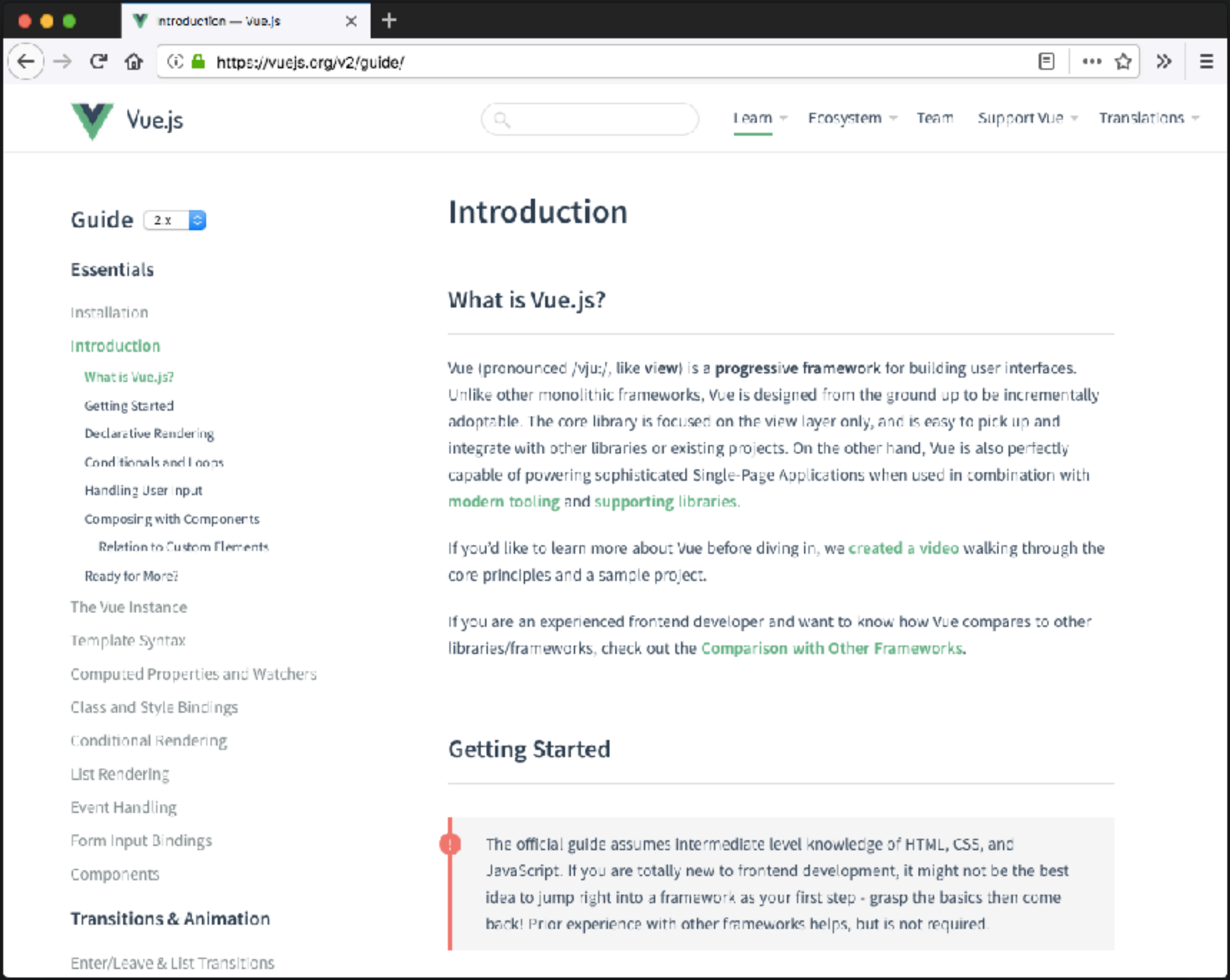
7 secret patterns Vue consultants don’t want you to know!

By 1 member of the Vue core team
you’ll never believe is shorter in real life!

Me

Me



Introduction

What is Vue.js?

Vue (pronounced /vjuː/, like **view**) is a **progressive framework** for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries.

If you'd like to learn more about Vue before diving in, we created a video walking through the core principles and a sample project.

If you are an experienced frontend developer and want to know how Vue compares to other libraries/frameworks, check out the Comparison with Other Frameworks.

Getting Started

The official guide assumes intermediate level knowledge of HTML, CSS, and JavaScript. If you are totally new to frontend development, it might not be the best idea to jump right into a framework as your first step - grasp the basics then come back! Prior experience with other frameworks helps, but is not required.

Secrets

No need to take notes!

Productivity Boosts

Radical Tweaks

Unlocked Possibilities

Productivity Boost #1

Smarter Watchers

```
created() {
  this.fetchUserList()
},
watch: {
  searchText() {
    this.fetchUserList()
  }
}
```

Fetch on created, then watch

Smarter Watchers

```
created() {
  this.fetchUserList()
},
watch: {
  searchText: 'fetchUserList'
}
```

Watchers accept method names

Smarter Watchers

```
watch: {
  searchText: {
    handler: 'fetchUserList',
    immediate: true
  }
}
```

And can call *themselves* on created

Productivity Boost #2

Component Registration

```
import BaseButton from './base-button'
import BaseIcon from './base-icon'
import BaseInput from './base-input'

export default {
  components: {
    BaseButton,
    BaseIcon,
    BaseInput
  }
}
```

Frequently imported components

Component Registration

```js
import BaseButton from './base-button'
import BaseIcon from './base-icon'
import BaseInput from './base-input'

export default {
  components: {
    BaseButton,
    BaseIcon,
    BaseInput
  }
}
```

```html
<BaseInput
  v-model="searchText"
  @keydown.enter="search"
/>
<BaseButton @click="search">
  <BaseIcon name="search"/>
</BaseButton>
```

So much JS to support such a small template!

Component Registration

```
import Vue from 'vue'
import upperFirst from 'lodash/upperFirst'
import camelCase from 'lodash/camelCase'

// Require in a base component context
const requireComponent = require.context(
  '.', false, /base-[\w-]+\.vue$/
)

requireComponent.keys().forEach(fileName => {
  // Get component config
  const componentConfig = requireComponent(fileName)

  // Get PascalCase name of component
  const componentName = upperFirst(
    camelCase(fileName.replace(/^\.\//, '').replace(/\.\w+$/, ''))
  )

  // Register component globally
  Vue.component(componentName, componentConfig.default || componentConfig)
})
```

Component Registration

```
import BaseButton from './base-button'
import BaseIcon from './base-icon'
import BaseInput from './base-input'

export default {
  components: {
    BaseButton,
    BaseIcon,
    BaseInput
  }
}
```

```
<BaseInput
  v-model="searchText"
  @keydown.enter="search"
/>
<BaseButton @click="search">
  <BaseIcon name="search"/>
</BaseButton>
```

Component Registration

```
<BaseInput
  v-model="searchText"
  @keydown.enter="search"
/>
<BaseButton @click="search">
  <BaseIcon name="search"/>
</BaseButton>
```

Productivity Boost #3

Module Registration

```
import auth from './modules/auth'
import posts from './modules/posts'
import comments from './modules/comments'
// ...

export default new Vuex.Store({
  modules: {
    auth,
    posts,
    comments,
    // ...
  }
})
```

Tedious!

Module Registration

```
import camelCase from 'lodash/camelCase'
const requireModule = require.context('.', false, /\.js$/)
const modules = {}

requireModule.keys().forEach(fileName => {
  // Don't register this file as a Vuex module
  if (fileName === './index.js') return

  const moduleName = camelCase(
    fileName.replace(/(\.\/|\.js)/g, '')
  )
  modules[moduleName] = requireModule(fileName)
})

export default modules
```

Module Registration

```
import camelCase from 'lodash/camelCase'
const requireModule = require.context('.', false, /\.js$/)
const modules = {}

requireModule.keys().forEach(fileName => {
  // Don't register this file as a Vuex module
  if (fileName === './index.js') return

  const moduleName = camelCase(
    fileName.replace(/(\.\/|\.js)/g, '')
  )
  modules[moduleName] = {
    namespaced: true,
    ...requireModule(fileName),
  }
})

export default modules
```

Module Registration

```js
import auth from './modules/auth'
import posts from './modules/posts'
import comments from './modules/comments'
// ...

export default new Vuex.Store({
  modules: {
    auth,
    posts,
    comments,
    // ...
  }
})
```

Say goodbye!

Module Registration

```
import modules from './modules'

export default new Vuex.Store({
  modules
})
```

Secrets

☑ Productivity Boosts

Radical Tweaks

Unlocked Possibilities

Radical Tweak #1

Cleaner Views

```
data() {
  return {
    loading: false,
    error: null,
    post: null
  }
},
watch: {
  '$route': {
    handler: 'resetData',
    immediate: true
  }
},
methods: {
  resetData() {
    this.loading = false
    this.error = null
    this.post = null
    this.getPost(this.$route.params.id)
  },
  getPost(postId) {
    // ...
  }
}
```

So much setup!

Cleaner Views

```
data() {
  return {
    loading: false,
    error: null,
    post: null
  }
},
created () {
  this.getPost(this.$route.params.id)
},
methods: {
  getPost(postId) {
    // ...
  }
}
```

Radical Tweak #2

Transparent Wrappers

```
<template>
  <input
    :value="value"
    @input="$emit('input', $event.target.value)"
  >
</template>
```

Transparent Wrappers

```
<template>
  <label>
    {{ label }}
    <input
      :value="value"
      @input="$emit('input', $event.target.value)"
    >
  </label>
</template>
```

```
<BaseInput @focus.native="doSomething">
```

❗

Transparent Wrappers

```
<template>
  <label>
    {{ label }}
    <input
      :value="value"
      v-on="listeners"
    >
  </label>
</template>
```

```
computed: {
  listeners() {
    return {
      ...this.$listeners,
      input: event =>
        this.$emit('input', event.target.value)
    }
  }
}
```

```
<BaseInput @focus.native="doSomething">
```

Transparent Wrappers

```
<template>
  <label>
    {{ label }}
    <input
      :value="value"
      v-on="listeners"
    >
  </label>
</template>
```

```
computed: {
  listeners() {
    return {
      ...this.$listeners,
      input: event =>
        this.$emit('input', event.target.value)
    }
  }
}
```

```
<BaseInput @focus.native="doSomething">
```

Transparent Wrappers

```
<template>
  <label>
    {{ label }}
    <input
      :value="value"
      v-on="listeners"
    >
  </label>
</template>
```

```
computed: {
  listeners() {
    return {
      ...this.$listeners,
      input: event =>
        this.$emit('input', event.target.value)
    }
  }
}
```

```
<BaseInput @focus="doSomething">
```

Transparent Wrappers

```
<template>
  <label>
    {{ label }}
    <input
      :value="value"
      v-on="listeners"
    >
  </label>
</template>
```

```
computed: {
  listeners() {
    return {
      ...this.$listeners,
      input: event =>
        this.$emit('input', event.target.value)
    }
  }
}
```

```
<BaseInput @focus="doSomething">
```

Transparent Wrappers

```
<template>
  <label>
    {{ label }}
    <input
      :value="value"
      v-on="listeners"
    >
  </label>
</template>
```

```
<BaseInput
  placeholder="What's your name?"
  @focus="doSomething"
/>
```

Transparent Wrappers

```
<template>
  <label>
    {{ label }}
    <input
      v-bind=$attrs"
      :value="value"
      v-on="listeners"
    >
  </label>
</template>
```

```
inheritAttrs: false
```

😄

Fully transparent wrappers!

```
<BaseInput
  placeholder="What's your name?"
  @focus="doSomething"
/>
```

✓

Secrets

☑ Productivity Boosts

☑ Radical Tweaks

Unlocked Possibilities

Unlocked Possibility #1

Single-Root Components

```
<template>
  <ul>
    <NavBarRoutes :routes="persistentNavRoutes"/>
    <NavBarRoutes
      v-if="loggedIn"
      :routes="loggedInNavRoutes"
    />
    <NavBarRoutes
      v-else
      :routes="loggedOutNavRoutes"
    />
  </ul>
</template>
```

Single-Root Components

```
<template>
  <ul>
    <NavBarRoutes :routes="persistentNavRoutes"/>
    <NavBarRoutes
      v-if="loggedIn"
      :routes="loggedInNavRoutes"
    />
    <NavBarRoutes
      v-else
      :routes="loggedOutNavRoutes"
    />
  </ul>
</template>
```

```
<template>
  <li
    v-for="route in routes"
    :key="route.name"
  >
    <router-link :to="route">
      {{ route.title }}
    </router-link>
  </li>
</template>
```

Single-Root Components

```
<template>
  <ul>
    <NavBarRoutes :routes="persistentNavRoutes"/>
    <NavBarRoutes
      v-if="loggedIn"
      :routes="loggedInNavRoutes"
    />
    <NavBarRoutes
      v-else
      :routes="loggedOutNavRoutes"
    />
  </ul>
</template>
```

```
functional: true,
render(h, { props }) {
  return props.routes.map(route =>
    <li key={route.name}>
      <router-link to={route}>
        {route.title}
      </router-link>
    </li>
  )
}
```

😄

Multi-root components!

Unlocked Possibility #2

Rendering non-HTML

Rendering non-HTML

```
// Init
const map = new mapboxgl.Map(/* ... */)

map.on('load', () => {
  // Data
  map.addSource(/* ... */)

  // Layers
  map.addLayer(/* ... */)
  map.addLayer(/* ... */)

  // Hover effects
  map.on('mousemove', /* ... */)
  map.on('mouseleave', /* ... */)
})
```

Rendering non-HTML

```
<MapboxMap>
  <MapboxMarkers
    :items="cities"
    primary
  >
    <template slot-scope="city">
      <h3>{{ city.name }}</h3>
    </template>
  </MapboxMarkers>
  <MapboxNavigation/>
</MapboxMap>
```

```
// Init
const map = new mapboxgl.Map(/* ... */)

map.on('load', () => {
  // Data
  map.addSource(/* ... */)

  // Layers
  map.addLayer(/* ... */)
  map.addLayer(/* ... */)

  // Hover effects
  map.on('mousemove', /* ... */)
  map.on('mouseleave', /* ... */)
})
```

Rendering non-HTML

```
<MapboxMap>
  <MapboxMarkers
    :items="cities"
    primary
  >
    <template slot-scope="city">
      <h3>{{ city.name }}</h3>
    </template>
  </MapboxMarkers>
  <MapboxNavigation/>
</MapboxMap>
```

```
created() {
  const { map } = this.$parent
  map.on('load', () => {
    map.addSource(/* ... */)
    map.addLayer(/* ... */)
  })
}
beforeDestroy() {
  const { map } = this.$parent
  map.on('load', () => {
    map.removeSource(/* ... */)
    map.removeLayer(/* ... */)
  })
},
render(h) {
  return null
}
```

Rendering non-HTML

```
<template slot-scope="city">
  <h3>{{ city.name }}</h3>
</template>
```

Rendering non-HTML

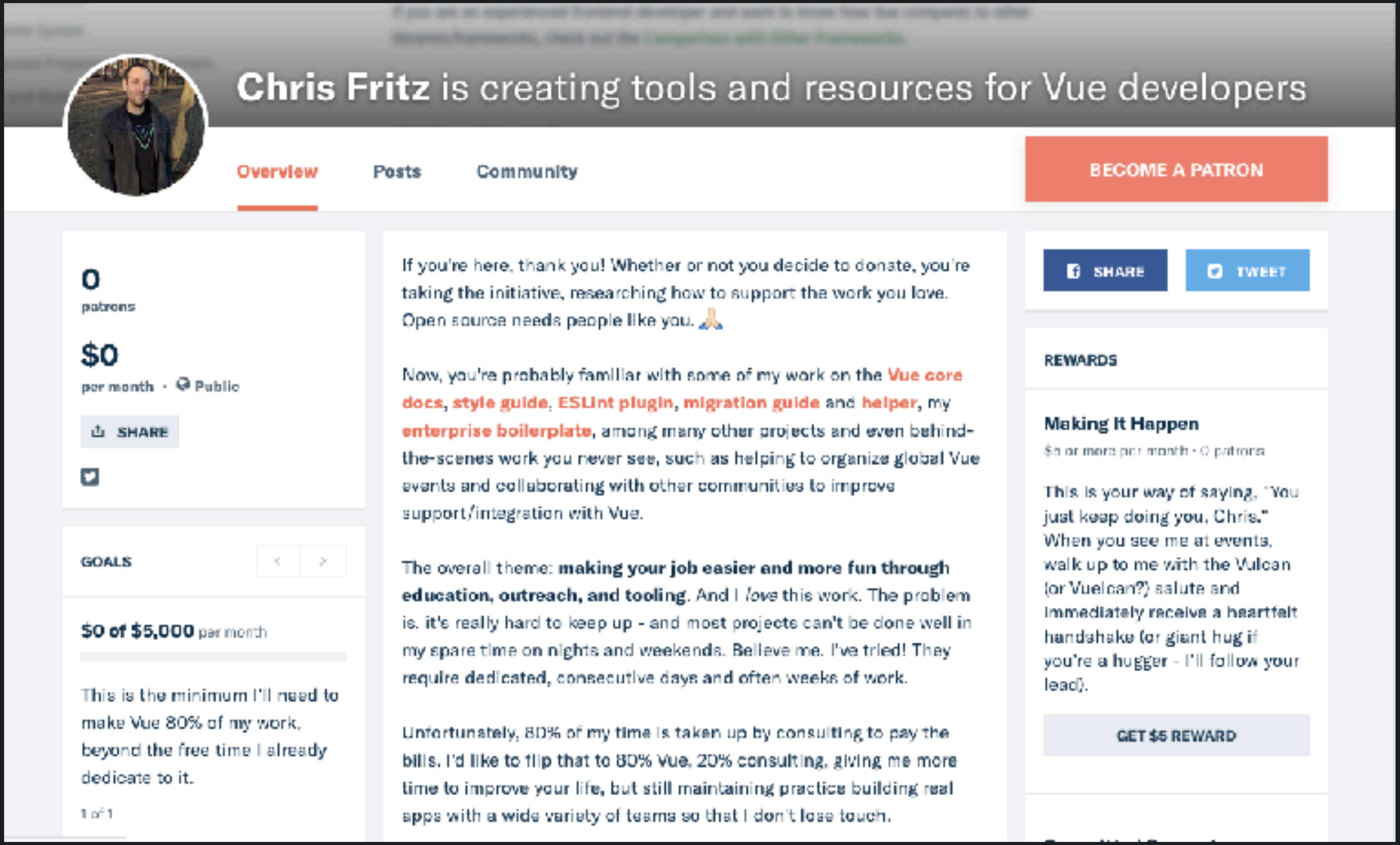
```
<template slot-scope="city">
  <h3>{{ city.name }}</h3>
</template>
```

```
createPopup(marker) {
  this.removeOpenPopups()

  const popupContentVNodes =
    this.$scopedSlots.default(marker.properties)
  const popupHtml = new Vue({
    render: h => h('div', popupContentVNodes)
  }).$mount().$el.innerHTML

  this.lastAddedPopup = new mapboxgl.Popup({
    offset: this.markerOffset
  })
    .setLngLat(marker.geometry.coordinates)
    .setHTML(popupHtml)
    .addTo(this.map)
}
```

Even more patterns!

github.com/chrisvfritz/vue-enterprise-boilerplate

Announcement!



Chris Fritz is creating tools and resources for Vue developers

0
patrons

$0
per month · Public

GOALS

$0 of $5,000 per month

This is the minimum I'll need to make Vue 80% of my work, beyond the free time I already dedicate to it.

1 of 1

If you're here, thank you! Whether or not you decide to donate, you're taking the initiative, researching how to support the work you love. Open source needs people like you. 🙏

Now, you're probably familiar with some of my work on the **Vue core docs**, **style guide**, **ESLint plugin**, **migration guide** and **helper**, my **enterprise boilerplate**, among many other projects and even behind-the-scenes work you never see, such as helping to organize global Vue events and collaborating with other communities to improve support/integration with Vue.

The overall theme: **making your job easier and more fun through education, outreach, and tooling**. And I *love* this work. The problem is, it's really hard to keep up - and most projects can't be done well in my spare time on nights and weekends. Believe me. I've tried! They require dedicated, consecutive days and often weeks of work.

Unfortunately, 80% of my time is taken up by consulting to pay the bills. I'd like to flip that to 80% Vue, 20% consulting, giving me more time to improve your life, but still maintaining practice building real apps with a wide variety of teams so that I don't lose touch.

REWARDS

Making It Happen
$5 or more per month · 0 patrons

This is your way of saying, "You just keep doing you, Chris." When you see me at events, walk up to me with the Vulcan (or Vuelcan?) salute and immediately receive a heartfelt handshake (or giant hug if you're a hugger - I'll follow your lead).

patreon.com/chrisvuefritz

Questions?

github.com/chrisvfritz/7-secret-patterns

@chrisvfritz