**VUE**

**Dure totale du module : 21H00**

Table des matières

[Objectifs pédagogiques : 3](#_Toc111650657)

[Présentation et installation de VUE 4](#_Toc111650658)

[Présentation de VUE : 4](#_Toc111650659)

[Installation de VUE (mode CDN) : 7](#_Toc111650660)

[Installation de VUE (mode CLI) 7](#_Toc111650661)

# Objectifs pédagogiques :

Être capable de comprendre les enjeux des Frameworks

Être capable d'initialiser et paramétrer un projet grâce à un Frameworks front-end

Être capable de réaliser des interface utilisateur grâce à un Frameworks front-end

~~Être capable de mettre en place un système de Routing (url)~~

Être capable de gérer des requêtes HTTP (BDD)

Être capable de connecter une application à une base de données

# Présentation et installation de VUE

## Présentation de VUE :

****

**Qu’est-ce que VUE ?**

VUE est un **Framework Javascript** permettant de construire des applications **Front End** interactives, et **réactives** plus facilement.

**Framework Javascript** : ensemble de librairie(s) tierce(s) qui met à disposition des utilitaires, des fonctionnalités et des bonnes pratiques sur comment construire des applications Javascript

**Javascript** : Un langage de programmation supporté par tous les navigateurs.  
Permet de manipuler les éléments HTML, les données dans une page web.

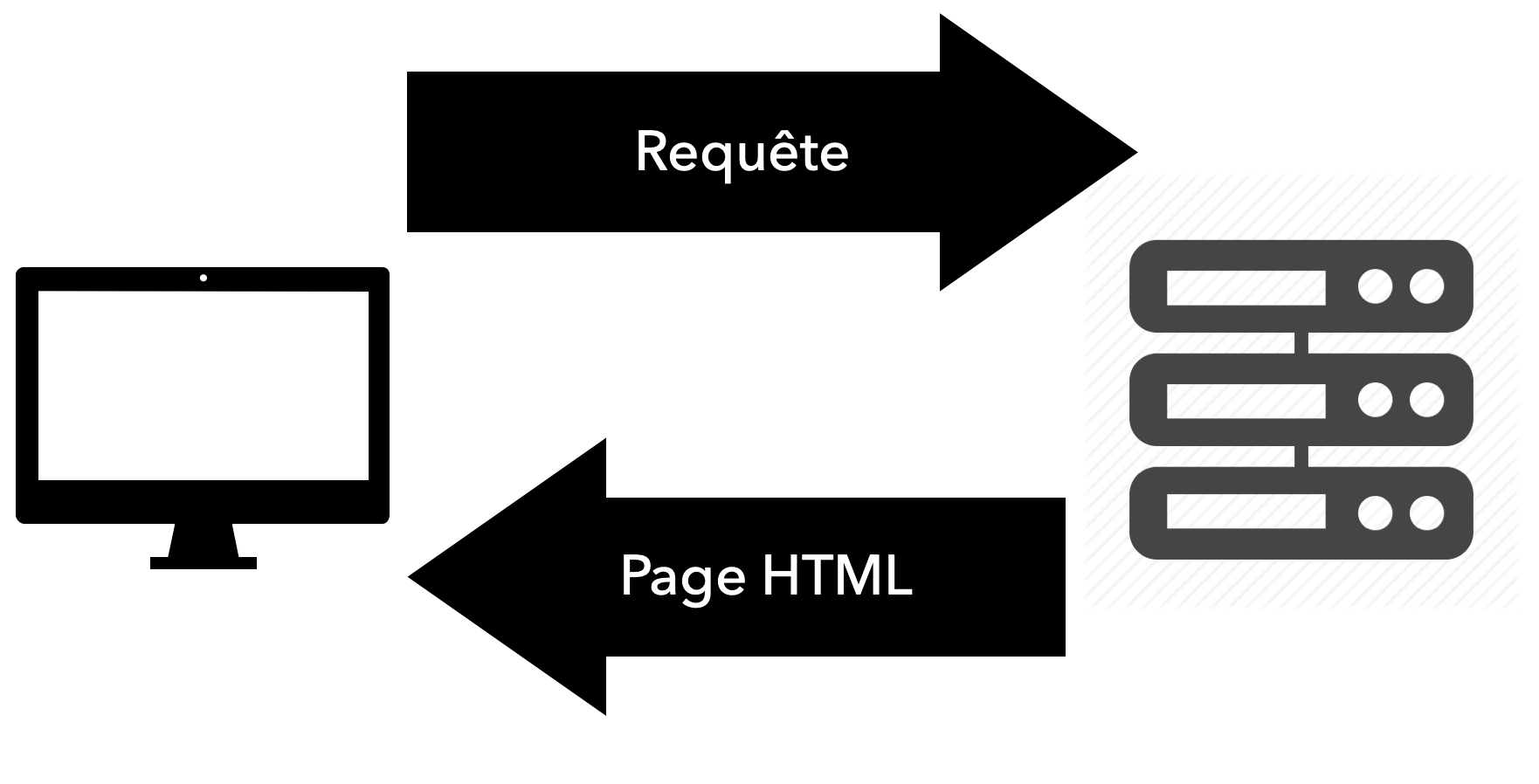
**Front End** : « Ce que l’utilisateur voit » HTML + CSS + Javascript.

**Réactives** : L’App est capable de réagir aux actions de l’utilisateur et mettre à jour ce qui s’affiche à l’écran. ==> Le Look et le Feeling d’une appli mobile.

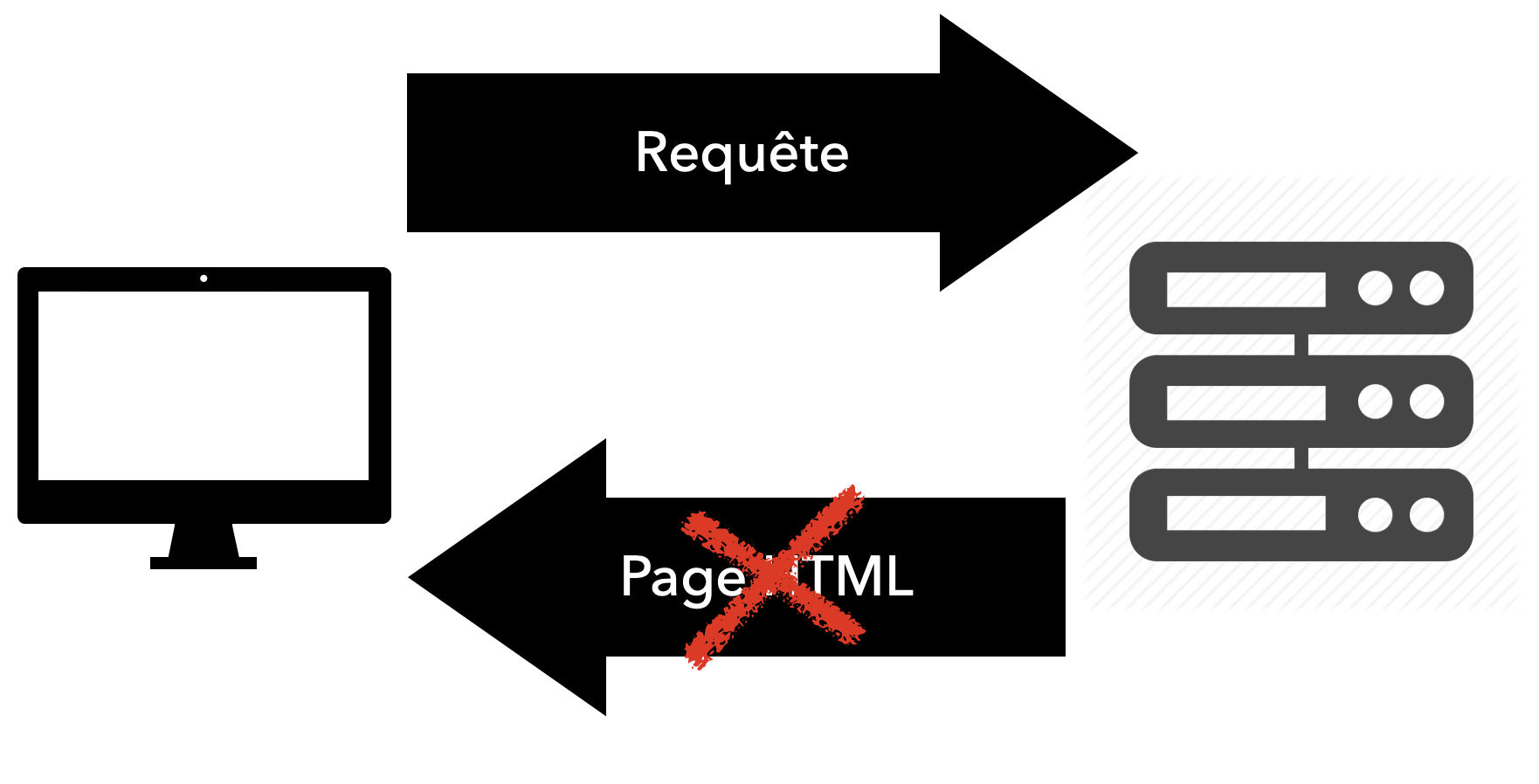
[#](https://worldline.github.io/vuejs-training/fr/presentation/#qu-est-ce-que-vue-js) « À la différence des autres Frameworks monolithiques, Vue a été conçu et pensé pour pouvoir être adopté de manière incrémentale. Le cœur de la bibliothèque est concentré uniquement sur la partie vue, et il est vraiment simple de l’intégrer avec d’autres bibliothèques ou projets existants. D’un autre côté, Vue est tout à fait capable de faire tourner des applications web monopages quand il est couplé avec des outils modernes et des bibliothèques complémentaires. »

Vue.js a été créé à l'origine par une seule personne, Evan You, ancien ingénieur de Google ayant notamment travaillé avec Angular.js puis sur le framework Meteor. Après avoir expérimenté une mécanique de réactivité qu'il jugeait plus intéressante que celle d'Angular, il décide de publier ses premiers résultats en juillet 2013. Cinq ans plus tard, Vue.js est le 3ème projet sur Github en nombre de stars et Evan enchaîne les conférences à travers le monde.

Généralement, dans des applications web « classiques », les interactions avec le serveur (click sur un bouton pour envoyer un message par exemple) entrainent un rechargement de la page.



L’arrivée de Javascript a permis d’éviter ces rechargements de pages, dans le sens ou JS s’exécute dans le navigateur sur la page qui est chargée, JS va pouvoir manipuler les éléments du DOM de la page.



Sur l’aspect technique, il est tout à fait possible de construire n’importe quelle application web en se basant sur du Javascript, sans recourir à l’utilisation d’un Framework (Angular, Vue, React, Svelte …)

Mais :

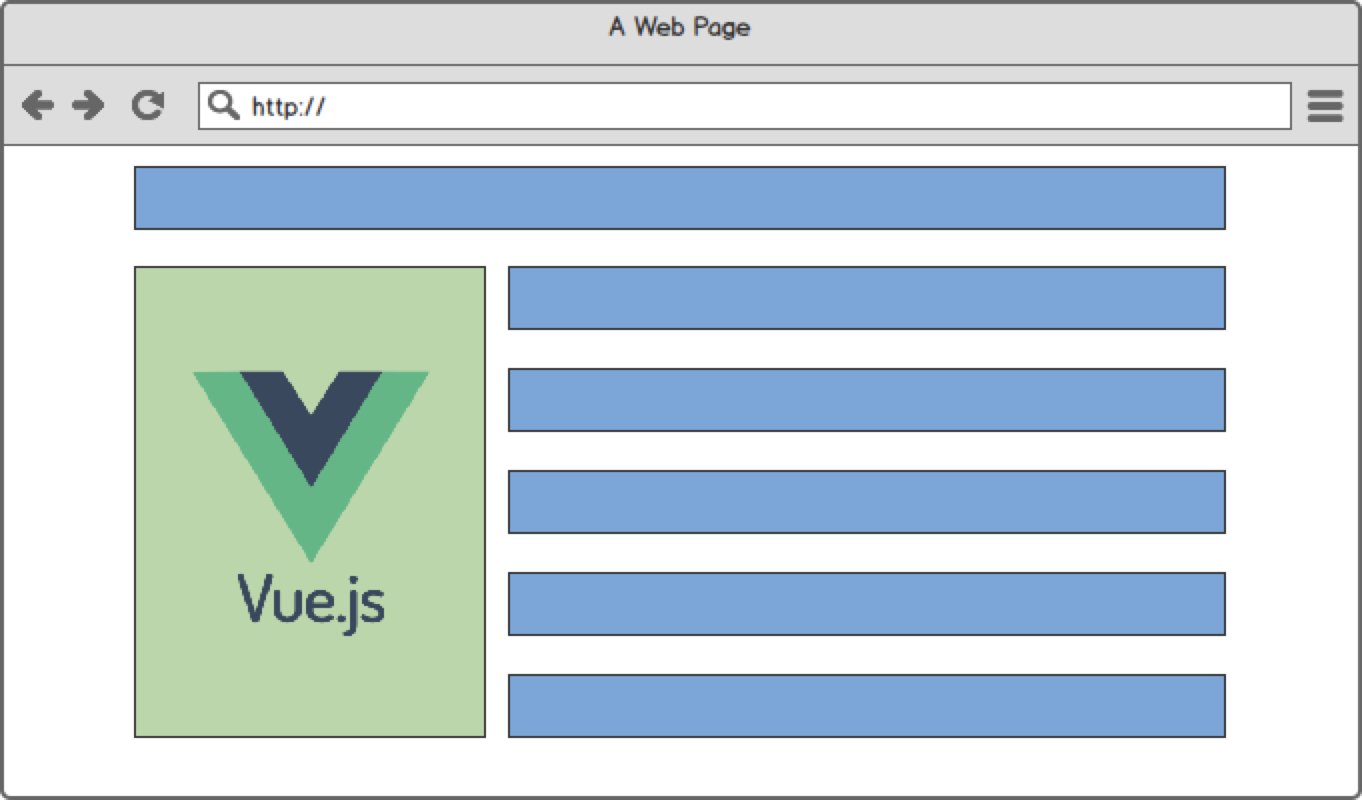
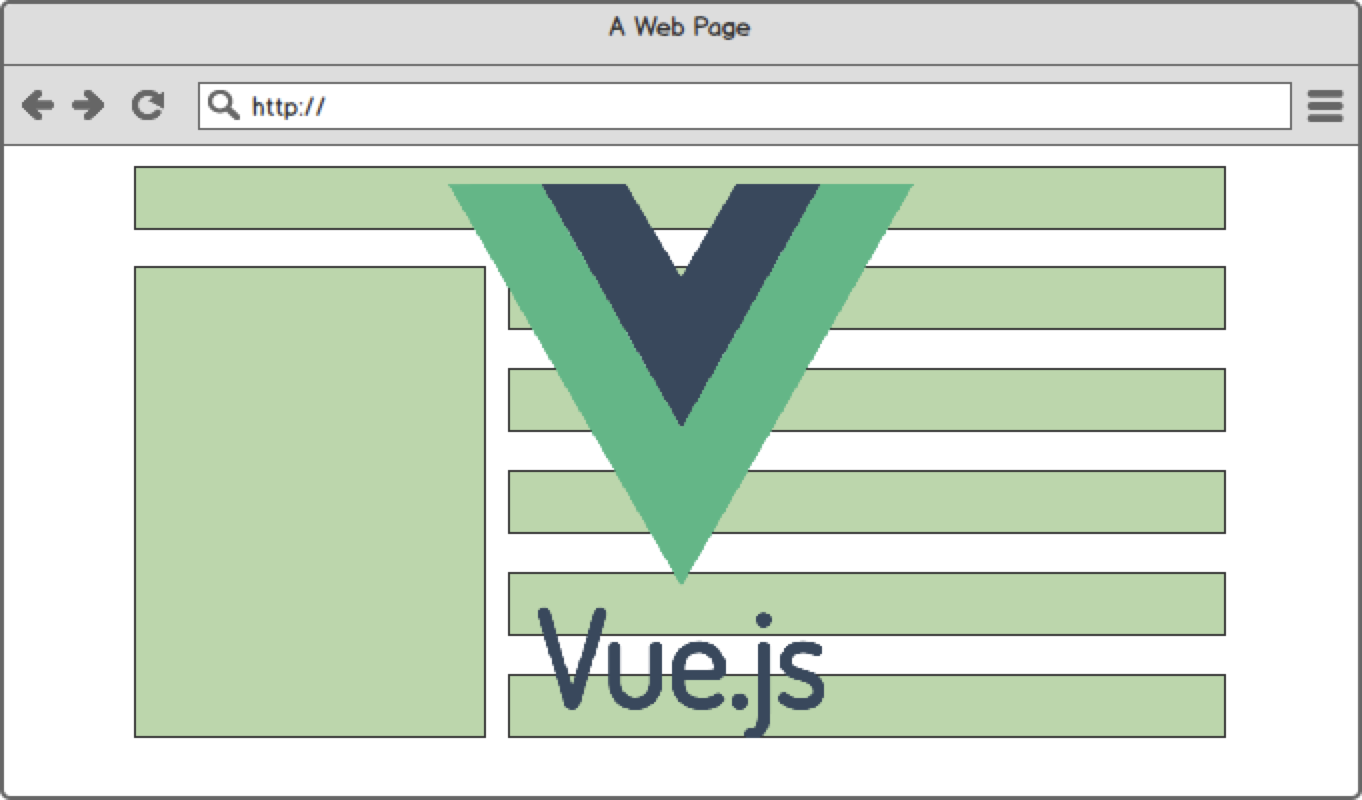
* Nous devrons écrire tout le code de l’application (On réinvente la roue ?)
* Risque de réaliser un code non optimisé comprenant des erreurs, des bugs (difficilement maintenable / scalable).
* Travailler en équipe sera plus difficile, tout le monde ne connait pas votre manière de coder.

L’utilisation d’un Framework permet de répondre à ces problématiques, il propose un ensemble de composants réutilisables, un cadre de travail avec des règles à respecter ainsi que de bonnes pratiques.

VUE adopte une approche différente des autres frameworks (monolithique), Vue se définit comme un framework progressif, on peut choisir d’utiliser de deux manières :

L’approche plutôt en mode widget (maquette gauche) (Vue contrôle 1 ou plusieurs parties spécifiques d’une application), cela permet de facilement intégrer et utiliser vue dans des projets existants.

L’approche en mode « monolithique » (maquette droite), on gère toute une application en VUE

## Installation de VUE (mode CDN) :

En mode CDN, nous n’allons pas Installer VUE, nous allons directement l’utiliser depuis son server d’hébergement.

<https://vuejs.org/guide/quick-start.html#without-build-tools>

Il suffira d’intégrer cette balise dans nos pages HTML :

<script src="https://unpkg.com/vue@3"></script>

## Installation de VUE (mode CLI)

(Nous mettrons en place cette technique dans la seconde partie du cours)  
En mode CLI (Command Line Interface), nous pouvons créer des projets via des lignes de commande (terminal) entièrement gérés en VUE.

Cette méthode d’installation passe par des outils de gestion de package (il en existe plusieurs mais le plus commun est l’outil NPM qui fait partie de NodeJS)

<https://nodejs.org/en/>

Une fois NodeJS installé vous pouvez utiliser son outil de gestionnaire de package NPM pour installer dans vos projets tout type de librairies, Framework… et notamment VUE CLI

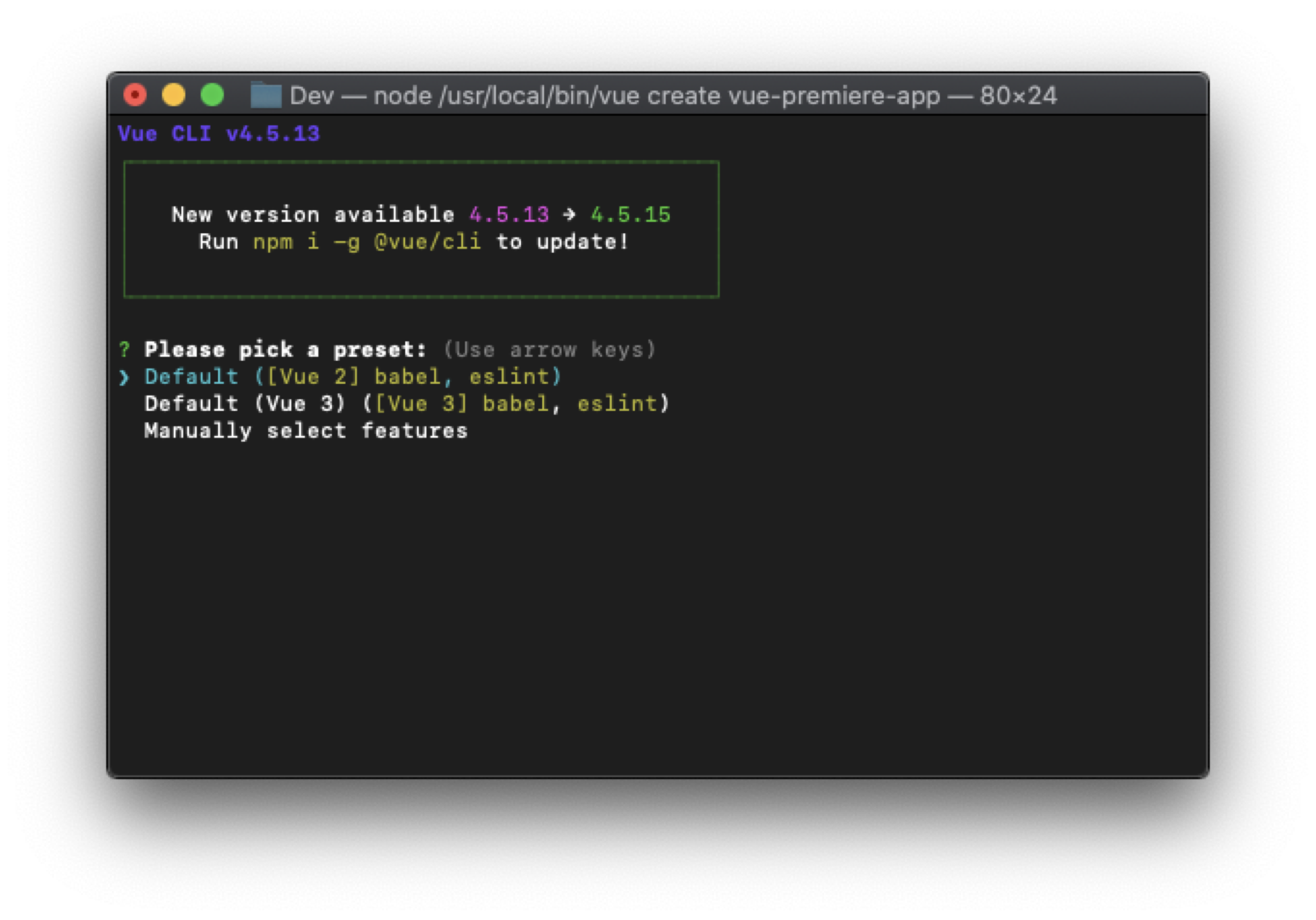
Dans un terminal, la commande ci-dessous va installer VUE CLI sur votre machine :

* npm install -g @vue/cli

Le « -g » va permettre d’installer VUE CLI globalement sur votre machine vous pourrez par la suite utiliser des commandes pour créer des projets VUE dans n’importe quel dossier de votre machine.

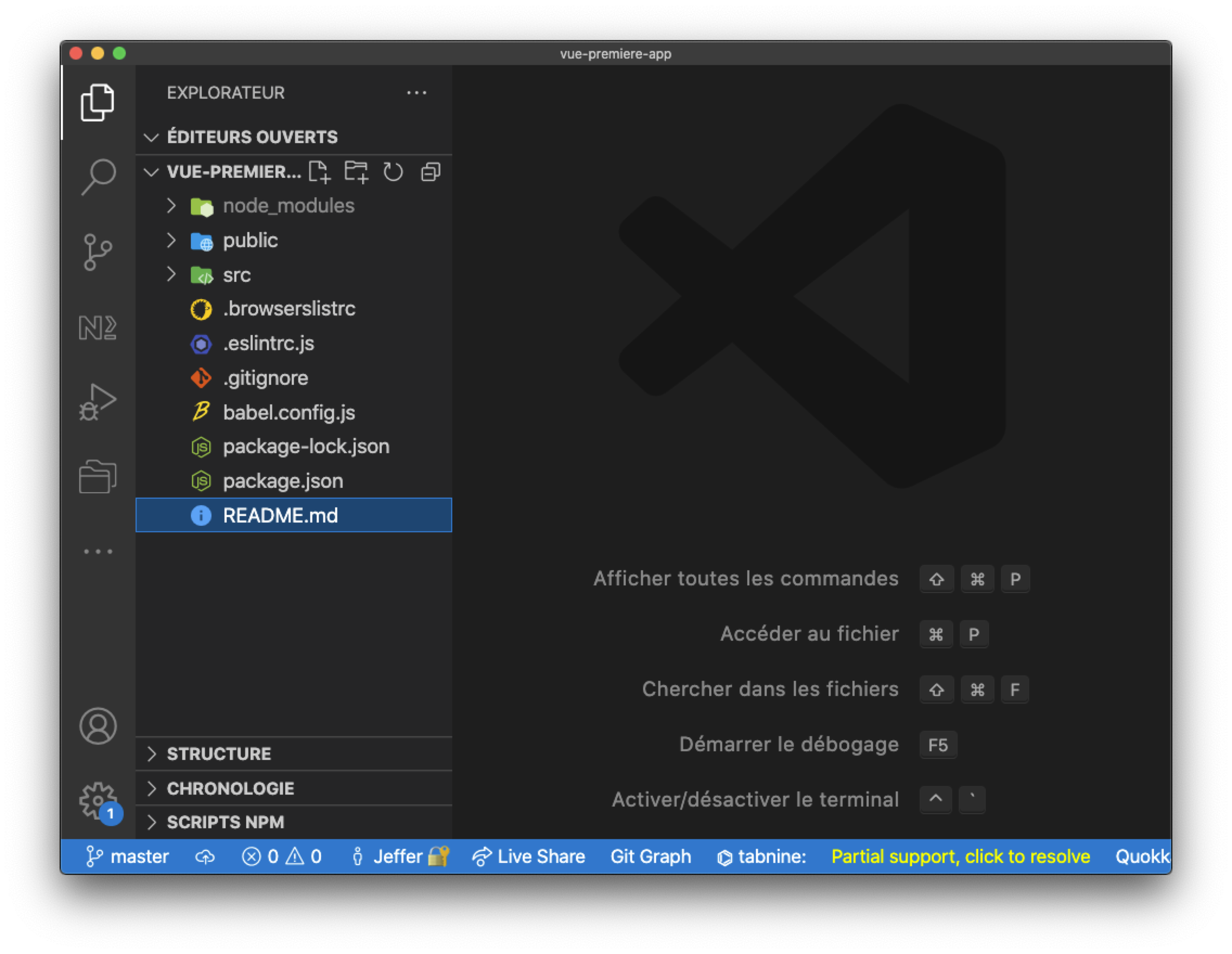
Pour créer et initialiser un nouveau projet VUE :

* Se déplacer dans le terminal dans un dossier (avec la commande cd)
* Et créer une nouvelle application avec la commande :   
  vue create leNomDeVotreApplication



Le CLI de VUE permet de créer et initialiser directement tout un projet, nous pouvons choisir entre Vue2, Vue3 (La dernière version de Vue que nous utiliserons) et la possibilité de configurer manuellement un projet.

Le CLI va donc créer un nouveau dossier pour votre projet (que vous pouvez ouvrir dans VSCode) avec tout un ensemble de fichiers de base.



Dossier node\_modules : c’est dans ce dossier que sont installées toutes les packages, dépendances du projet (exemple si l’on choisit d’installer Bootstrap dans notre projet, les fichiers de Bootstrap se retrouveront dans node\_modules)

Dossier public : ou l’on retrouve la page index.html, la seule page de l’application (qui affichera un composant).

Dossier src : c’est dans ce dossier que l’on développera nos composants, de base toute l’application vue possède un composant racine le fichier App.vue, ainsi qu’un autre composant HelloWorld.vue

Fichier eslint : le fichier de configuration de l’analyseur de code

Fichier babel : le fichier de configuration de Babel (un compilateur JS), son rôle est de traduire des fichiers .vue en JavaScript interprétable directement par les navigateurs

Fichier package.json : le fichier qui va lister les dépendances du projet.