

[Accueil](#) > [Cours](#) > [Wilo Ahadi](#) > Installer Vue.js 3 dans Laravel

Installer Vue.js 3 dans Laravel

Mis à jour il y a 11 mois

Un tutoriel pour télécharger le framework JavaScript VueJs 3 en utilisant npm, le compiler avec laravel-mix puis l'intégrer pour l'utiliser dans un projet Laravel



Domaine

Développement web

Technologies

Laravel, JavaScript, Vue.js

Développeur PHP H/F

Développeur

Sommaire

- [Introduction](#)
- [Télécharger vue.js 3 dans Laravel](#)
- [Compiler et intégrer vue.js 3 dans Laravel](#)
- [Utiliser vue.js 3 sur vue \(template Blade\)](#)

Voir aussi [Un tutoriel pour comprendre et créer \(mettre en place\) un flux RSS dans un projet Laravel](#)

→ [En savoir plus](#)

Introduction

Vue.js est un framework JavaScript open-source qui permet de développer des interfaces utilisateur et des Single Page Application (SPA) ou « applications web monopages » en français.

Nous voulons voir dans ce guide comment mettre en place **Vue js 3** dans un projet **Laravel** en utilisant **npm** et **Laravel Mix**.

👉 Voir aussi :

- [Créer un projet Laravel avec Laragon](#)
- [Installer Vue.js 2 dans Laravel](#)

Télécharger vue.js 3 dans Laravel

Pour télécharger Vue.js dans un projet Laravel, il faut s'assurer d'avoir **Node.js** et **Npm** (Node Package Manager) installés dans votre ordinateur.

Nous pouvons installer la dernière version de Node.js et Npm à partir de [l'installateur officiel de Node.js](#) disponible pour les plateformes Windows, MacOS et Linux.

Pour vérifier que Node.js et Npm sont installés et accessibles à travers la console, on exécute les commandes `node -v` et `npm -v` pour voir leurs versions :

```
C:\laragon\www\akilischool-v3
λ npm -v
6.14.11

C:\laragon\www\akilischool-v3
λ node -v
v14.15.5
```

Dans un nouveau projet Laravel (Je suis à la version 8.69.0), nous avons les dépendances (modules Node) suivants au fichier **package.json** :

```
{
  "devDependencies": {
    "axios": "^0.21",
    "laravel-mix": "^6.0.6",
    "lodash": "^4.17.19",
    "postcss": "^8.1.14"
  }
}
```

Nous devons commencer par télécharger ces dépendances dans le répertoire **/node_modules** en exécutant la commande `npm` suivante :

```
npm install
```

Avant de télécharger vue.js, nous pouvons voir les tags de distributions `dist-tags` disponibles en exécutant la commande `npm view vue` :

```
λ npm view vue
vue@2.6.14 | MIT | deps: none | versions: 359
Reactive, component-oriented view layer for modern web interfaces.
https://github.com/vuejs/vue#readme

keywords: vue

dist
.tarball: https://registry.npmjs.org/vue/-/vue-2.6.14.tgz
.shasum: e51aa5250250d569a3fbad3a8a5a687d6036e235
.integrity: sha512-x2284lgYvjOMj3Za7kqzRcUSxBboHqtgRE2zlos1qWaOye5yUmHn42LB1250NJB1RwEcdrB0JRwyPTEPhfQjiQ==
.unpackedSize: 3.0 MB

maintainers:
- posva <posva13@gmail.com>
- yyx990803 <yyx990803@gmail.com>

dist-tags:
beta: 3.2.0-beta.8  csp: 1.0.28-csp  latest: 2.6.14  next: 3.2.20

published 4 months ago by posva <posva13@gmail.com>
```

Puis pour télécharger Vue 3, on exécute la commande `npm` suivante :

```
npm install vue@next
```

Cela importe Vue dans le répertoire **/node_modules**. Le fichier **package.json** devient :

```
{
  "devDependencies": {
    "axios": "^0.21",
    "laravel-mix": "^6.0.6",
    "lodash": "^4.17.19",
    "postcss": "^8.1.14"
  },
  "dependencies": {
    "vue": "^3.2.21"
  }
}
```

Compiler et intégrer vue.js 3 dans Laravel

Maintenant que `"vue": "^3.2.21"` fait partie de dépendances du projet, nous devons l'intégrer dans l'application Laravel pour l'utiliser.

Commençons par créer un [Single File Component](#) (SFC), nommons-le « **AppComponent.vue** », dans le répertoire `/resources/js/components/`.

Insérons le template et le script suivant au composant `/resources/js/components/AppComponent.vue` créé :

```
<script>
export default {
  data () {
    return {
      message : "Bienvenue sur Vue 3"
    }
  }
}
</script>

<template>
  <div>
    <h1>{{ message }}</h1>
  </div>
</template>
```

Nous pouvons maintenant intégrer Vue avec ce composant dans l'application Laravel. Pour ce faire, éditons le fichier `/resources/js/app.js` de la manière suivante :

1. Importer `createApp` pour créer une nouvelle instance de l'application Vue 3
2. Importer le composant `AppComponent.vue`
3. Monter l'application Vue 3 sur l'élément `div#app` en enregistrant le composant `AppComponent.vue`

L'implémentation de ces instructions nous donne le code source suivant au fichier `/resources/js/app.js` :

```
require('./bootstrap');

// 1. On importe createApp
import { createApp } from "vue"
```

```
// 2. On importe AppComponent.vue
import AppComponent from "./components/AppComponent.vue"

// 3. On monte l'application Vue sur l'élément #app
createApp(AppComponent).mount("#app")
```

Compiler les assets avec Laravel Mix

Pour le support de fichiers ***.vue** (SFC) lors de la compilation avec **webpack**, nous devons appeler la méthode `mix.vue()` au fichier **/webpack.mix.js** :

```
mix.js('resources/js/app.js', 'public/js')
    .vue() // Compiler les *.vue
    .postCss('resources/css/app.css', 'public/css', [
        //
    ]);
```

Puis finalement pour compiler les fichiers CSS et JavaScript, on exécute la commande **npm** suivante :

```
npm run dev
```

Pour compiler et minifier les scripts pour la production :

```
npm run production
```

Si la console vous informe que des dépendances additionnelles sont requises telles que **vue-loader** sur la capture ci-dessous :

```
λ npm run dev

> @ dev C:\laragon\www\laravel-8
> npm run development

> @ development C:\laragon\www\laravel-8
> mix

    Additional dependencies must be installed. This will only take a moment.
    Running: npm install vue-loader@^16.2.0 --save-dev --legacy-peer-deps
```

Installez-les en exécutant la commande `npm install package1 package2 ... :`

```
npm install vue-loader
```

Le fichier **package.json** devient :

```
{
  "devDependencies": {
    "axios": "^0.21",
    "laravel-mix": "^6.0.6",
    "lodash": "^4.17.19",
    "postcss": "^8.1.14",
    "vue-loader": "^16.8.2"
  },
  "dependencies": {
    "vue": "^3.2.21"
  }
}
```

Réexécutez la commande `npm run dev` ou `npm run production` pour compiler les assets.

Si la compilation avec webpack réussit, nous obtenons les fichiers :

- **/public/js/app.js** pour les assets JavaScript
- **/public/css/app.css** pour les assets CSS

Utiliser vue.js 3 sur vue (template Blade)

Maintenant que nous avons les assets JavaScript compilés au fichier **/public/js/app.js**, nous pouvons importer ce fichier sur une vue ([Template Blade](#)) pour utiliser l'application Vue :

```
<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Laravel - Vue 3</title>
</head>
<body>

  <!-- L'élément HTML d'identifiant "app" -->
  <div id="app" ></div>

  <!-- On importe le fichier « /public/js/app.js » -->
  <script type="text/javascript" src="{{ asset('js/app.js') }}" ></script>
```

```
</body>
</html>
```

Bon début de code ! 😎

#laravel

#javascript

#vue.js



Cette publication vous a plu ?

Partagez-la avec vos ami(e)s sur les réseaux sociaux.



Wilo Ahadi, l'auteur

Passionné de l'informatique, je suis spécialiste en techniques des systèmes et réseaux, développeur web et mobile, infographiste et designer, ... J'aime partager mon expérience en formant sur la plateforme Akili School

 Voir profil

 Suivre


Commentaires

0 Commentaires

 S'identifier ▾

Commencer la discussion...

S'IDENTIFIER AVEC

OU INSCRIVEZ-VOUS SUR DISQUS 

Nom

Les meilleurs ▾



Soyez le premier à commenter.

Autres publications



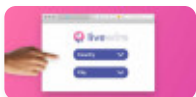
Laravel : Envoyer un mail via le serveur SMTP Google



Laravel : Manipuler les images avec Intervention image



Wow.js et animate.css : Afficher et animer les éléments au niveau de la scroll bar



Laravel Livewire : liste déroulante pour choix de pays et ville



Laravel : Créer un flux RSS valide

→ [Voir toutes les publication de Wilo Ahadi](#)

Sélection ebook



9.99 \$

Créer un site web dynamique en PHP relié à une base de données MySQL - 2

Module 2 : Interaction APPLICATION et BASE DE DONNÉES

 Développement web

Newsletter

Ne manquez plus les nouvelles publications en vous abonnant (Free) à la lettre d'information.

✓ Je m'abonne

Akili School est une plateforme qui vous permet d'apprendre et d'enseigner le numérique.
Trouvez des articles, cours, formations, screencasts et ebooks



📍 République Démocratique du Congo, Kinshasa

≡ Menu

- [Les cours](#)
- [Les formations](#)
- [Les livres](#)
- [Mon compte](#)

🔗 Aide & Support

- [Guide d'utilisation](#)
- [Conditions d'utilisation](#)
- [Politique de confidentialité](#)
- [Nous contacter](#)

✉ Newsletter

✓ Je m'abonne