



mais ce que je veux savoir en gros c'est surtout
best practices, comment faire un SPA,
comment loader du data d'un API pi faire que
ce soit reactive, style view/model

ok!

je risque de pas etre capable de tout passer en
revue en 30 min XD

mais je note

A photograph of a man with dark hair and a white t-shirt singing into a blue and silver microphone. A woman with long brown hair is visible on the left, holding a clear glass. The background is a blurred indoor setting with green and blue lights.

Javascript

Vue.js

pas l'temps de niaiser

Tommy Lebel, 25 octobre 2018

5 étapes essentielles pour
démarrer du bon pied avec Vue.js

Par où commencer ?

1. La base

La (très sous-estimée) documentation officielle:

<https://vuejs.org>



Le Framework JavaScript Évolutif



POURQUOI VUE.JS ?

COMMENCER



GITHUB

Sponsors spéciaux



Créer des APIs pour votre navigateur web

Accessible

Familier avec HTML, CSS et JavaScript ? Lisez le guide et développez sans plus attendre !

Polyvalent

Un écosystème incrémentalement adoptable faisant le pont entre une bibliothèque et un framework complet.

Performant

Minimum utile de 20ko min+gzip.
Ultra rapide avec son DOM virtuel.
Effort d'optimisation minimal.

The screenshot shows a code editor on the left displaying the contents of `index.html`, and a web browser window on the right showing the rendered output of the code.

index.html Content:

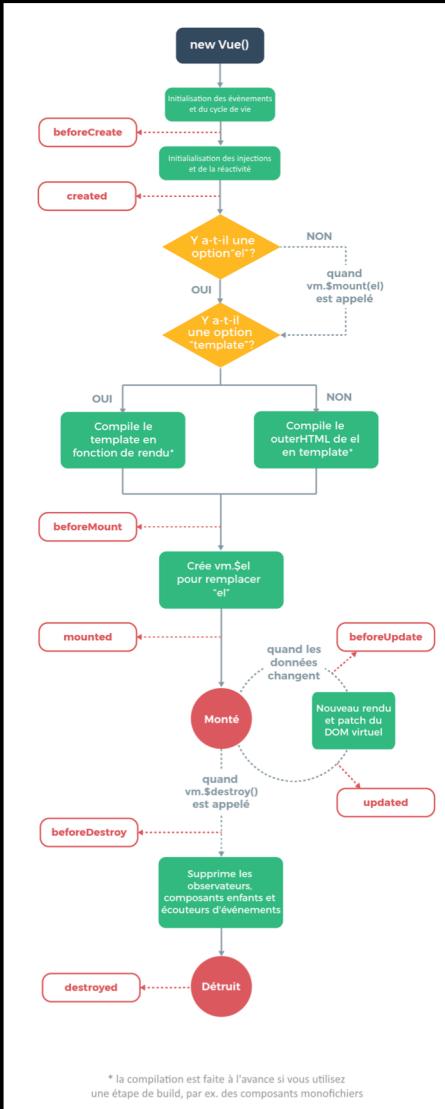
```
1 <div id="app">
2   <ul>
3     <li v-for="product in products">
4       {{ product }}
5     </li>
6   </ul>
7 </div>
8
9
10 <script src="https://unpkg.com/vue"></script>
11 <script>
12   const app = new Vue({
13     el: '#app',
14     data: {
15       products: [
16         'Boots',
17         'Jacket',
18         'Hiking Socks'
19       ]
20     }
21   })
```

Rendered Output in Browser:

- Boots
- Jacket
- Hiking Socks

Extrait en accéléré de la vidéo officielle "Pourquoi Vue.js": <https://fr.vuejs.org/>

Instance de Vue – Cycle de vie

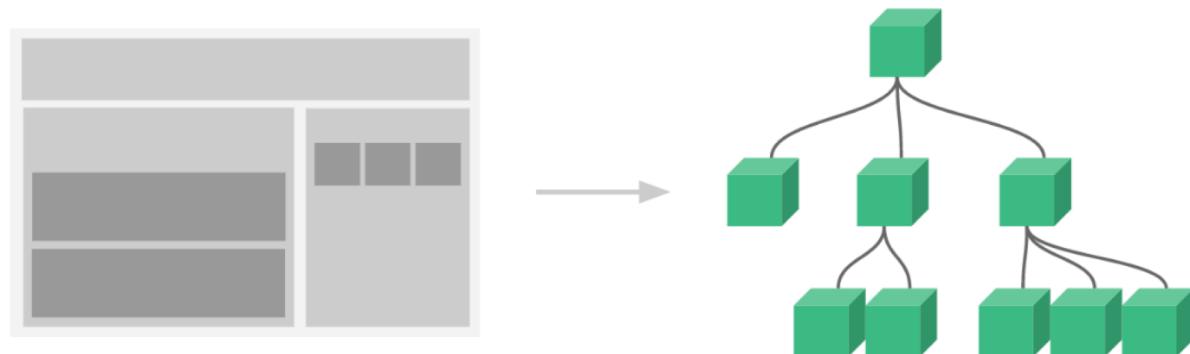


<https://fr.vuejs.org/v2/guide/instance.html#Diagramme-du-cycle-de-vie>

Les composants

Organisation des composants

Il est commun pour une application d'être organisée en un arbre de composants imbriqués :



Par exemple, vous pouvez avoir des composants pour l'entête, la barre latérale, la zone de contenu ; chacun contenant lui aussi d'autres composants pour la navigation, les liens, les billets de blog, etc.

Section “Composants en détail”

Création de composants

Cette page suppose que vous avez déjà lu les principes de base des [composants](#). Lisez-les en premier si les composants sont quelque chose de nouveau pour vous.



Join over 300,000
developers using
CircleCI's integration
with GitHub.

ads via Carbon

Noms de composant

Lors de la création de composants, il faudra toujours spécifier un nom. Par exemple, la déclaration se fera comme suit :

```
Vue.component('my-component-name', { /* ... */ })
```

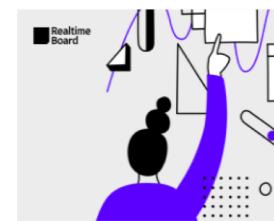
JS

Le nom du composant est le premier argument de `Vue.component`.

Section “Réactivité dans le détail”

Réactivité dans le détail

C'est maintenant l'heure du grand plongeon ! L'une des fonctionnalités les plus emblématiques de Vue est le système de réactivité non obstrusif. Les modèles sont simplement des objets JavaScript. Lorsque vous les modifiez, la vue se met à jour. Cela rend la gestion d'état très simple et intuitive, mais il est également important de comprendre comment cela fonctionne pour éviter quelques erreurs classiques. Dans cette section, nous allons nous pencher sur certains détails de bas niveau du système de réactivité de Vue.



An infinite workspace
for visual teams. Sign
up for a free trial of
RealtimeBoard.

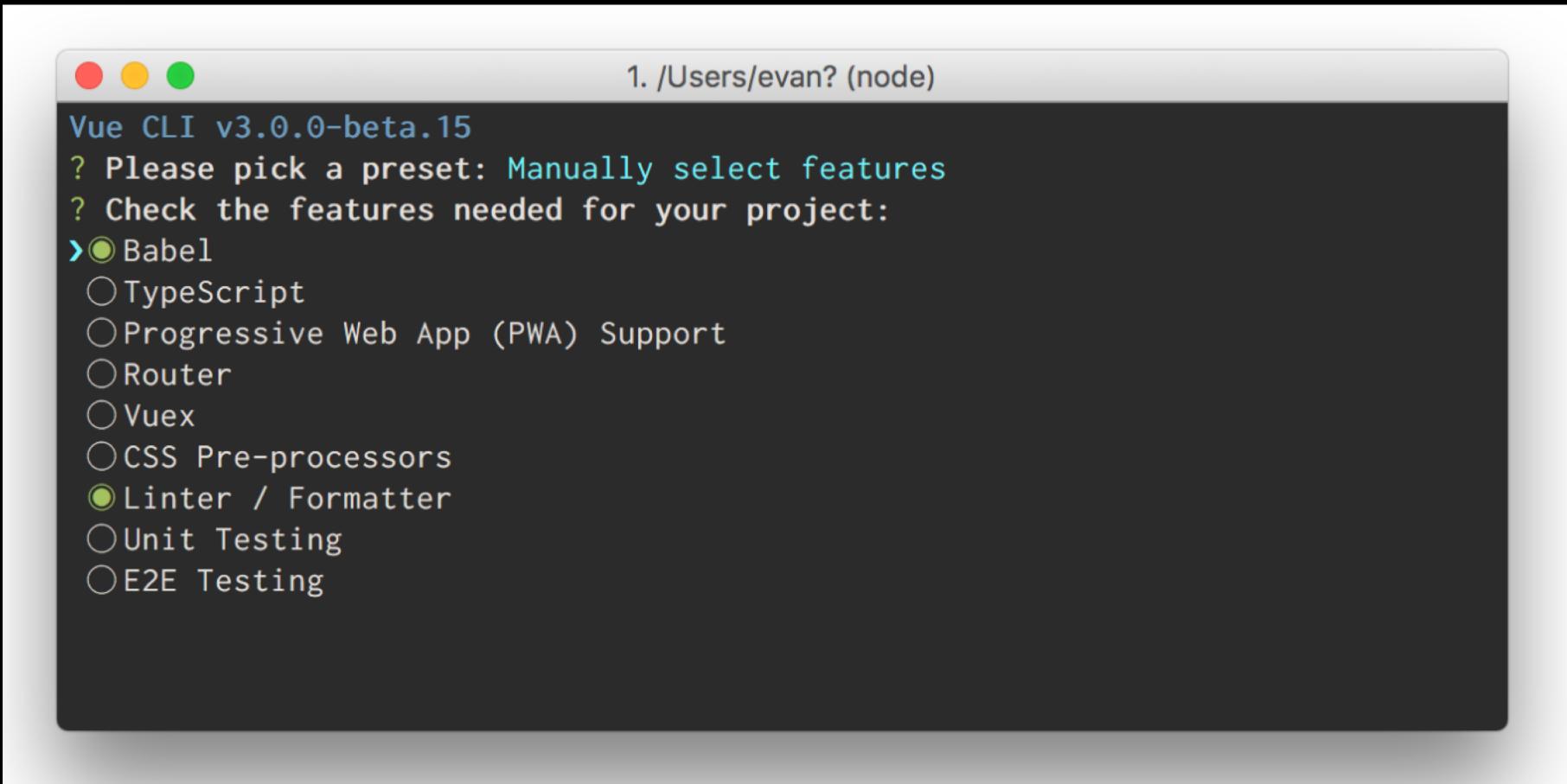
ads via Carbon

Comment les modifications sont tracées ?

Lorsque vous passez un simple objet JavaScript à une instance de Vue via son option `data`, Vue parcourra toutes ses propriétés et les convertira en accesseurs / mutateurs (« `getters/setters` ») en utilisant `Object.defineProperty`. Il s'agit d'une fonctionnalité ES5.

2. Les (incontournables) bibliothèques officielles

vue-cli 3.0



1. /Users/evan? (node)

```
Vue CLI v3.0.0-beta.15
? Please pick a preset: Manually select features
? Check the features needed for your project:
>● Babel
○ TypeScript
○ Progressive Web App (PWA) Support
○ Router
○ Vuex
○ CSS Pre-processors
● Linter / Formatter
○ Unit Testing
○ E2E Testing
```

<https://cli.vuejs.org/>

Un CLI GUI 😱

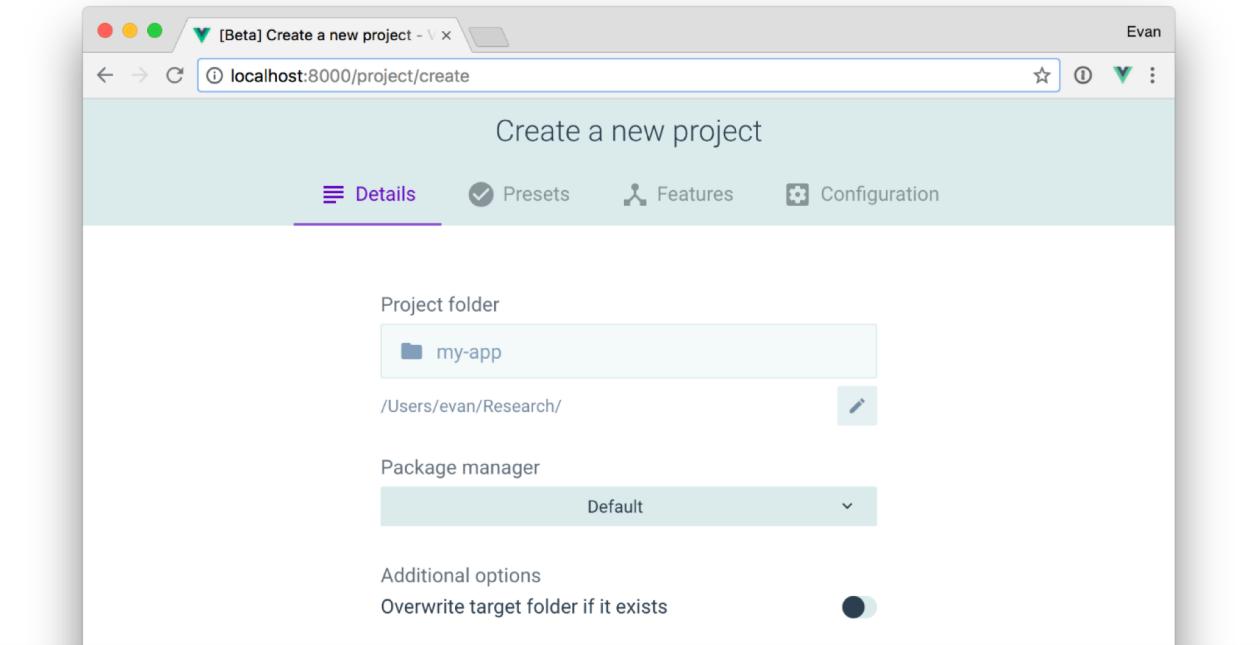
Using the GUI

You can also create and manage projects using a graphical interface with the `vue ui` command:

```
vue ui
```

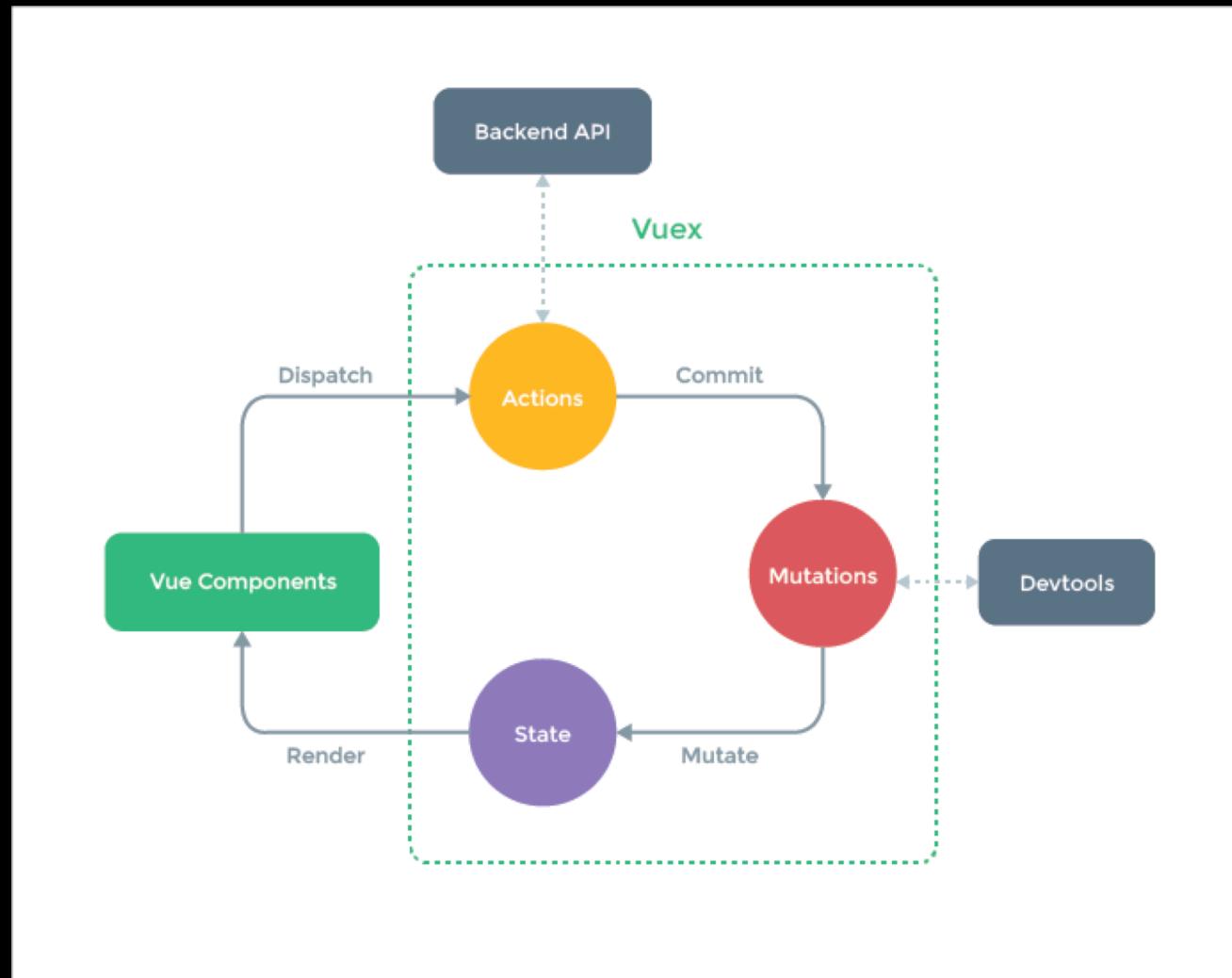
sh

The above command will open a browser window with a GUI that guides you through the project creation process.



<https://cli.vuejs.org/guide/creating-a-project.html#using-the-gui>

vuex



<https://vuex.vuejs.org/fr/>

vue-router

HTML

```
<script src="https://unpkg.com/vue/dist/vue.js"></script>
<script src="https://unpkg.com/vue-router/dist/vue-router.js"></script>

<div id="app">
  <h1>Hello App!</h1>
  <p>
    <!-- use router-link component for navigation. -->
    <!-- specify the link by passing the `to` prop. -->
    <!-- `<router-link>` will be rendered as an `<a>` tag by default -->
    <router-link to="/foo">Go to Foo</router-link>
    <router-link to="/bar">Go to Bar</router-link>
  </p>
  <!-- route outlet -->
  <!-- component matched by the route will render here -->
  <router-view></router-view>
</div>
```

html

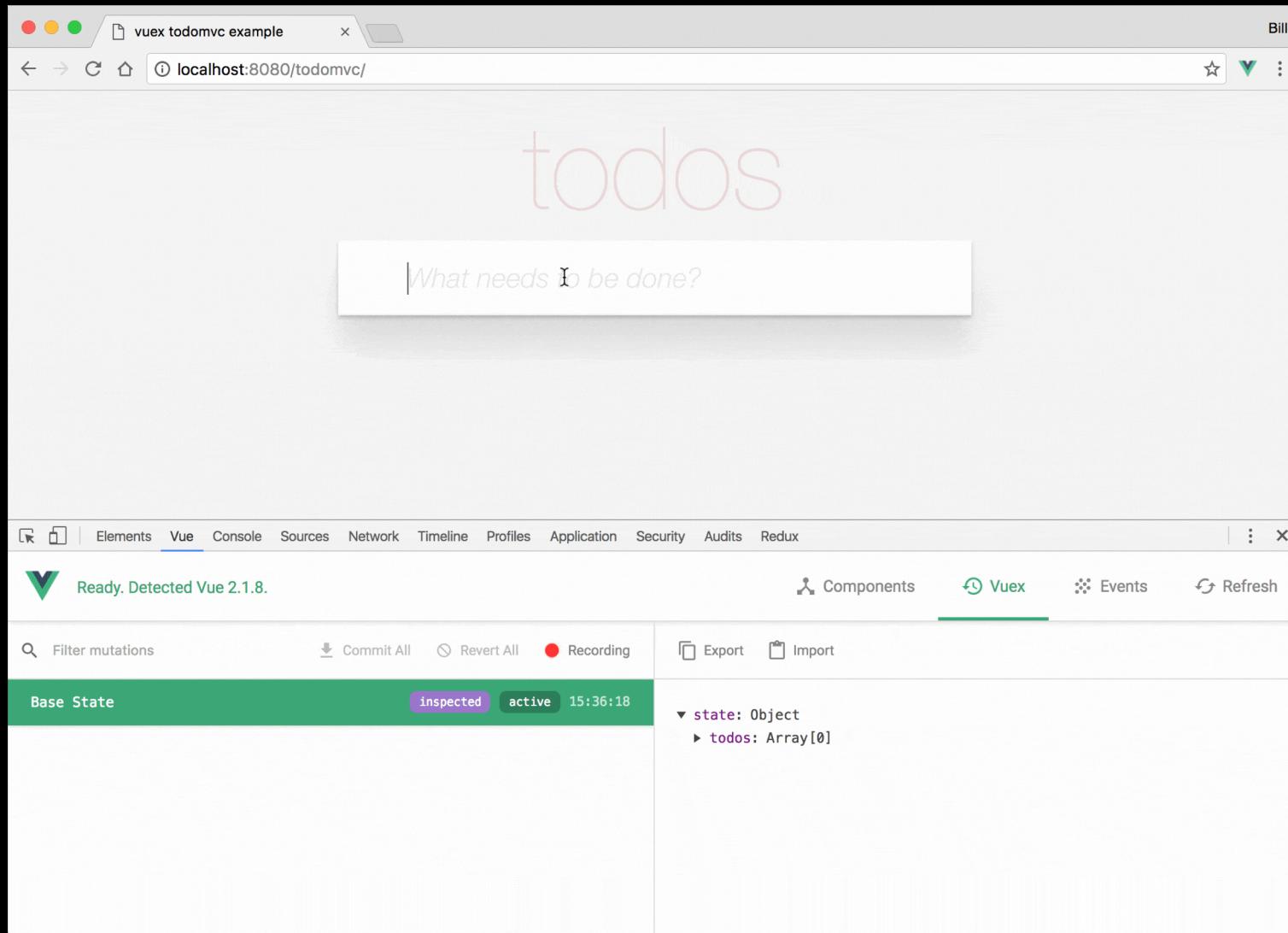
vue-test-utils

```
// test.js  
  
// Import the `mount()` method from the test utils  
// and the component you want to test  
import { mount } from '@vue/test-utils'  
import Counter from './counter'  
  
// Now mount the component and you have the wrapper  
const wrapper = mount(Counter)  
  
// You can access the actual Vue instance via `wrapper.vm`  
const vm = wrapper.vm  
  
// To inspect the wrapper deeper just log it to the console  
// and your adventure with the Vue Test Utils begins  
console.log(wrapper)
```

js

<https://vue-test-utils.vuejs.org>

vue-devtools



<https://github.com/vuejs/vue-devtools>

3. Les bibliothèques non officielles

jest

 **Jest 23.6**

Docs API Help Blog Akses English Search GitHub

Jest

 Delightful JavaScript Testing

TRY OUT JEST GET STARTED WATCH TALKS LEARN MORE

★ Star 21,148



Developer Ready

Complete and ready to set-up JavaScript testing solution. Works out of the box for any React project.



Instant Feedback

Fast interactive watch mode runs only test files related to changed files and is optimized to give signal quickly.



Snapshot Testing

Capture snapshots of React trees or other serializable values to simplify testing and to analyze how state changes over time.

<https://jestjs.io>

vuelidate



Simple, lightweight model-based validation for `Vue.js 2.0`

- Model based
- Decoupled from templates
- Dependency free, minimalistic library
- **Support for collection validations**
- **Support for nested models**

- **Contextified validators**
- Easy to use with custom validators (e.g. Moment.js)
- Support for function composition
- Validates different data sources: Vuex getters, computed values, etc.
- High test coverage

<https://monterail.github.io/vuelidate/>

axios

JS

```
new Vue({
  el: '#app',
  data () {
    return {
      info: null
    }
  },
  mounted () {
    axios
      .get('https://api.coindesk.com/v1/bpi/currentprice.json')
      .then(response => (this.info = response))
  }
})
```

HTML

```
<div id="app">
  {{ info }}
</div>
```

<https://vuejs.org/v2/cookbook/using-axios-to-consume-apis.html>

vue-i18n

HTML

```
<script src="https://unpkg.com/vue/dist/vue.js"></script>
<script src="https://unpkg.com/vue-i18n/dist/vue-i18n.js"></script>

<div id="app">
  <p>{{ $t("message.hello") }}</p>
</div>
```

html

JavaScript

```
// If using a module system (e.g. via vue-cli), import Vue and VueI18n and then call Vue
// import Vue from 'vue'
// import VueI18n from 'vue-i18n'
//
// Vue.use(VueI18n)

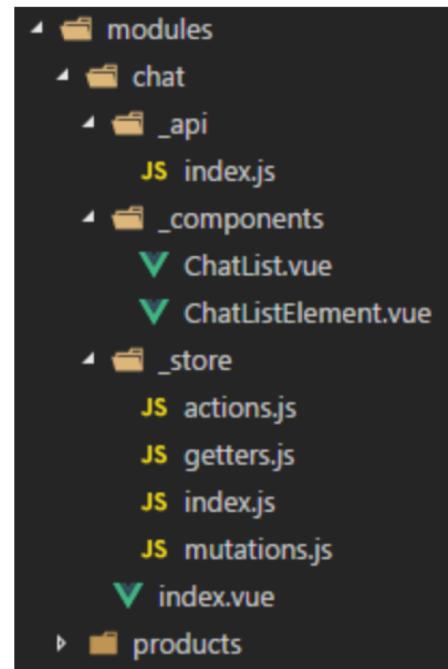
// Ready translated locale messages
const messages = {
  en: {
    message: {
      hello: 'hello world'
    }
  },
  ja: {
```

js

4. Comment architecturer son app?

Large-scale Vuex application structures – par Kevin Peters

inside the modules directory, we have the modules of this *Vue.js* application. It looks something like this:



Inside the modules directory

<https://medium.com/3yourmind/large-scale-vuex-application-structures-651e44863e2f>

vue-enterprise-boilerplate – par Chris Fritz

Vue Enterprise Boilerplate

 PASSED

This is an ever-evolving, very opinionated architecture and dev environment for new Vue SPA projects using [Vue CLI 3](#). Questions, feedback, and for now, even bikeshedding are welcome. 😊 If you'd like to increase the time I can spend on this project, as well as other Vue resources, please consider becoming a [sponsor on Patreon](#). 🙏

Features

- **Thorough documentation:** Written with the same care as Vue's core docs to quickly train new team members and consolidate knowledge.
- **Guaranteed consistency:** Opinionated linting for Vue, JavaScript/JSON, SCSS, and Markdown, integrated into Visual Studio Code and run against staged files on pre-commit.
- **First-class tests:** Practice test-driven development with both unit and end-to-end tests. Unit tests with Jest live as first-class citizens alongside your source files, while Cypress provides reliable end-to-end tests in an intuitive GUI for development.
- **Speedy development:** Between configurable generators, handy aliases, and global base components, your productivity will skyrocket.

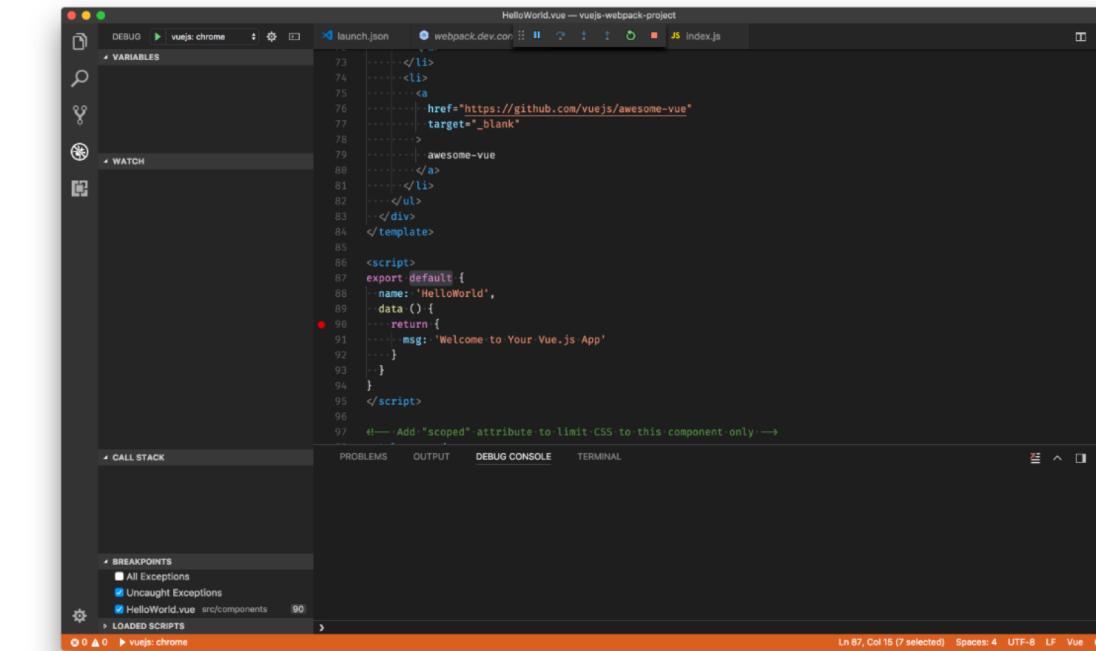
<https://github.com/chrisvfritz/vue-enterprise-boilerplate>

5. Le livre de recettes officiel

Tutoriels officiels

Setting a Breakpoint

1. Set a breakpoint in `src/components/HelloWorld.vue` on `line 90` where the `data` function returns a string.



The screenshot shows the VS Code interface with the "DEBUG" tab selected. The main editor area displays the code for `HelloWorld.vue`. A red dot indicates a breakpoint is set on line 90, which contains the `return` statement of the `data` function. The sidebar on the left shows the "WATCH" and "BREAKPOINTS" sections. The "BREAKPOINTS" section has a checkbox for "HelloWorld.vue" checked. The bottom status bar shows the file is "vuejs: chrome".

```
<template>
<ul>
  <li>
    <a href="https://github.com/vuejs/awesome-vue" target="_blank">
      awesome-vue
    </a>
  </li>
</ul>
</div>
</template>

<script>
export default {
  name: 'HelloWorld',
  data () {
    return {
      msg: 'Welcome to Your Vue.js App'
    }
  }
}
</script>
<!-- Add "scoped" attribute to limit CSS to this component only -->
```

<https://fr.vuejs.org/v2/cookbook/>

6. BONUS

Le Slack officiel de VueMtl

<http://bit.ly/vuemtl-slack1>