



VUE JS

THE ULTIMATE FRONTEND FRAMEWORK

\$whoami.

Xavi Sánchez Mir

Fullstack developer at Tokiota

Universitat Politècnica de Catalunya

Universidad Politécnica de Madrid

 @xavism9



about tokiota.

Software company in bcn & madrid

Microsoft technologies

4 departaments	Development
	Infrastructure
	Sharepoint
	Big Data

70+ in Barcelona, 20+ in Madrid

 @tokiotait

agenda.



START

composing with
components

data, props and methods

01

02

03

04

what is vuejs and cli

general syntax

agenda.



handling user
inputs

template syntax

05

06

07

08

loops and conditionals

lifecycle hooks

agenda.



agenda.



hands on 🎉

13

14

ROSES

dev tooling

01. what is vue js and cli.



Vue is a **progressive framework** for building user interfaces.

The core library is focused on the **view layer only**, and is easy to integrate

Vue is also perfectly capable of powering sophisticated **Single-Page Applications**.

01. what is vue js and cli.



cli means **command line interface**, it allows us to create vuejs applications out of the box, without losing time in the configuration.

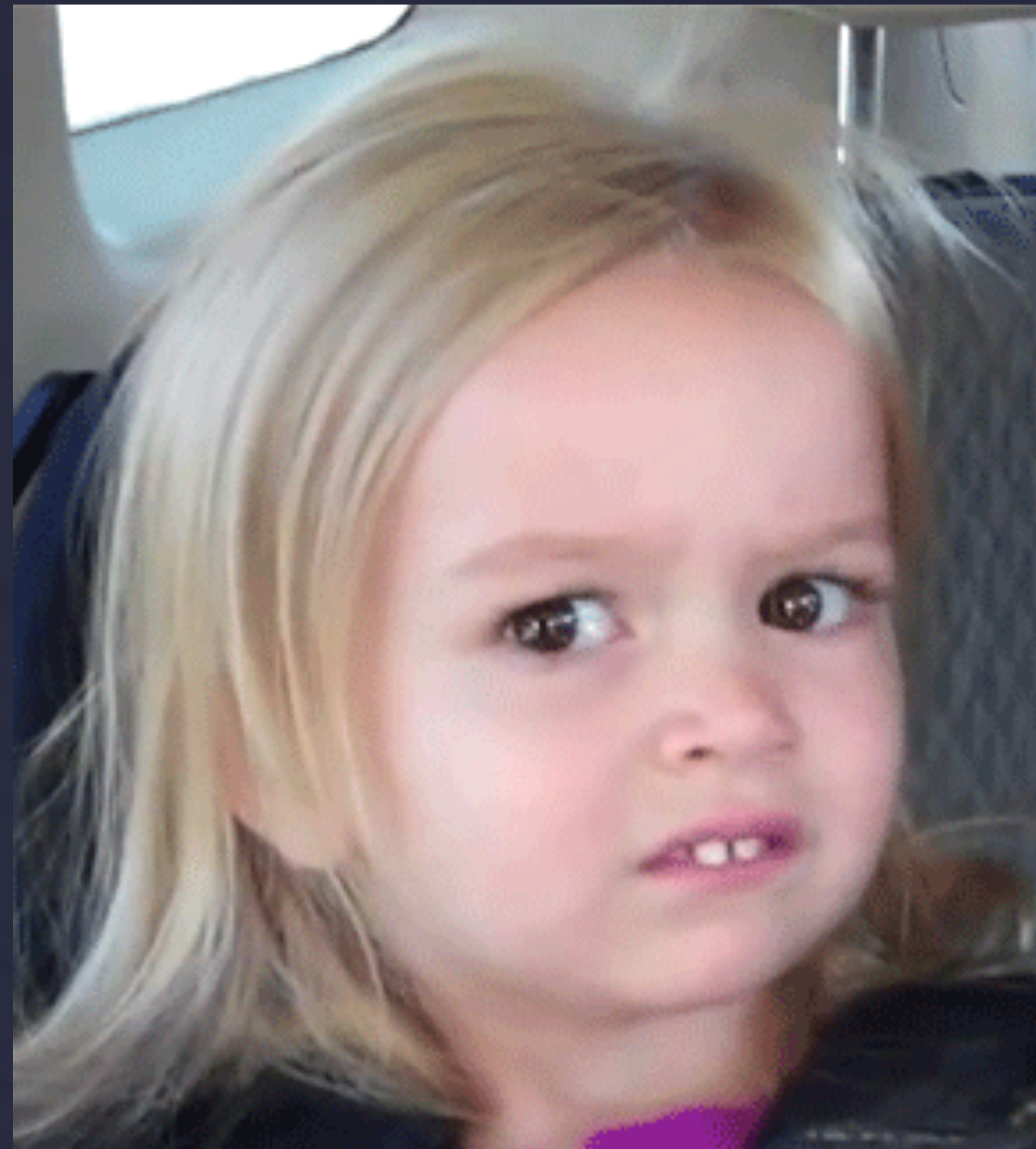
```
npm install @vue/cli
```

```
vue create project-name
```

A terminal window showing the Vue CLI v3.0.0-beta.1 installation process. The window title is "Projects — vue create myproject — vue — node /usr/local/bin/vue...". The terminal output shows the following prompts and options:

```
Vue CLI v3.0.0-beta.1
? Please pick a preset: Manually select features
? Check the features needed for your project: (Press <space> to select
, <a> to toggle all, <i> to invert selection)
> ☐ TypeScript
  ☐ Progressive Web App (PWA) Support
  ☐ Router
  ☐ Vuex
  ☐ CSS Pre-processors
  ☐ Linter / Formatter
  ☐ Unit Testing
  ☐ E2E Testing
```

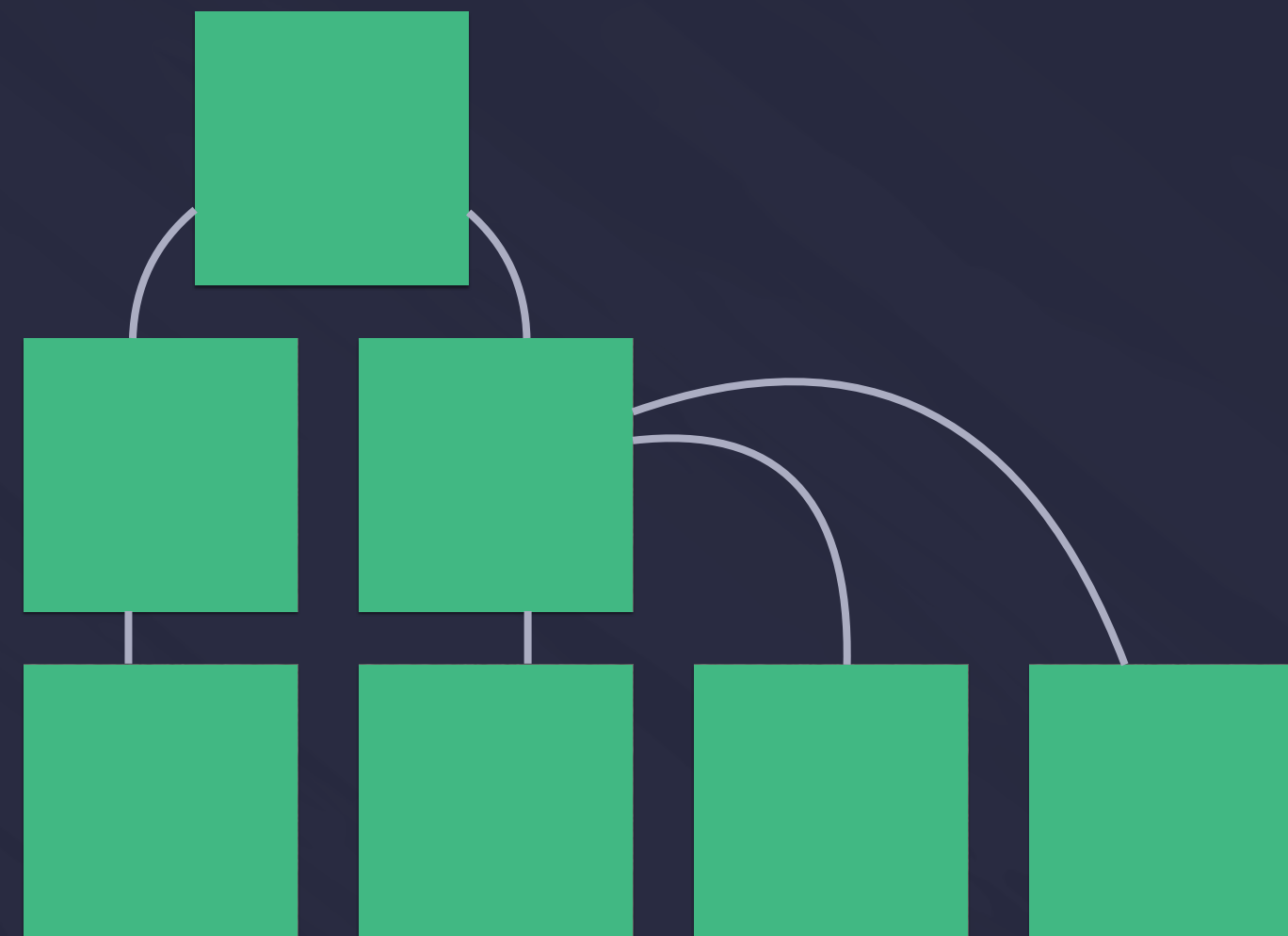
01. what is vue js and cli.



02. composing with components.



Component-based, allows us to transform large apps into **small, self-contained** and often **reusable** components.



03. general syntax.



```
<template>
  <div class="name">
  </div>
</template>
```

template

```
<script>
export default {
  name: 'ComponentName',
  props: {
    msg: String
  }
}
</script>
```

scripts

```
<style scoped>
h3 {
  margin: 40px 0 0;
}
</style>
```

style

04. data, props and methods.



```
export default {  
  name: 'HelloWorld',  
  data() {  
    return {  
      message: 'Hi!'  
    }  
  },  
  props: {  
    msg: String  
  },  
  methods: {  
    sayHi() {  
      console.log('Hi')  
    }  
  }  
}
```

05. loops and conditionals.



```
<li v-for="(item, index) in items" :key="index"></li>
```

```
<h1 v-if="ok">Yes</h1>  
<h1 v-else>No</h1>
```


06. handling user inputs.



```
<button v-on:click="reverseMessage">Reverse Message</button>
```

`v-on => @`

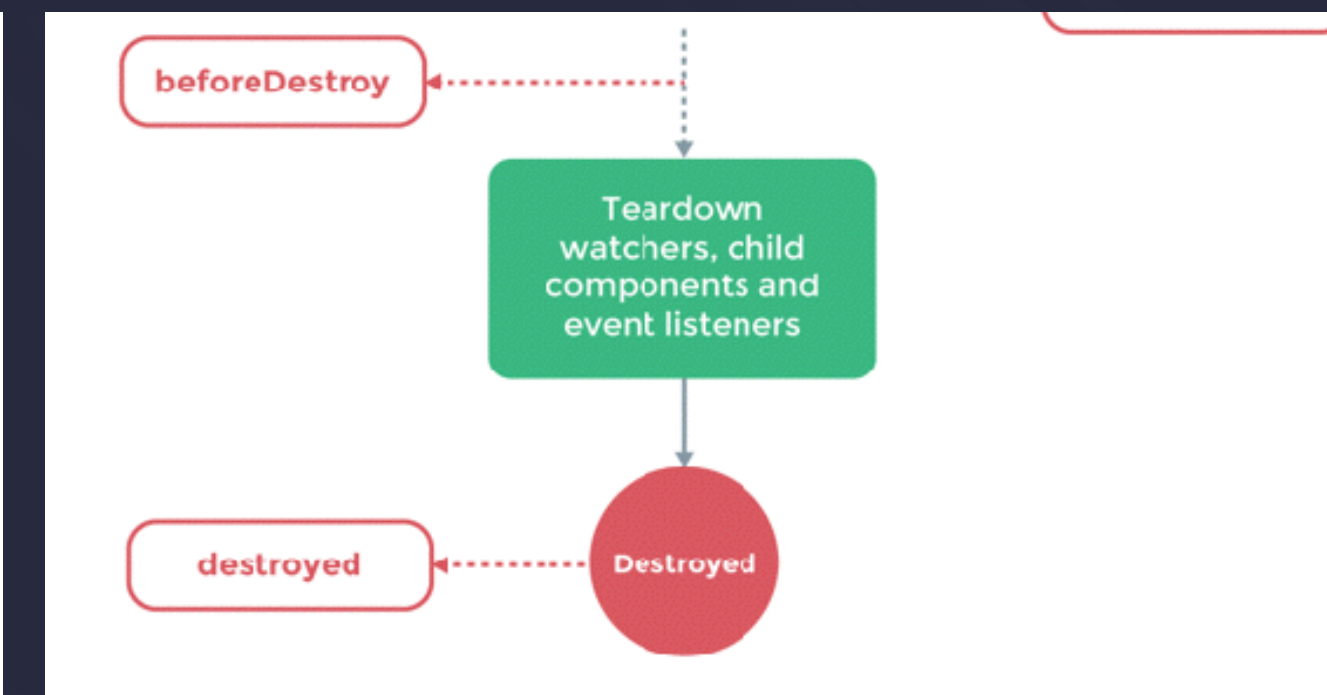
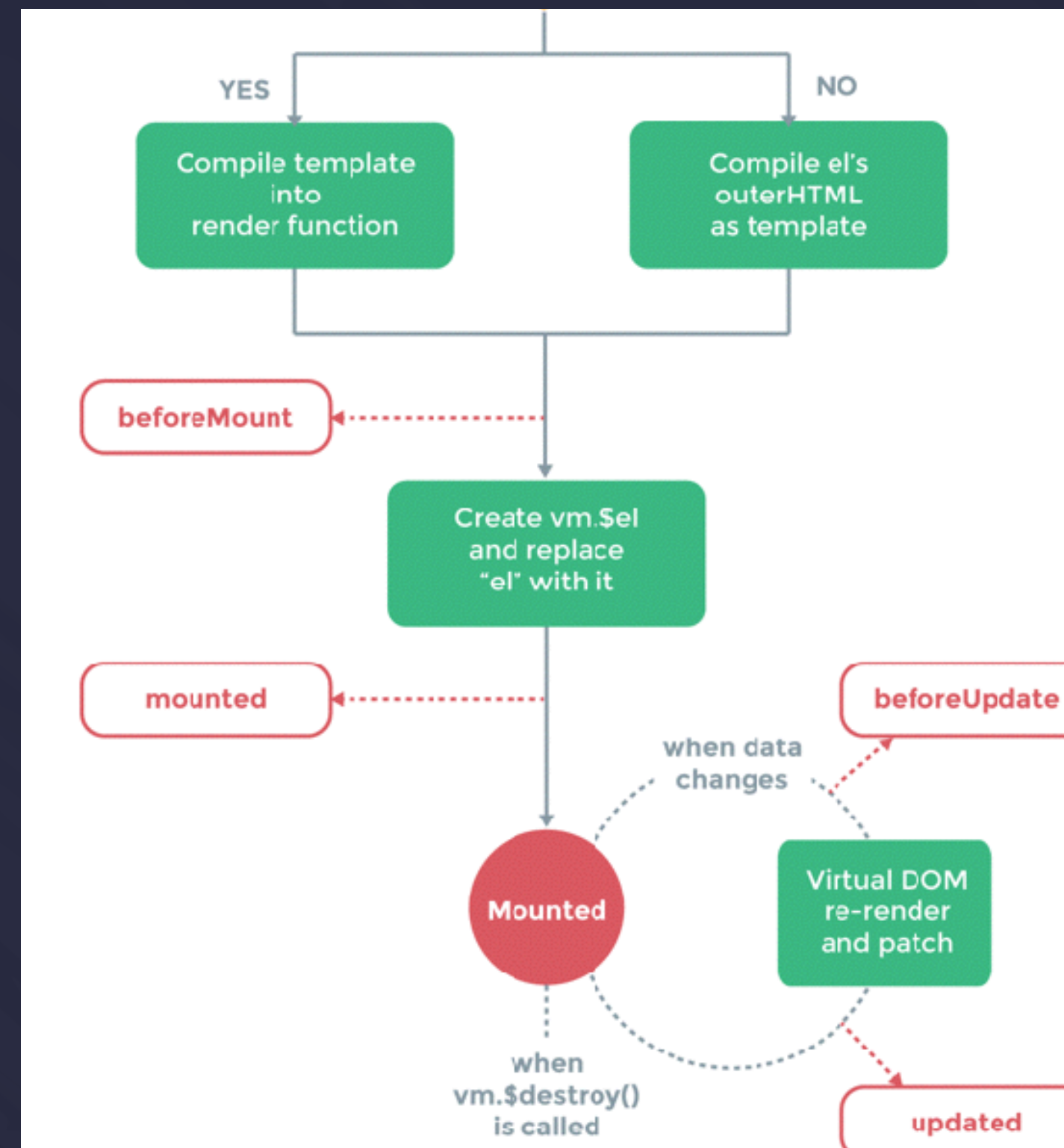
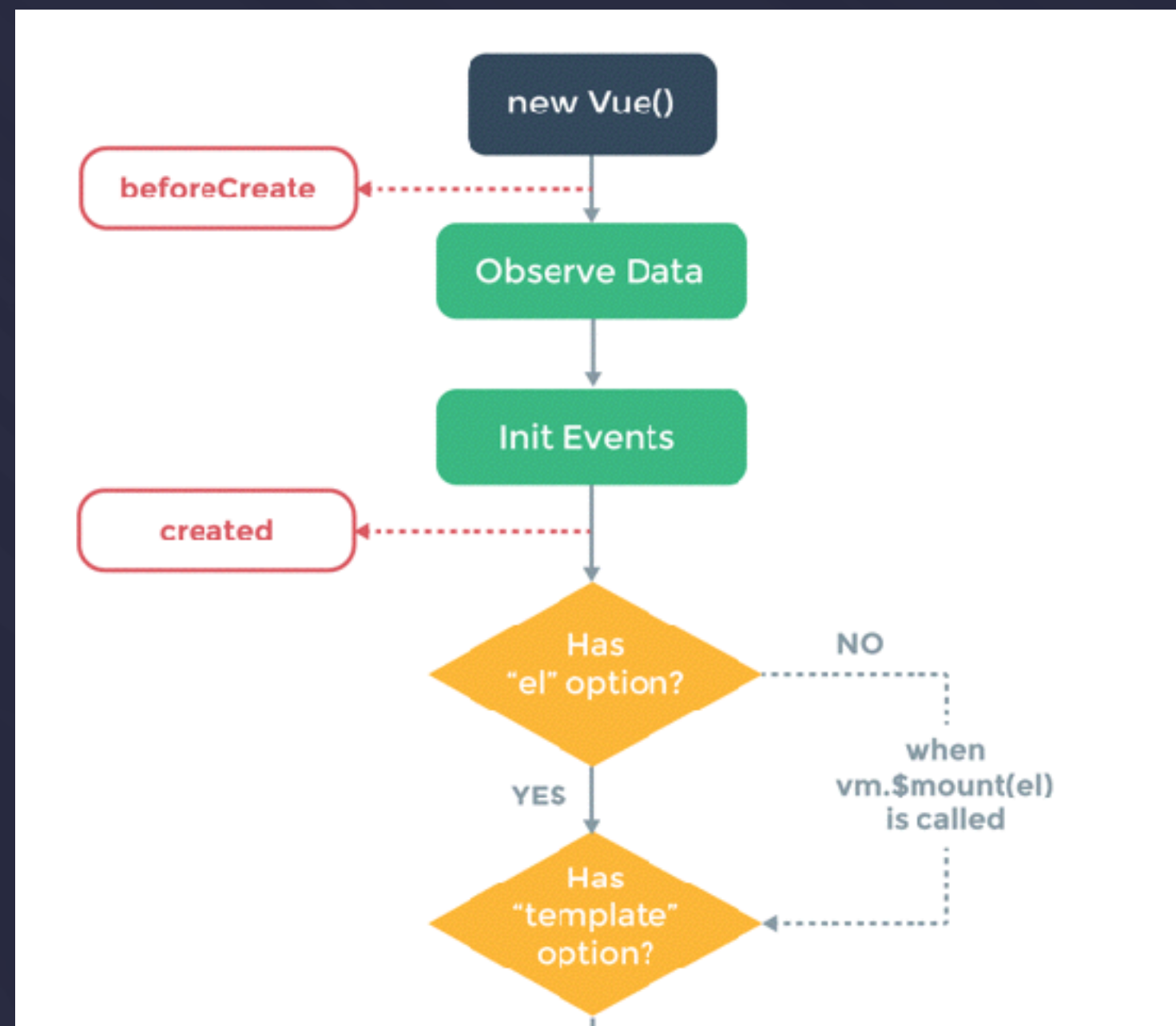
`@click` (click.once, click.stop, click.capture)

`@hover`

`@keyup, @keydown, @keypressed` (keyup.13, keyup.ctrl, key.enter ...)

`v-on:ownEvent="doWhatever"`

07. lifecycle hooks.



08. template syntax.



```
<p>Using mustaches: {{ rawHtml }}</p>  
<p>Using v-html directive: <span v-html="rawHtml"></span></p>  
<p v-if="conditional"></p>
```

```
<a v-bind:href="url"> ... </a>  
<a :href="url"> ... </a>
```

09. directives.



Directives are special attributes with the `v-` prefix. Directive attribute values are expected to be **a single JavaScript expression**.

There are a set of created directives to use, but you can also create your own ones.

<https://vuejs.org/v2/guide/custom-directive.html>

10. computed.



```
<div id="example">
  <p>Original message: "{{ message }}"</p>
  <p>Computed reversed message: "{{ reversedMessage }}"</p>
</div>

<script>
computed: {
  // a computed getter
  reversedMessage: function () {
    // `this` points to the vm instance
    return this.message.split('').reverse().join('')
  }
}
</script>
```

11. watchers.



```
<div id="example">
  <p>Original message: "{{ message }}"</p>
</div>

<script>
watch: {
  message: function () {
    console.log('Your message has changed')
  }
}
</script>
```

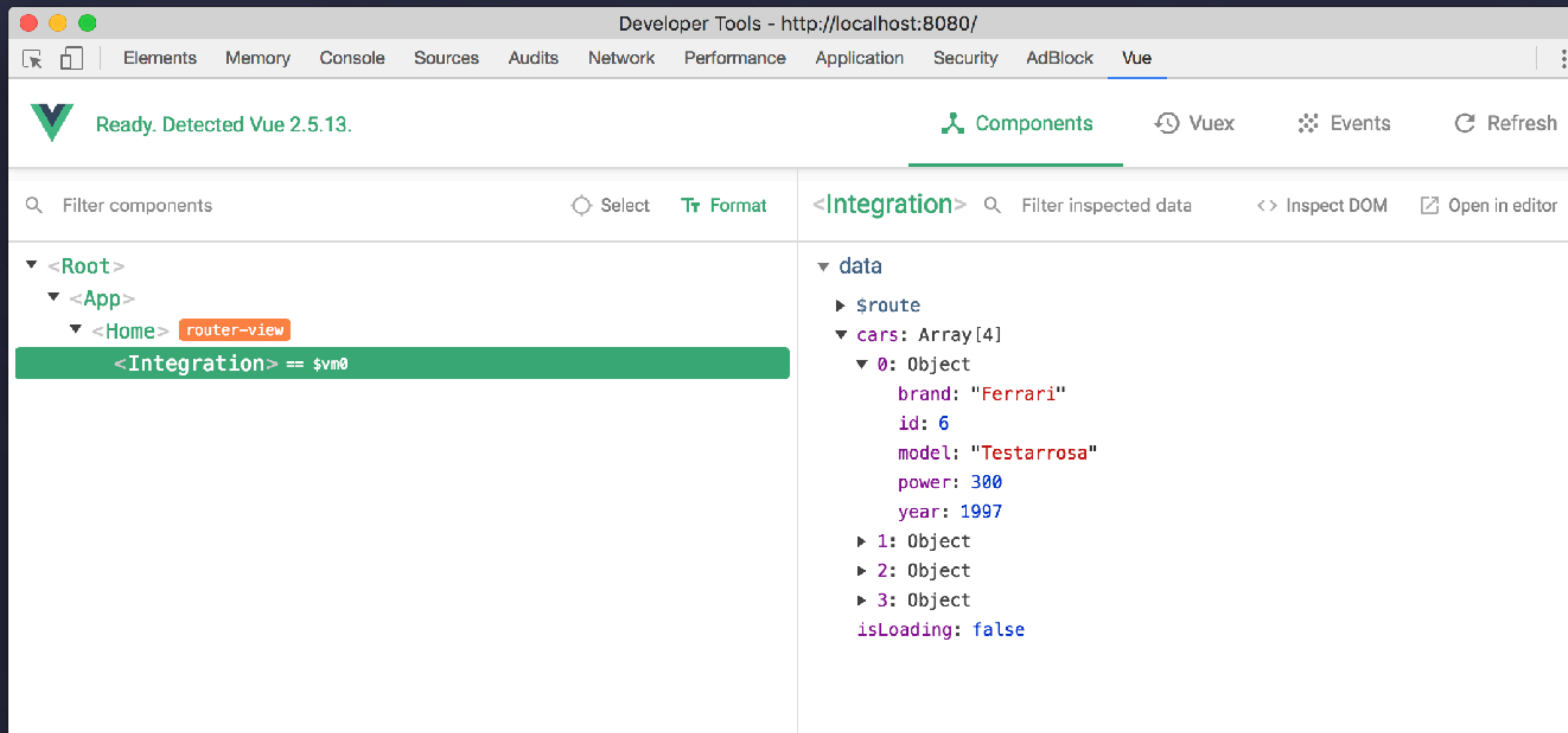

12. routing.



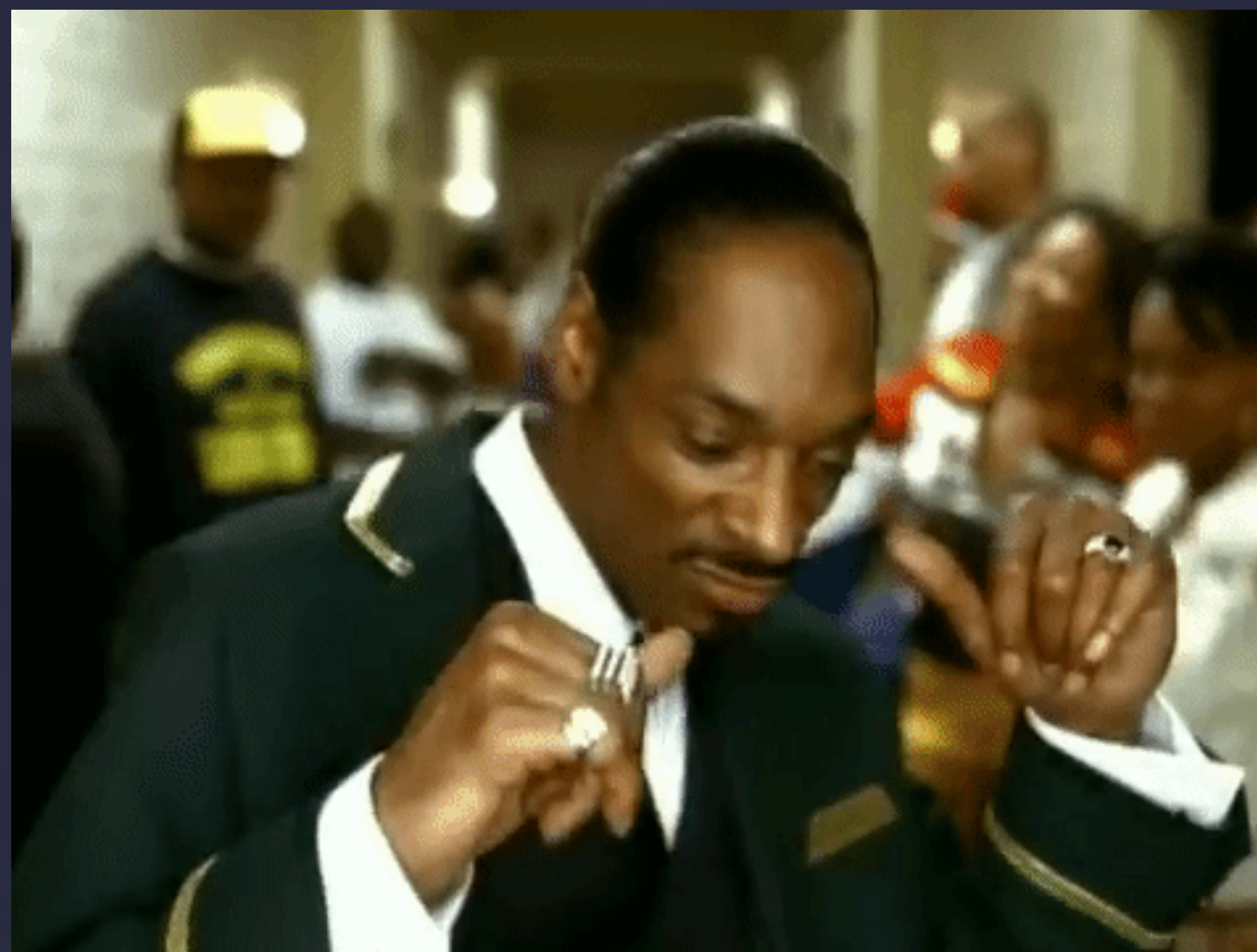
Create an **SPA** as simple as possible, just enjoy how it works

```
const routes = {  
  '/': Home,  
  '/about': About  
}
```

13. dev tooling.



14. hands on. 🎉



thanks for your **attention.** 🙌🌹