

Création d'un gestionnaire de membres de club sportif : Mise en place de l'environnement pour Laravel

NodeJS et NPM



Pour utiliser *Laravel Mix* vous devez disposer de **NPM**, donc aussi de **node.js**. Si vous n'êtes pas trop sûr d'avoir ça sur votre machine entrez ça dans votre console :

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

Si vous obtenez un numéro de version, c'est que vous en disposez, mais assurez-vous quand même que ce ne soient pas des versions trop anciennes sinon vous aurez des soucis !

Node.js est un environnement d'exécution **JavaScript**, c'est-à-dire qu'il permet d'exécuter du code JavaScript en dehors des navigateurs, directement sur le serveur.

Open-source et cross-platform, Node JS windows donne la possibilité de construire entièrement une application web moderne avec un seul et même langage.

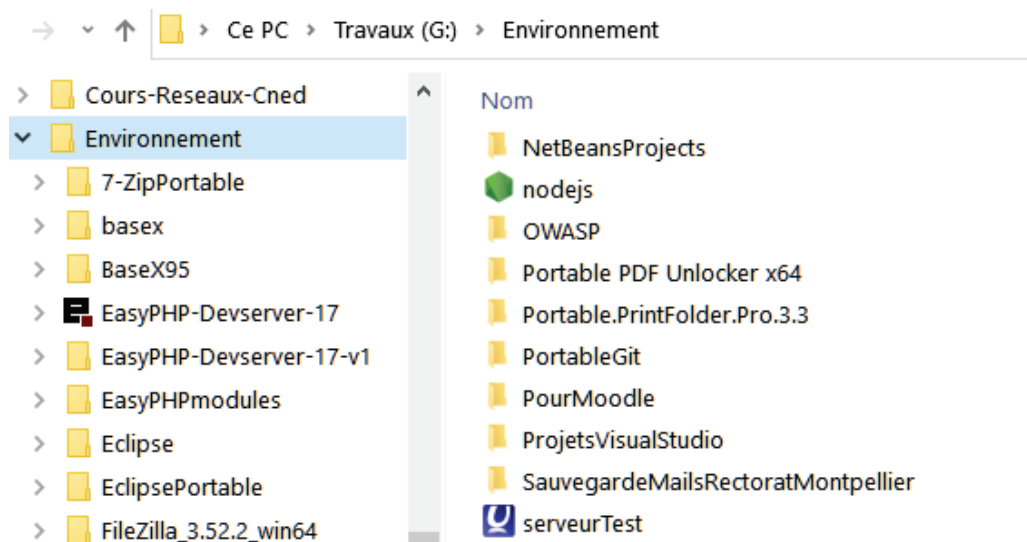
Commencez par installer NodeJS¹

Il est nécessaire de récupérer une version « binaire » de Node.js.

⚠ Ne passez donc pas par l'installateur (.msi) car il modifiera directement le PATH de Windows de manière globale !

Allez sur <https://nodejs.org/en/download/> pour télécharger Node.js dernière version (Windows **Binary** (.zip))

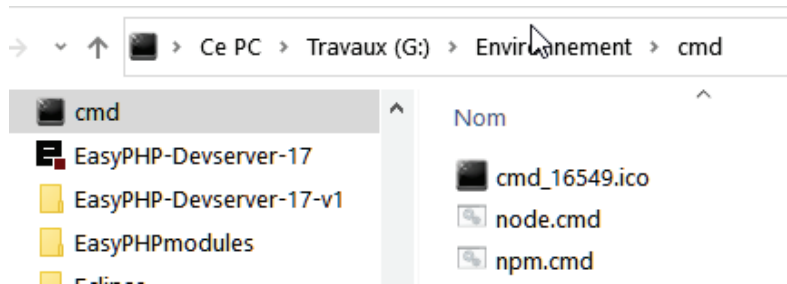
Je l'ai installé (décompressé) par exemple sur mon DD externe de travail



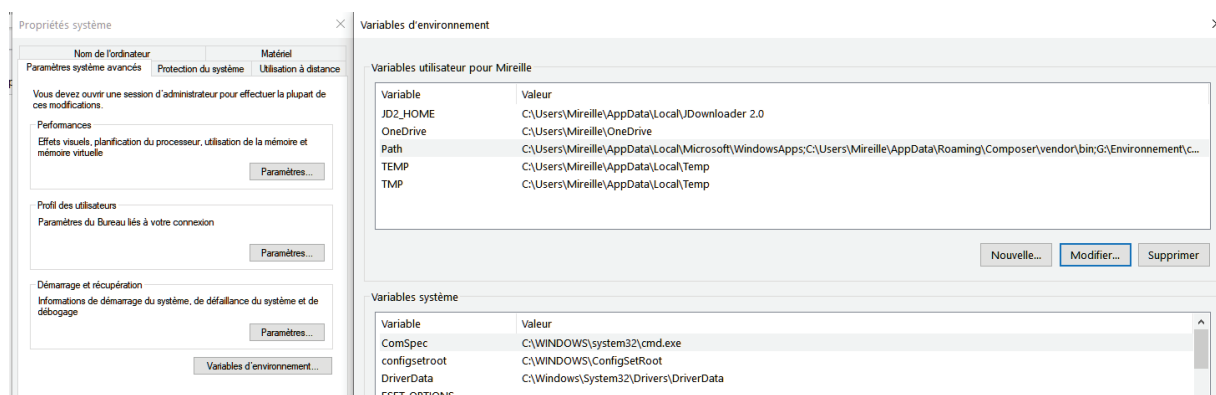
¹ <https://codedesign.fr/tutorial/plusieurs-versions-php-windows-nodejs/>

Créez ensuite un dossier cmd pour les scripts de lancement...

Exemple : G:\Environnement\cmd



⚠ Ajoutons le chemin vers ce nouveau dossier dans la variable d'environnement « Path » de Windows



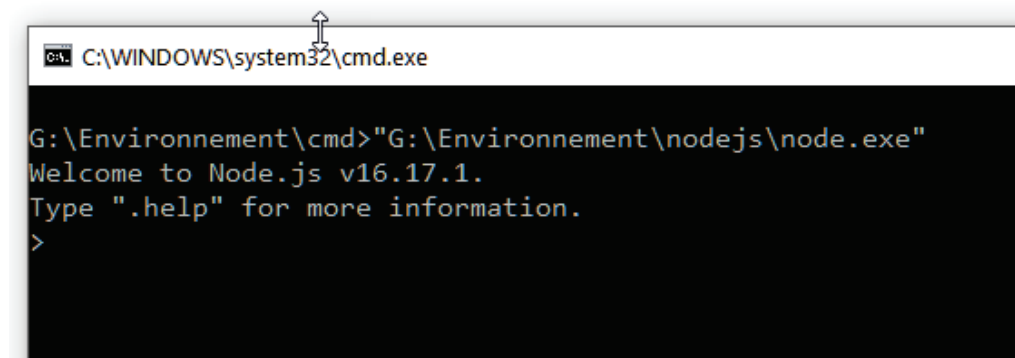
Pour Node

Créer un fichier **node.cmd** et placez-le dans votre dossier
 \cmd\node.cmd (La commande node lancera Node.js depuis une console)

Placez-y les lignes suivantes (à personnaliser en fonction de votre environnement) :

```
"G:\Environnement\nodejs\node.exe" %*
```

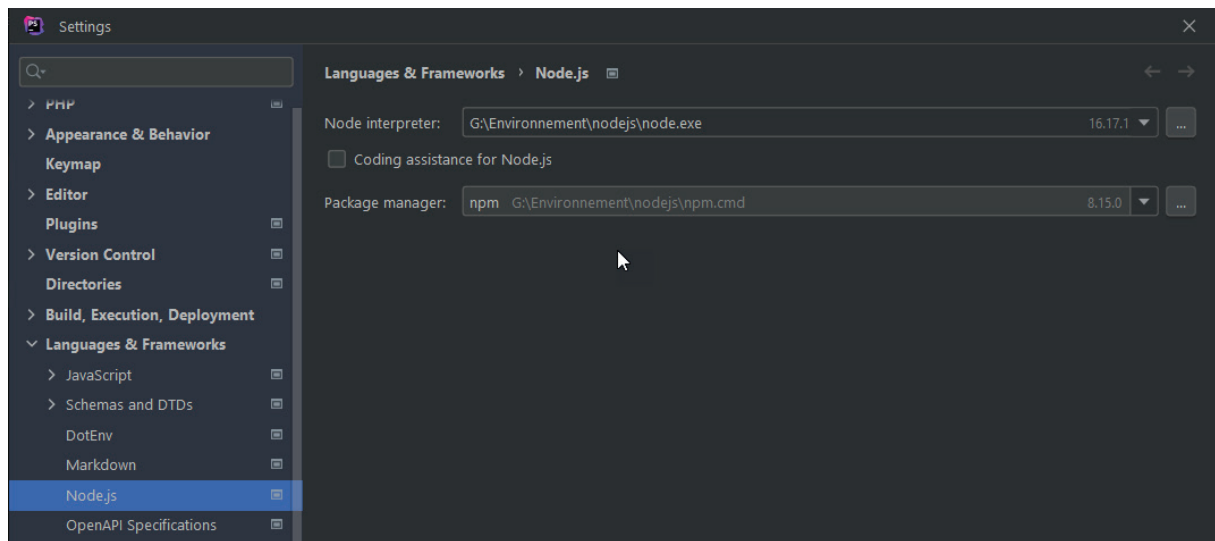
Double cliquez sur votre fichier node.cmd pour lancer nodejs et vérifier que tout est OK



Pour NPM (npm.cmd)

```
"G:\Environnement\nodejs\npm.cmd" %*
```

Paramétrez votre projet PhpStorm (File→settings)



Saisir dans la console phpstorm :

```
G:\Environnement\nodejs\npm.cmd install
```

L'environnement de travail est prêt

Installer Bootstrap 5 via npm et laravel Mix dans Laravel

Introduction

Bootstrap 5 est un framework CSS qui permet de créer des sites réactifs et adaptés aux mobiles. Il contient des codes HTML et CSS pour les formulaires, les boutons, le système de grille réactif et d'autres composants interactifs, ainsi que des plugins JavaScript pour les boîtes modales, les alertes toasts, ...

Nous allons voir comment télécharger le framework CSS Bootstrap 5 dans un projet Laravel avec npm, l'importer puis le compiler avec Laravel Mix et l'intégrer dans une page web (template Blade) pour l'utiliser.

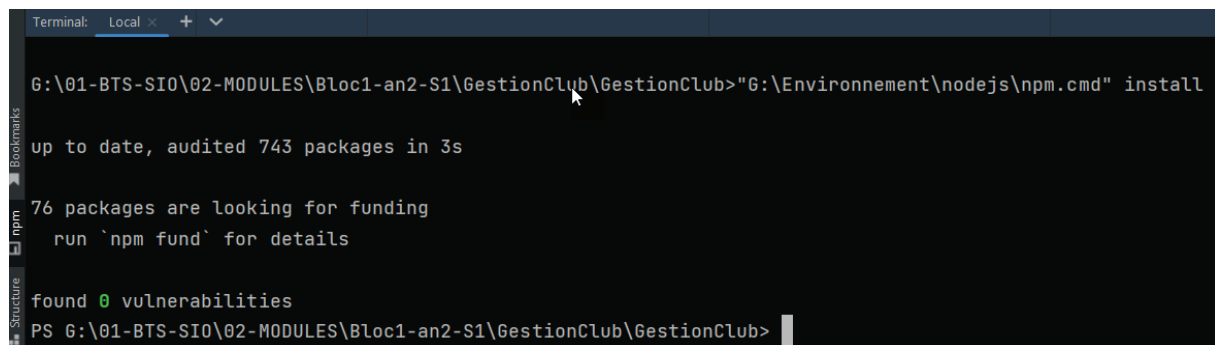
Télécharger Bootstrap 5 via npm

Si vous regardez le contenu du fichier `/package.json` à la racine du projet, on y trouve les lignes suivantes :

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

Nous pouvons télécharger les packages « axios », « laravel-mix », « lodash » et « postcss » dans le dossier `/node_modules` en exécutant la commande npm suivante :

```
npm install
```



```
Terminal: Local + v
G:\01-BTS-SIO\02-MODULES\Bloc1-an2-S1\GestionClub\GestionClub>G:\Environnement\nodejs\npm.cmd install
up to date, audited 743 packages in 3s
76 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
PS G:\01-BTS-SIO\02-MODULES\Bloc1-an2-S1\GestionClub\GestionClub>
```

Nous allons utiliser laravel-mix pour compiler les fichiers JavaScript et Sass de Bootstrap 5 dans le dossier **/public** de Laravel.

Pour voir la version actuelle de Bootstrap, nous pouvons exécuter la commande *npm* suivante :

```
npm view bootstrap version
```

Puis pour télécharger Bootstrap 5 dans le dossier `/node_modules` et mettre à jour automatiquement le fichier `/package.json` :

```
npm install bootstrap --save-dev
```

Le sous dossier « bootstrap » a été rajouté dans le répertoire `node_modules` de votre application

Finalement, nous devons avoir les lignes suivantes au fichier `/package.json` :

```
"devDependencies": {
  "axios": "^0.21",
  "bootstrap": "^5.2.1",
  "laravel-mix": "^6.0.6",
  "lodash": "^4.17.19",
  "postcss": "^8.1.14"
}
```

Maintenant que nous avons Bootstrap 5 téléchargé dans le dossier `/node_modules` du projet Laravel, nous devons importer ses fichiers Javascript et Sass² avant de les compiler avec Laravel Mix.

📁 Pour les fichiers Sass de Bootstrap, créons le dossier « `scss` » dans `/resources` puis un nouveau fichier `app.scss` dans `/resources/scss/`.

Insérons ensuite la ligne suivante au fichier `/resources/scss/app.scss` :

```
@import "bootstrap";
```

Pour le JavaScript de Bootstrap, insérons la ligne suivante au fichier `/resources/js/app.js` :

```
import "bootstrap";
```

Les sources Sass et JavaScript de Bootstrap seront importés depuis `/node_modules/bootstrap/`

Compiler les assets Bootstrap avec Laravel Mix

Pour compiler les fichiers JavaScript et Sass de Bootstrap avec Laravel Mix, nous devons éditer le fichier `/webpack.mix.js` de la manière suivante :

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

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

mix.js('resources/js/app.js', 'public/js')
  .postCss('resources/css/app.css', 'public/css', [
    //
  ])
  .sass("resources/scss/app.scss", "public/css");
```

² Derrière l'acronyme se cache une promesse : des feuilles de style syntaxiquement impressionnantes. SASS est à la fois un préprocesseur et un langage de feuille de style. Le fait qu'il s'agisse d'un préprocesseur est dû au fait qu'il doit être converti. Avant que SASS puisse être lu, le code source doit être compilé en CSS ordinaire
<https://openclassrooms.com/fr/courses/6106181-simplifiez-vous-le-css-avec-sass/6596497-utilisez-les-preprocesseurs-css-pour-des-fonctionnalites-avancees>

Nous pouvons finalement compiler les assets Bootstrap 5 dans les fichiers

📄 /public/js/app.js et /public/css/app.css en exécutant la commande npm suivante :

```
npm run dev
```

Lors de la compilation, les packages « sass-loader », « sass » et « resolve-url-loader » nous seront demandés et téléchargés

🔗 Il vous faudra relancer *npm run dev*

Installer Font Awesome

Font Awesome est une police d'écriture (police de caractères) qui permet d'insérer des icônes sur une page web à partir des propriétés CSS liées aux icônes.

Nous allons voir comment télécharger Font Awesome dans un projet Laravel en utilisant npm, importer ses fichiers Sass et les compiler avec laravel-mix puis afficher les icônes sur une page web.

Exécutez la commande npm suivante :

```
npm install --save @fortawesome/fontawesome-free
```

Cette commande télécharge @fortawesome/fontawesome-free/ dans le répertoire /node_modules. Le fichier /package.json est aussi mis à jour, présentant la version installée de Font Awesome :

```
"dependencies": {
  "@fortawesome/fontawesome-free": "^6.2.0"
}
```

Nous allons importer Font Awesome à partir de ses fichiers Sass (.scss)

Au fichier 📄 /resources/scss/app.scss, insérons les lignes suivantes pour importer les fichiers .scss de Font Awesome avec les styles d'icônes « solid », « regular » et « brands » depuis le répertoire /node_modules

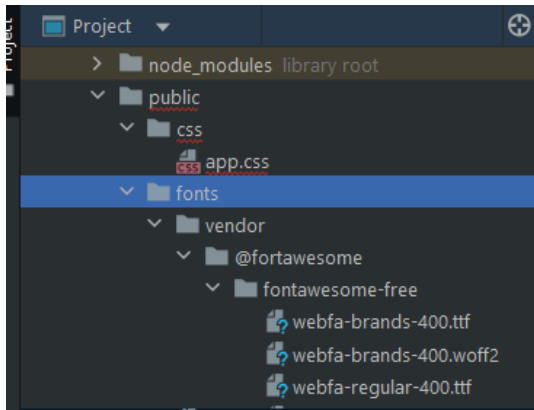
Au fichier 📄 /resources/scss/app.scss, insérons les lignes suivantes pour importer les fichiers .scss de Font Awesome avec les styles d'icônes « solid », « regular » et « brands » depuis le répertoire /node_modules :

```
/* Importation de Font Awesome */
@import "~@fortawesome/fontawesome-free/scss/fontawesome";

/* Importation des styles d'icônes */
@import "~@fortawesome/fontawesome-free/scss/solid";
@import "~@fortawesome/fontawesome-free/scss/regular";
@import "~@fortawesome/fontawesome-free/scss/brands";
```

Nous pouvons finalement compiler les assets dans les fichiers 📄 /public/js/app.js et /public/css/app.css en exécutant la commande npm suivante :

```
npm run dev
```



Cette commande compile le fichier /resources/scss/app.scss vers /public/css/app.css puis importe les fonts Font Awesome dans /public/fonts/vendor/@fortawesome/fontawesome-free/

Usage

La compilation terminée, nous pourrions importer le fichier /public/app.css dans une page web (template Blade) pour utiliser les classes de Font Awesome

Exemple

```
<!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 - Fontawesome</title>

    <!-- Importation de /public/css/app.css -->
    <link rel="stylesheet" type="text/css" href="{{ asset('css/app.css') }}">

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

Font Awesome présente les classes suivantes pour les différents styles d'icônes :

- « fas » : icônes de style « Solid » (Free)
- « far » : icônes de style « Regular » (Pro)
- « fal » : icônes de style « Light » (Pro)
- « fad » : icônes de style « Duotone » (Pro)
- « fab » : icônes « Brands » (Free)

À chacune de ces classes nous devons ajouter une autre classe avec le préfixe « fa- » suivi du nom du caractère (nom de l'icône) à afficher à travers une balise ou <i> de la manière suivante :

```
<span class="fas fa-home" ></span>
<span class="far fa-credit-card" ></span>
<span class="fal fa-car" ></span>
<span class="fab fa-amazon" ></span>
```

[La page Icons](https://fontawesome.com) chez fontawesome.com présente la liste de toutes les icônes disponibles.

Voir aussi : <https://fontawesome.com/docs/web/add-icons/how-to>

Et <https://fontawesome.com/docs/web/use-with/scss>

<https://fontawesome.com/docs/web/style/styling>