

# Podz

---

## Application de publication de podcasts

Projet de TPI de Samuel Roland

## Compétences requises

Pour modifier ce projet vous avez besoin de savoir développer avec Laravel, Livewire, AlpineJS et TailwindCSS. Laravel et Livewire sont les plus importants à maîtriser pour faire des modifications. AlpineJS et TailwindCC peuvent être appris au fur et à mesure. Les documentations de ces 4 frameworks sont bien écrites et ils existent plusieurs formations et des tutoriels sur le sujet pour les prendre en main.

## Outils requis

Vous avez besoin de:

- PHP v8+
- MySQL v8+
- Git v2+
- Un IDE pour PHP: VSCode avec des extensions spécifiques, PHPStorm, autre...
- NodeJS v17+ (incluant NPM v8+)
- Composer v2+

## Mise en place de l'environnement

Configurer votre fichier `php.ini` pour activer les extensions suivantes:

- PDO SQLITE
- PDO MySQL
- Openssl
- Curl
- GD
- mbstring
- fileinfo

Ces 2 paramètres dans la configuration de PHP (fichier `php.ini`) doivent être augmentées au-dessus de 150MB: `upload_max_filesize` et `post_max_size` afin de permettre l'upload de fichiers.

# Mise en place du projet

1. Ouvrir un terminal de type Cmder (avec commandes Unix supportées), cloner le repository et aller dans le dossier:

```
git clone https://github.com/samuelroland/podz.git  
cd podz
```

2. Installer les paquets Composer et NPM

```
composer install  
npm install
```

3. Copier le fichier d'environnement et générer la clé d'application

```
cp .env.example .env  
php artisan key:generate
```

Créer une base de données nommée `podz` sur votre serveur MySQL, et remplissez les valeurs de `DB_USERNAME` et le `DB_PASSWORD` dans le fichier `.env`. Si la base de données a un autre nom, il suffit de changer `DB_DATABASE`.

4. Créer le lien symbolique pour l'accès au stockage local

```
php artisan storage:link
```

5. Créer les tables dans la base de données

```
php artisan migrate
```

6. Si vous voulez développer Podz, vous pouvez remplir la base de données avec des données aléatoires de tests

```
php artisan db:seed
```

7. Plus tard, si vous souhaitez recréer votre base de données et y ajouter des données de tests

```
php artisan migrate:fresh --seed
```

## Lancer le serveur de développement

1. Vous avez besoin de compiler les fichiers JS et CSS avec Laravel Mix en mode watch (rafraîchissement à chaque modification):

```
npm run watch
```

Le résultat de ce processus sont les fichiers `public/css/app.css` et `public/js/app.js`.

2. Si vous n'allez pas travailler dessus mais que vous avez juste besoin de voir l'application, vous pouvez lancer cette autre commande qui fait le processus 1 seule fois.

```
npm run dev
```

3. Et dans un autre terminal, vous pouvez démarrer votre serveur:

```
php artisan serve
```

4. Il ne reste plus qu'à ouvrir votre navigateur sur `localhost:8000` ou l'autre adresse affichée et vous devriez pouvoir vous connecter avec les identifiants suivants: `sam@example.com` - `password` ou vous pouvez aussi créer un nouveau compte.

## Préparation et mise en production

Une fois l'environnement et le projet mis en place (sans le remplissage des données de tests), il faut compiler et optimiser les fichiers JS et CSS pour la production afin d'avoir des fichiers légers (minification, purgation, ...).

1. Compiler pour la production

```
npm run prod
```

2. Définir le `APP_DEBUG` à `false` et regarder les autres informations de déploiement sur [la documentation Laravel](#)
3. Vous pouvez maintenant démarrer un serveur Apache ou Nginx dans le dossier `public`, puis visiter l'URL ou l'IP définie pour ce serveur. A cette étape, Podz est bien déployé si vous arrivez à l'utiliser dans votre navigateur.