

Compilation des assets (vite)

First React.js "Hello World" with Laravel & Vite

1. Introduction :

Vite est un outil moderne de construction d'applications frontales qui fournit un environnement de développement extrêmement rapide et qui regroupe votre code pour la production. Lorsque vous créez des applications avec Laravel, vous utilisez généralement Vite pour regrouper les fichiers CSS et JavaScript de votre application dans des ressources prêtes pour la production.

Laravel s'intègre parfaitement à Vite en fournissant un plugin officiel et une directive Blade pour charger vos ressources pour le développement et la production.

2. Installation de Node

Vous devez vous assurer que Node.js (16+) et NPM sont installés avant de lancer Vite et le plugin Laravel :

```
node -v  
npm -v
```

3. Installation de Vite et du plugin Laravel

Nous installons React lui-même :

```
npm install react@latest react-dom@latest
```

De plus, pour que React fonctionne avec Vite, nous devons installer un plugin spécifique :

```
npm install --save-dev @vitejs/plugin-react
```

Vous pouvez ensuite inclure le plugin dans votre fichier de configuration **vite.config.js** :

```
import { defineConfig } from 'vite';  
import laravel from 'laravel-vite-plugin';  
import react from '@vitejs/plugin-react';
```

```
export default defineConfig({
  plugins: [
    laravel([
      'resources/css/app.css',
      'resources/js/app.jsx',
    ]),
    react(),
  ],
});
```

Nous ajoutons import react et react() sans paramètres. Remarquez également que j'ai renommé resources/js/app.js en resources/js/app.jsx. C'est exactement le fichier dont nous allons nous occuper ensuite.

1. Afficher Hello World dans Blade

Installons un nouveau projet Laravel et ajoutons du texte statique à la page d'accueil. Mais ce texte statique proviendrait du composant React.js, et non de Blade?

Allons à **resources/views/welcome.blade.php** et ajoutons du HTML statique comme la balise `<h1>` :

```
<h1>Hello world from blade</h1>
```

2. Afficher le composant React.js dans Blade

Donc, à ce stade :

- Nous avons un composant React.
- Nous avons un emplacement id="app" pour le charger.
- Nous avons installé React et configuré Vite.

Enfin, nous lions le tout en créant une application React dans le fichier **resources/js/app.jsx** - en renommant l'ancien fichier par défaut de Laravel **resources/js/app.js**. Au bas de la ligne **import './bootstrap'**, nous ajoutons ceci :

// **resources/js/app.jsx**

```
import './bootstrap';
import '../css/bootstrap.min.css';
import '../css/app.css';

import ReactDOM from 'react-dom/client';
import HelloWorld from '../components/HelloWorld';

ReactDOM.createRoot(document.getElementById('app')).render( <HelloWorld/> );
```

Nous importons le composant et le montons (render) dans la <div id="app"> du fichier Blade.

Maintenant, nous devons démarrer le serveur Vite, en exécutant :

```
npm run dev
```

```
VITE v4.1.2 ready in 21719 ms

→ Local:   http://127.0.0.1:5173/
→ Network: use --host to expose
→ press h to show help

LARAVEL v9.45.0 plugin v0.7.4

→ APP_URL: http://localhost
```

Enfin, dans **resources/views/welcome.blade.php**, ne laissez également que du JS, aucun CSS n'est nécessaire ici :

// **resources/views/welcome.blade.php**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  @viteReactRefresh
  @vite(['resources/js/app.jsx'])

  <title>Document</title>
</head>
<body>
```

```
<div id="app"></div>
<h1>Hello world from blade</h1>
</body>
</html>
```

Vous devrez également inclure la directive supplémentaire **@viteReactRefresh** Blade dans votre directive **@vite** existante.

```
@viteReactRefresh
@vite('resources/js/app.jsx')
```

La directive **@viteReactRefresh** doit être appelée avant la directive **@vite**.