

# Documentation Laravel

---

[Retour à toute les documentations](#)

## Règles

- "Saisie utilisateur"
- 'Élément cliquable/sélectionnable'
- Nom de fichier, dossier ou autre
- <Élément à remplacer>

lien, raccourci clavier et phrase de demande de saisie

commande, extrait code et extrait de fichier

## Table des matières

---

- [Documentation Laravel](#)
  - [Règles](#)
  - [Table des matières](#)
  - [Installation de Laravel](#)
    - [Installation de Laravel CLI](#)
  - [Configuration de Laravel](#)
    - [Intégration de plusieurs base de données](#)
    - [Intégration de Tailwind CSS](#)
    - [Intégration de Livewire](#)
  - [Utilisation de Laravel](#)
    - [Création d'un projet Laravel](#)
    - [Gestion de la base de données](#)
    - [Lancement d'un projet Laravel en local grâce au serveur web Apache](#)
    - [Lancement d'un projet Laravel en local grâce au serveur de développement de Laravel](#)
    - [Rendre le serveur de développement de laravel accessible sur tout les appareils d'un réseau local](#)
  - [Livewire](#)
    - [Ajout de Livewire à un projet Laravel](#)
    - [Création d'un composant Livewire](#)

## Installation de Laravel

- Documentation officiel, complète et très bien expliqué

<https://laravel.com/docs/10.x>

- Source

<https://www.webhi.com/how-to/how-to-install-laravel-on-ubuntu-debian-apache-nginx/>

- Installer les différentes dépendances PHP dont Laravel à besoin

```
sudo apt install php php-cli php-common php-mbstring php-xml php-zip php-mysql php-pgsql php-sqlite3 php-json php-bcmath php-gd php-tokenizer php-xmlwriter
```

- [Installer Composer](#)
- [Installer le serveur web Apache pour php](#)

## Installation de Laravel CLI

- Installation de Laravel CLI grâce à [Composer](#)

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

## Configuration de Laravel

- Configuration des permissions dans votre projet Laravel (normalement vous n'avez pas besoin de faire cette étape)

```
sudo chown -R www-data:www-data /path/to/your-project-name  
sudo chmod -R 755 /path/to/your-project-name
```

## Intégration de plusieurs base de données

- Source

<https://arjunamrutiya.medium.com/laravel-multiple-database-connectivity-a-step-by-step-guide-72cecb5d9223>

## Intégration de Tailwind CSS

- Installation de Tailwind CSS

```
npm install -D tailwindcss postcss autoprefixer
```

- Création du fichier de configuration de Tailwind CSS

```
npx tailwindcss init -p
```

- Ajoutez le code suivant dans le fichier `tailwind.config.js`

```
/** @type {import('tailwindcss').Config} */  
export default {  
  content: [  
    "./resources/**/*.blade.php",  
    "./resources/**/*.js",  
    "./resources/**/*.vue",  
  ],  
}
```

```
],  
  theme: {  
    extend: {},  
  },  
  plugins: [],  
}
```

- Ajoutez le code suivant dans le fichier `resources/css/app.css`

```
@tailwind base;  
@tailwind components;  
@tailwind utilities;  
@tailwind forms;
```

- Vous pouvez compiler les fichiers CSS de Tailwind CSS en utilisant la commande suivante
  - Vous pouvez voir ça comme l'activation du style Tailwind CSS. Il faudra refaire cette commande à chaque fois que vous ajouter un nouveau style dans votre projet

```
npm run build
```

## Intégration de Livewire

- [Ajouter et utiliser Livewire à un projet Laravel](#)

## Utilisation de Laravel

### Création d'un projet Laravel

- Créer un projet Laravel grâce à `Composer`

```
composer create-project --prefer-dist laravel/laravel your-project-name
```

- Créer un projet Laravel grâce à la commande `laravel`

```
laravel new your-project-name
```

- Créer un projet Laravel avec une version spécifique

```
composer create-project --prefer-dist laravel/laravel your-project-name "8.*"
```

### Gestion de la base de données

- Créer une migration

```
php artisan make:migration create_<table-name>_table
```

- Créer un modèle

```
php artisan make:model <ModelName>
```

- Créer un modèle et une migration

```
php artisan make:model <ModelName> -m
```

- Modifier le fichier de Model pour ajouter le nom de la table

```
protected $fillable = [  
    'name',  
    'email',  
];
```

- Exécuter les migrations

```
php artisan migrate
```

- Créer un contrôleur

```
php artisan make:controller <ControllerName>
```

## Lancement d'un projet Laravel en local grâce au serveur web Apache

- Si vous n'avez pas créer le projet Laravel dans le dossier `/var/www/html` il faut créer un lien symbolique entre le dossier du projet Laravel et le dossier `/var/www/html`

```
sudo ln -s /path/to/your-project-name /var/www/html/your-project-name
```

- Créer un fichier de configuration pour le projet Laravel

```
sudo nano /etc/apache2/sites-available/your-project-name.conf
```

- Ajouter le code suivant dans le fichier de configuration

```
<VirtualHost *:80>  
    ServerName your-domain-or-ip  
    DocumentRoot /var/www/html/your-project-name/public  
    <Directory /var/www/html/your-project-name>  
        AllowOverride All  
    </Directory>  
</VirtualHost>
```

- Activez le module de réécriture Apache :

```
sudo a2enmod rewrite
```

- Activez l'hôte virtuel :

```
sudo a2ensite your-project-name.conf
```

- Redémarrez Apache pour que les modifications prennent effet :

```
sudo systemctl restart apache2
```

## Lancement d'un projet Laravel en local grâce au serveur de développement de Laravel

- Lancez le serveur web de développement de Laravel

```
php artisan serve
```

- Si vous avez cette erreur `The stream or file "/path/to/your-project-name/storage/logs/laravel.log" could not be opened: failed to open stream: Permission denied` il faut modifier le créateur du dossier `storage` et de son contenu

- Source de la solution

```
https://stackoverflow.com/questions/30639174/laravel-5-ubuntu-14-04-permission-denied-on-storage-log
```

```
sudo chown -R ${USER}:www-data /path/to/your-project-name/storage
```

- Redonnez les droits au dossier `storage` si nécessaire

```
sudo chmod -R 775 /path/to/your-project-name/storage
```

- Relancez le serveur web de développement de Laravel

```
php artisan serve
```

## Rendre le serveur de développement de laravel accessible sur tout les appareils d'un réseau local

- Lancer le serveur laravel avec la commande suivante

```
php artisan serve --host=0.0.0.0 --port=8000
```

- Récupérer l'adresse IP ou le nom de la machine sur lequel le serveur laravel est lancé

- Récupérer l'adresse IP sous Linux

```
hostname -I
```

- Récupérer le nom de la machine sous Linux

```
hostname
```

- Récupérer l'adresse IP sous Windows

```
ipconfig
```

- Récupérer l'adresse IP sous Windows grâce à l'interface graphique

- <https://support.microsoft.com/en-us/windows/find-your-ip-address-in-windows-f21a9bbc-c582-55cd-35e0-73431160a1b9>

- Ouvrir un navigateur sur un autre appareil connecté au même réseau

- Entrer l'adresse IP de la machine sur lequel le serveur laravel est lancé suivi du numéro de port, en l'occurrence ":8000"

```
<adresse_ip>:8000
```

OU

```
<nom>:8000
```

- En règle général l'adresse IP est de la forme 192.168.1.XX

```
192.168.1.XX:8000
```

## Livewire

### Ajout de Livewire à un projet Laravel

- Installer LiveWire grâce à **Composer**

```
composer require livewire/livewire
```

### Création d'un composant Livewire

- Exécuter la commande suivante dans le dossier racine de votre projet Laravel pour créer un composant Livewire

```
php artisan make:livewire <NomDuComposant>
```

---

[Retour à toute les documentations](#)