

# Laravel i webpack

## Instal·lar Bulma i Vue

---



**BULMA**.io



**Vue.js**

**Autor: Dani Prados - [dprados@cendrassos.net](mailto:dprados@cendrassos.net) - curs 20-21**

### **Webpack l'eina per gestionar el Frontend a Laravel.**

Laravel proporciona un seguit d'eines per automatitzar la gestió de recursos del frontend. Entre d'altres permet generar fitxers css unificats, compactar els fitxers de javascript i tot de manera automatitzada i previsible.

---

---

## Requeriments previs

Per poder utilitzar Webpack ens cal tenir instal·lat NodeJS i npm, podem tenir-lo instal·lat localment o podem utilitzar Docker per executar-lo en contenidors.

## Instal·lar Bulma en un projecte Laravel

Primer de tot crearem un projecte amb Laravel

```
$ composer create-project laravel/laravel nom_del_projecte  
$ cd nom_del_projecte
```

Per poder utilitzar Webmix ens cal instal·lar les seves depedències.

```
$ npm install
```

Ara instal·larem Bulma en el nostre projecte

```
$ npm install -D bulma
```

Ara tenim dues opcions, podem utilitzar l'eina postCss o Sass.

Si volem utilitzar postCSS en el fitxer webpack.mix.js hem de tenir una configuració similar a aquesta;

```
const mix = require('laravel-mix');  
  
/*  
|-----  
| Mix Asset Management  
|-----  
|  
| Mix provides a clean, fluent API for defining some Webpack build steps  
| for your Laravel application. By default, we are compiling the Sass  
| file for the application as well as bundling up all the JS files.  
|  
*/
```

---

```
mix.js('resources/js/app.js', 'public/js')  
.sass('resources/sass/app.scss', 'public/css')  
.sourceMaps();
```

Un cop comprovat podem editar el fitxer `/resources/css/app.css` afegint la següent línia.

```
@import "~bulma/css/bulma.css"
```

I després executar

```
$ npm run dev
```

I ja podem utilitzar el nostre full d'estils que estarà disponible a la carpeta `/public/css/app.css`.

Exemple de Layout utilitzant els estils

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1">  
    <title>Hello Bulma!</title>  
    <link rel="stylesheet" href="css/app.css">  
  </head>  
  <body>  
    <section class="section">  
      <div class="container">  
        @yield('content')  
      </div>  
    </section>  
  </body>  
</html>
```

---

## Sass

Amb Webmix també podem utilitzar Sass i aprofitar totes les opcions de personalització i automatització que ens dona. Si instal·lem el paquet Laravel/ui, ens activa el mòdul Sass.

Podem ajustar el fitxer webpack.mix.js per utilitzar Sass.

```
const mix = require('laravel-mix');

/*
|-----
| Mix Asset Management
|-----
|
| Mix provides a clean, fluent API for defining some Webpack build steps
| for your Laravel application. By default, we are compiling the Sass
| file for the application as well as bundling up all the JS files.
|
*/

mix.js('resources/js/app.js', 'public/js')
    .sass('resources/sass/app.scss', 'public/css')
    .sourceMaps();
```

Ara hem de crear la carpeta "sass" si no existeix i el fitxer app.scss.

En el fitxer app.scss posarem el següent contingut.

```
@import "~bulma/bulma"
```

Un cop desat executarem la comanda, el primer cop instal·larà les dependències i ens indicarà que el tornem a executar.

```
$ npm run dev
```

Tornem a executar.

---

```
$ npm run dev
```

I ja tenim els estils a punt.

## Resolució de problemes.

Tant els paquets Laravel/ui com Laravel/Breeze canvien la configuració de Webmix i afegeixen plantilles Blade, un cop instal·lem algun d'aquests paquets hem de tornar a ajustar la configuració a les necessitats del nostre projecte i revisar les plantilles que han afegit perquè utilitzin el nostre Layout i full d'estils.

## Instal·lar Vue

### Opció 1 (Laravel/ui)

Per instal·lar Vue en un projecte Laravel també podem fer servir Webmix.

```
$ composer require laravel/ui  
  
$ php artisan ui vue  
  
$ npm install && npm run dev
```

També podem afegir les plantilles d'auth de Larvel/ui, encara que és més recomanable utilitzar Breeze.

```
$ php artisan ui vue --auth
```

Laravel/ui ens ha instal·lat Vue 2, si volem actualitzar a Vue 3 ens cal fer alguns canvis.

```
$ npm install --save vue@next  
$ npm install --save-dev vue-loader@next  
  
$ npm install && npm run dev
```

Ara ens faltaria fer uns ajustos al fitxer app.js, Laravel/ui l'ha definit per funcionar amb Vue 2 i amb Vue 3 no ens funcionarà.

---

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

import { createApp } from 'vue'

import Component from './components/ExampleComponent.vue';

const app = createApp({});

app.component('example-component', Component);

app.mount('#app');
```

Podem actualitzar la plantilla `welcome.blade.php` per comprovar que funciona correctament.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Hello Bulma!</title>
    <link rel="stylesheet" href="{{ asset('css/app.css') }}">
  </head>
  <body>
    <section class="section" id="app">
      <div class="container">
        <example-component></example-component>
      </div>
    </section>
    <script src="{{ asset('/js/app.js') }}"></script>
  </body>
</html>
```

---

Ja tenim a punt el projecte per poder utilitzar Vue.

Cal tenir present que el paquet Laravel/ui instal·la el Bootstrap en el nostre projecte. Si volem fer servir Bulma o un altre Framework hem d'ajustar la configuració com hem comentat a l'apartat anterior.

També ens afegeix les dependències que té Bootstrap de llibreries Javascript, si no fem servir Bootstrap podem esborrar la línia "require('./bootstrap');" del fitxer app.js.

Per comprovar el funcionament podem iniciar el servidor web.

```
$ php artisan serve
```

Si estem treballant en full d'estils o desenvolupant un component Vue podem utilitzar la següent comanda

```
$ npm run watch
```

Cada cop que detecti un canvi regenerarà el full d'estils i el fitxer de JavaScript automàticament.

## **Opció 2 (Directament)**

Podem instal·lar Vue 3 directament sense utilitzar Laravel/ui. Primer instal·lem el Webmix.

```
$ npm install && npm run dev
```

Instal·lem Vue 3 i les seves depedències.

```
$ npm install --save vue@next  
$ npm install --save-dev vue-loader@next
```

Ara ens faltaria configurar el fitxer webpack.mix.js.

```
const mix = require('laravel-mix');
```

```
/*
|-----
| Mix Asset Management
|-----
|
| Mix provides a clean, fluent API for defining some Webpack build steps
| for your Laravel application. By default, we are compiling the Sass
| file for the application as well as bundling up all the JS files.
|
*/

mix.js('resources/js/app.js', 'public/js')
    .vue()
    .sass('resources/sass/app.scss', 'public/css')
    .sourceMaps();
```

I ja podem instal·lar totes les dependències i fer el primer mix.

```
$ npm install && npm run dev
```

Ara podem provar com funciona Vue 3 en el nostre projecte.

En la carpeta `/resources/js` crearem la carpeta `components` i a dins el fitxer `component.vue`

```
<template>
  <div class="notification">Hola des Vue 3</div>
</template>

<script>
  export default {
    mounted() {
      console.log('Component mounted.')
    }
  }
</script>
```



---

```
}  
</script>
```

Editem el fitxer resources/js/app.js

```
import { createApp } from 'vue'  
  
import Component from './components/component.vue';  
  
const app = createApp({});  
  
app.component('component', Component);  
  
app.mount('#app');
```

Podem actualitzar la plantilla welcome.blade.php per comprovar que funciona correctament.

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1">  
    <title>Hello Bulma!</title>  
    <link rel="stylesheet" href="{{ asset('css/app.css') }}">  
  </head>  
  <body>  
    <section class="section" id="app">  
      <div class="container">  
        <component></component>  
      </div>  
    </section>  
    <script src="{{ asset('/js/app.js') }}"></script>  
  </body>  
</html>
```

---

Un cop ho tenim tot a punt fem un Mix.

```
$ npm run dev
```

Ja tenim a punt el projecte a punt. Per comprovar el funcionament podem iniciar el servidor web.

```
$ php artisan serve
```

Si estem treballant en full d'estils o desenvolupant un component Vue podem utilitzar la següent comanda

```
$ npm run watch
```

Cada cop que detecti un canvi regenerarà el full d'estils i el fitxer de JavaScript automàticament.

### Recursos extres:

Laravel Mix

<https://laravel.com/docs/8.x/mix>

Sass

<https://sass-lang.com/>

Install Vue in Laravel.

<https://stackoverflow.com/questions/63853750/how-to-install-vue-js-in-laravel-8>

Laravel Breeze <https://laravel.com/docs/8.x/starter-kits>