

VueJS dans Laravel avec Vite

Configurer Laravel avec Vite et Vue.js permet de bénéficier des performances de Vite pour le développement rapide et le chargement des modules ainsi que de la puissance de Vue.js pour construire des interfaces utilisateur dynamiques. Voici les étapes pour configurer Laravel avec Vite et Vue.js :



Prérequis



1. Node.js (version 14 ou supérieure)
2. Composer
3. Laravel (version 8 ou supérieure)

Étapes de Configuration

```
vite.config.js
1 import { defineConfig } from 'vite'
2 import laravel from 'laravel-vite-plugin'
3 import vue from '@vitejs/plugin-vue'
4
```

Laravel + Vue + Vite

1. Créer un Nouveau Projet Laravel

Si vous n'avez pas encore un projet Laravel, créez-en un nouveau :

```
composer create-project laravel/laravel mon-projet  
cd mon-projet
```

2. Configurer l'environnement NodeJS

Avant toute chose, nous aurons besoin de nous assurer d'avoir la version la plus récente de **nodejs** (lien symbolique “node”). Pour ce faire, dans l'environnement Linux, nous éviterons d'utiliser le traditionnel “sudo apt install node” car il installe une vieille version du logiciel.

En conséquence, nous utiliserons la méthode d'installation suivante:

```
curl https://raw.githubusercontent.com/creationix/nvm/master/install.sh | bash  
nvm install --lts
```

Nous avons utilisé le logiciel “**nvm**” (node version manager) afin d'accéder aux versions les plus récentes. La commande:

```
nvm list-remote
```

nous permet de les lister et le flag ci-dessus “**--lts**” (latest) a pour but d'installer la toute dernière version.

Cette installation nous permet donc d'avoir **node** (lien symbolique de **nodejs**):

```
node -v
```

Mais aussi “**npm**” (node package manager), l'équivalent de “composer” en Javascript.

```
npm -v
```

Ce logiciel représente aussi l'équivalent de “php artisan serve” ainsi que le montrera la commande “**npm run dev**” ci-dessous.

Installer et lancer VueJS de cette manière dans Laravel:

```
composer require laravel/ui  
  
php artisan ui vue  
  
sudo npm install  
cd mon-projet
```

3. Si nécessaire, installer Vite

Installez les dépendances nécessaires pour utiliser Vite si vous rencontrez des difficultés lors de la procédure effectuée ci-dessus:

```
npm install --save-dev vite laravel-vite-plugin
```

Ajoutez Vue.js et ses dépendances :

```
npm install vue@next @vitejs/plugin-vue
```

4. Configurer la Vue de Laravel

Mettez à jour votre vue Blade (par exemple, `resources/views/welcome.blade.php`) pour inclure l'élément Vue :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
        content="width=device-width, initial-scale=1.0">
    <title>Laravel with Vite and Vue</title>
    @vite('resources/js/app.js')
</head>
<body>
    <div id="app"></div>
</body>
</html>
```

5. Lancer le Serveur de Développement

Exécutez les commandes suivantes pour lancer les serveurs de développement :

Verifiez que Vue.js fonctionne correctement en lancant:

```
~$ npm run dev
```

```
> dev
> vite
```

```
VITE v5.4.21 ready in 791 ms
```

```
Local: http://localhost:5173/
Network: http://192.168.1.87:5173/
press h + enter to show help
```

```
LARAVEL v12.44.0 plugin v1.3.0
```

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

Accédez à `http://localhost:5173` dans votre navigateur. Vous devriez voir “Hello, Vue 3 + Vite!” s’afficher.

avec Vite dans votre projet Laravel, en lancant:

```
php artisan serv && npm run dev
```

Vous devriez être en mesure d’utiliser les composants Vue au sein de votre projet Laravel.

Autres Configurations Optionnelles



Configuration Tailwind CSS (Optionnel)

Pour ajouter Tailwind CSS, vous pouvez suivre ces étapes supplémentaires :

-
1. Installez Tailwind CSS :

```
npm install tailwindcss postcss autoprefixer
npx tailwindcss init -p
```

2. Configurez Tailwind en ajoutant le chemin des fichiers dans `tailwind.config.js`

:

```
module.exports = {
  content: [
    './resources/**/*.{blade.php',
    './resources/**/*.{js',
    './resources/**/*.{vue',
  ],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

3. Ajoutez les directives Tailwind dans `resources/css/app.css` :

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

4. Incluez le fichier CSS dans votre configuration Vite (`vite.config.js`) :

```
export default defineConfig({  
  plugins: [  
    laravel({  
      input: ['resources/js/app.js',  
              'resources/css/app.css'],  
      refresh: true,  
    }),  
    vue(),  
  ],  
  // ... autre configuration  
});
```

Conclusion

Avec ces étapes, vous avez maintenant un environnement de développement Laravel configuré avec Vite et Vue.js, avec la possibilité d'utiliser Tailwind CSS si nécessaire.