissait d'objets.

L'une des caractéristiques les plus importantes du framework laravel est qu'il est livré avec ORM (Object Relation Mapping) intégré appelé Eloquent ORM.

Configuration du projet

Toutes les configurations d'environnement dans Laravel sont stockées dans le fichier `.env` dans le répertoire racine. Pour que notre projet fonctionne correctement, il y a quelques changements/verifications que nous devons faire.

URL de l'application

Tout d'abord, définissons l'URL de notre projet. Puisque nous sommes en développement, l'URL devrait être `http://localhost`.

```
APP_URL=http://localhost
```

Base de données

Si vous avez installé un logiciel proposant la gestion de vos bases de données comme WAMP, MAMP, UWamp, EasyPHP...

Pour l'exemple, je vais utiliser MySQL et me rendre sur PhpMyAdmin afin de créer ma base de données « Club ».



Clé d'application

La définition d'une clé d'application est essentielle à la sécurité de notre projet. Si la clé d'application n'est pas définie, vos sessions utilisateur et autres données cryptées ne seront pas sécurisées.

La commande suivante définira la clé d'application pour vous :

```
php artisan key:generate
```

```
G:\01-BTS-SIO\02-MODULES\Bloc1-an2-S1\GestionClub\GestionClub>php artisan key:generate
Application key set successfully.
```

Stockage de fichier

Cette configuration ne concerne pas l'environnement, mais elle est tout de même très importante. Afin de s'assurer que Laravel trouvera les fichiers média (images, vidéos...) que nous avons téléchargés, nous devons créer un lien symbolique dans le répertoire `public` avec cette commande :

```
php artisan storage:link
```

Démarrer le serveur

Pour commencer, rendez-vous à la racine du dossier laravel que nous avons nommé « GestionClub »

```
cd GestionClub
```

Enfin, il est temps de démarrer notre serveur. Nous avons deux choix, nous pouvons :

🔗 exécuter le serveur de développement local intégré :

```
php artisan serve
```

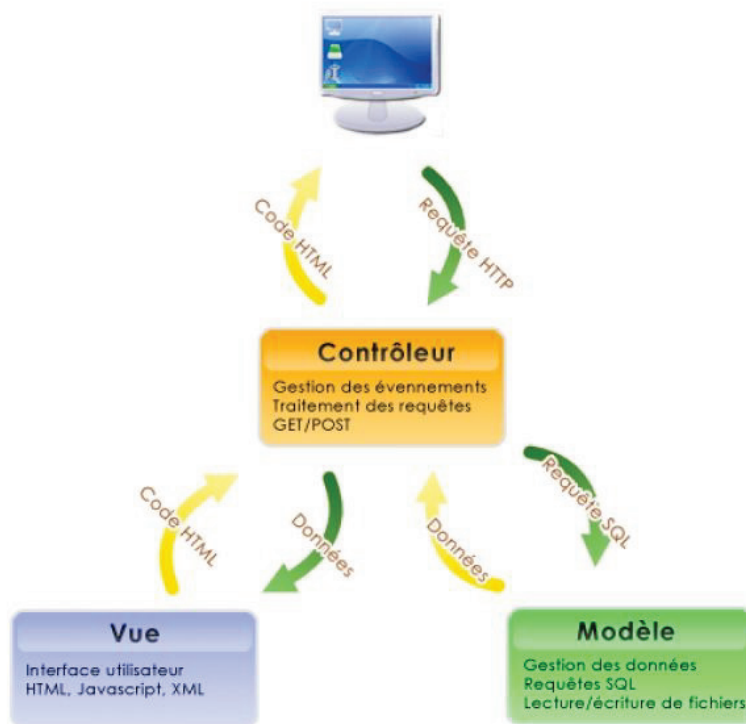
La commande `php artisan serve` lance un serveur web qui écoute sur le port 8000 par défaut. Il va permettre de tester le résultat de votre développement.

Vous pouvez tester que l'initialisation de votre projet est correcte en tapant l'URL `http://localhost:8000` dans votre navigateur.

🔗 Ou vous pouvez utiliser votre logiciel de développement proposant un serveur Apache ou Nginx (WAMP, MAMP, Docker, Valet...).

Ressources

Le modèle MVC en une image



Et une critique approfondie de ce modèle que j'ai tendance à partager mais qui est un peu complexe à comprendre :

<https://iparnet.fr/le-modele-mvc-nexiste-pas/>

Sur la toile

<https://walkerspider.com/cours/laravel/presentation-installation/>