
E6

Laravel

Contexte

Dans le cadre d'un de mes stages professionnels, j'ai été chargé de la réalisation de tests unitaires et fonctionnels pour la société Delta CE tout en étant encadré par l'équipe de développeur présente pour répondre mes questions et me guider sur cette tâche.

La société souhaitait disposer de tests PHP pour faciliter le développement de nouvelles fonctionnalités sur son application de gestion de communauté d'entreprise afin de ne plus avoir besoin de tester manuellement les fonctionnalités après l'ajout d'une nouvelle ou la migration vers une nouvelle version de PHP.

Avant de pouvoir réaliser ces tests il m'a d'abord fallu installer le Framework Laravel.

Description de l'entreprise

Delta CE est une entreprise créée en 2012 dont l'activité est de développer et proposer son propre logiciel de gestion de communauté d'entreprise, l'entreprise propose donc à d'autres professionnels d'utiliser l'application pour gérer leur communauté d'entreprises moyennant un abonnement payant.

Le service propose les fonctionnalités suivantes :

- Système de chat textuel
- Echange de fichiers
- Conversations vocales
- Création de groupes et comité
- Administration facile d'un groupe ou comité
- Formule d'abonnement avec plus ou moins d'avantages selon le prix et les besoins
- Allocations et gestion faciles de planning et temps dédié au comité

En termes de clients, l'entreprise dispose de plus de 300 clients CSE à leur actif et plus de 250 000 utilisateurs à l'aube de leur 7 ans de maîtrise dans ce domaine.

Expression des besoins

L'entreprise Delta CE souhaite mettre en place un système de test sur leur application de gestion de comité d'entreprise afin de faciliter le déploiement de nouvelles fonctionnalités et de mise à jour, pour se faire j'ai dû utiliser le système de base de Laravel (PHPUnit) pour effectuer et développer ces tests mais il m'a d'abord fallu installer Laravel

Intérêt du projet

L'intérêt d'utiliser Laravel pour ce projet et que ce Framework est très évolutif et très bien adapté à tout type de développement en plus de disposer d'une communauté très active autour de celui-ci.

Il est également fourni avec tout type de librairies divers et variées permettant la réalisation de nombreuses fonctionnalités simplement sans avoir besoin de télécharger plusieurs librairies et de les installer.

Avantage de Laravel

L'utilisation de Laravel dispose de nombreux avantages, en voici quelques-uns principaux pour le projet réalisé :

- Un Framework très évolutif
- Très complet
- Intégration de Symphony
- Intégration de nombreuses bibliothèques
- Une très grande communauté active
- Adapté aux projets de toutes tailles

Inconvénients de Laravel

L'utilisation de Laravel présente néanmoins quelques inconvénients :

- Nécessite des connaissances de base dans Symphony
- Possibilité de se perdre dans toutes les fonctionnalités
- Installation pourrait peut-être être plus simple
- Nécessite Docker

Réalisation et mise en pratique

Téléchargement et installation :

Pour commencer nous avons besoin d'installer docker ou docker desktop sur notre machine, qu'elle fonctionne sous Linux, Windows ou MacOS. Une fois installé, il faut entrer une ligne de commande en remplaçant « example-app » par notre nom de notre projet désiré, le projet sera ainsi créé dans le dossier depuis lequel on utilise la commande.

```
curl -s https://laravel.build/example-app | bash
```

Une fois le dossier configuré il faut lancer la ligne de commande suivante pour installer Laravel dans notre dossier de projet :

```
cd example-app  
  
./vendor/bin/sail up
```

Le site devrait ensuite être accessible depuis le localhost à l'adresse <http://localhost>.

Il est également possible d'installer d'abord PHP ainsi que Composer pour démarrer une installation depuis composer, pour se faire il faut entrer les commandes suivantes :

```
composer create-project laravel/laravel example-app  
  
cd example-app  
  
php artisan serve
```

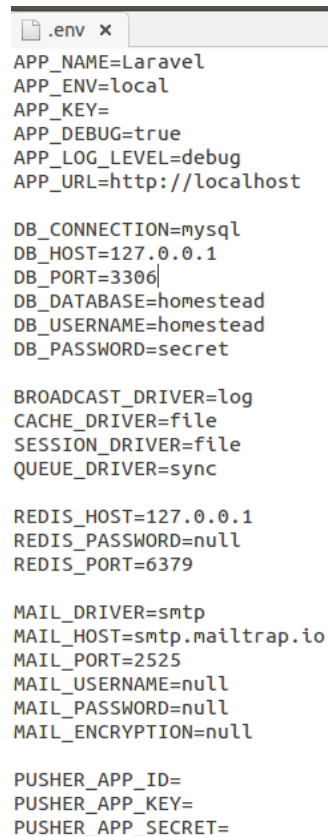
Il est également de créer un git directement lié au projet que l'on crée par le biais d'une ligne de commande également, nous pouvons aussi définir une branche sur laquelle nous souhaitons mettre l'installation de Laravel.

```
laravel new example-app --git --branch="main"
```

Configuration :

Toutes les informations de configuration de Laravel sont disponibles dans le fichier « config » situé dans le dossier du projet, ce fichier contient toutes les informations importantes tel que les informations de connexion a la base de données, la configuration du serveur de mail, mais aussi la configuration de la time zone pour notre application ainsi que les clé d'encryptions.

Il est également possible de définir des variables d'environnement qui seront utilisé dans les cas souhaités celons l'environnement de développement faisant tourner Laravel, toute ces variables d'environnement sont situées dans un fichier nommé .env

A screenshot of a code editor showing a file named .env. The file contains various configuration variables for a Laravel application, including database settings, cache, session, queue, Redis, mail, and Pusher settings. The variables are organized into groups with blank lines between them.

```
.env x
APP_NAME=Laravel
APP_ENV=local
APP_KEY=
APP_DEBUG=true
APP_LOG_LEVEL=debug
APP_URL=http://localhost

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=homestead
DB_USERNAME=homestead
DB_PASSWORD=secret

BROADCAST_DRIVER=log
CACHE_DRIVER=file
SESSION_DRIVER=file
QUEUE_DRIVER=sync

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_DRIVER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null

PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
```

Toutes les variables présentes dans ce fichier seront stockées dans une variable superglobale nommé « \$_ENV » lorsque l'application va recevoir une requête. Ces valeurs peuvent être récupérer grâce à une fonction « env() » disponible avec Laravel, cette fonction a besoin de 2 paramètres, l'un qui est le nom de la variable à récupérer dans le fichier .env, et l'autre est une valeur par défaut à retourner si la variable définit en premier paramètre n'existe pas.

Déploiement :

Pour assurer le bon déploiement d'une application Laravel, quelques prérequis sont nécessaires pour un bon fonctionnement et éviter au maximum les erreurs, les prérequis sont les suivants :

- PHP >= 7.3
- BCMath PHP Extension
- CType PHP Extension
- Fileinfo PHP Extension
- JSON PHP Extension
- Mbstring PHP Extension
- OpenSSL PHP Extension
- PDO PHP Extension
- Tokenizer PHP Extension
- XML PHP Extension

Il faut ensuite utiliser un serveur tournant sous Nginx et s'assurer que toutes les requêtes arrivant au serveur soient bien redirigées vers l'application sur le fichier « public/index.php », puis configurer le serveur avec un fichier de configuration comme celui-ci :

```
server {  
    listen 80;  
    server_name example.com;  
    root /srv/example.com/public;  
  
    add_header X-Frame-Options "SAMEORIGIN";  
    add_header X-Content-Type-Options "nosniff";  
  
    index index.php;  
  
    charset utf-8;  
  
    location / {  
        try_files $uri $uri/ /index.php?$query_string;  
    }  
  
    location = /favicon.ico { access_log off; log_not_found off; }  
    location = /robots.txt  { access_log off; log_not_found off; }  
  
    error_page 404 /index.php;  
  
    location ~ \.php$ {  
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;  
        fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;  
        include fastcgi_params;  
    }  
  
    location ~ /\.(!well-known).* {  
        deny all;  
    }  
}
```

Il est bien évidemment possible de modifier ce fichier de configuration et peut être personnalisé en fonction de nos besoin pour notre application.