

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


Hi!





introduction — Vue.js

https://vuejs.org/v2/guide/



Learn ▾Ecosystem ▾TeamSupport Vue ▾Translations ▾

Guide 2 x

Essentials

Installation

Introduction

What is Vue.js?

Getting Started

Declarative Rendering

Conditionals and Loops

Handling User Input

Composing with Components

Relation to Custom Elements

Ready for More?

The Vue Instance

Template Syntax

Computed Properties and Watchers

Class and Style Bindings

Conditional Rendering

List Rendering

Event Handling

Form Input Bindings

Components

Transitions & Animation

Enter/Leave & List Transitions

# 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

```
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>
```

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(
  './components', 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

```
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

# Lazy-Loaded Routes

```
{  
  path: '/admin',  
  name: 'admin-dashboard',  
  component: require('@views/admin').default  
}
```



# Lazy-Loaded Routes

```
{  
  path: '/admin',  
  name: 'admin-dashboard',  
  component: () => import('@views/admin')  
}
```

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

(Emitted value instead of an instance of Error)

Error compiling template:

```
<div></div>
```

```
<div></div>
```

- Component template should contain exactly one root element.  
If you are using v-if on multiple elements, use v-else-if to chain them instead.

# 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>
  )
}
```

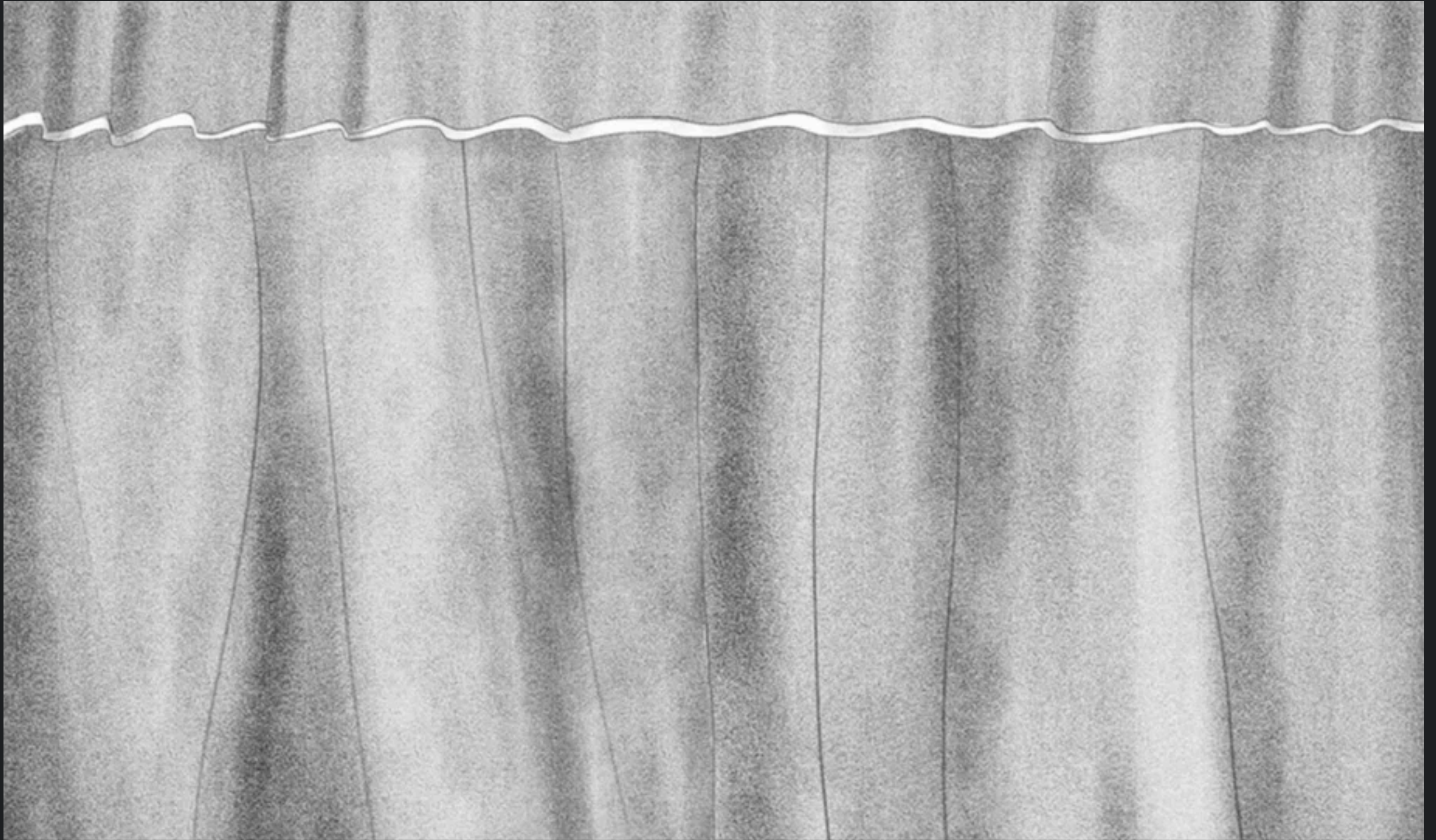


Multi-root components!



Unlocked Possibility #2

# Coordinated Transitions



# Coordinated Transitions

## TheaterCurtain.vue

```
<template>  
  <transition>  
    <div  
      class="curtain"  
      :class="direction"  
    />  
  </transition>  
</template>
```

```
<TheaterCurtain v-if="showCurtain" direction="center"/>  
<TheaterCurtain v-if="showCurtain" direction="left"/>  
<TheaterCurtain v-if="showCurtain" direction="right"/>
```

# Coordinated Transitions

## TheaterCurtain.vue

```
<template>  
  <transition>  
    <div  
      class="curtain"  
      :class="direction"  
    />  
  </transition>  
</template>
```

```
<TheaterCurtains :show="showCurtains"/>
```

# Coordinated Transitions

## TheaterCurtain.vue

```
<template>
  <transition>
    <div
      class="curtain"
      :class="direction"
    />
  </transition>
</template>
```

## TheaterCurtains.vue

```
functional: true,
render (h, { props }) {
  return props.show && [
    <TheaterCurtain direction='center' />
    <TheaterCurtain direction='left' />
    <TheaterCurtain direction='right' />
  ]
}
```

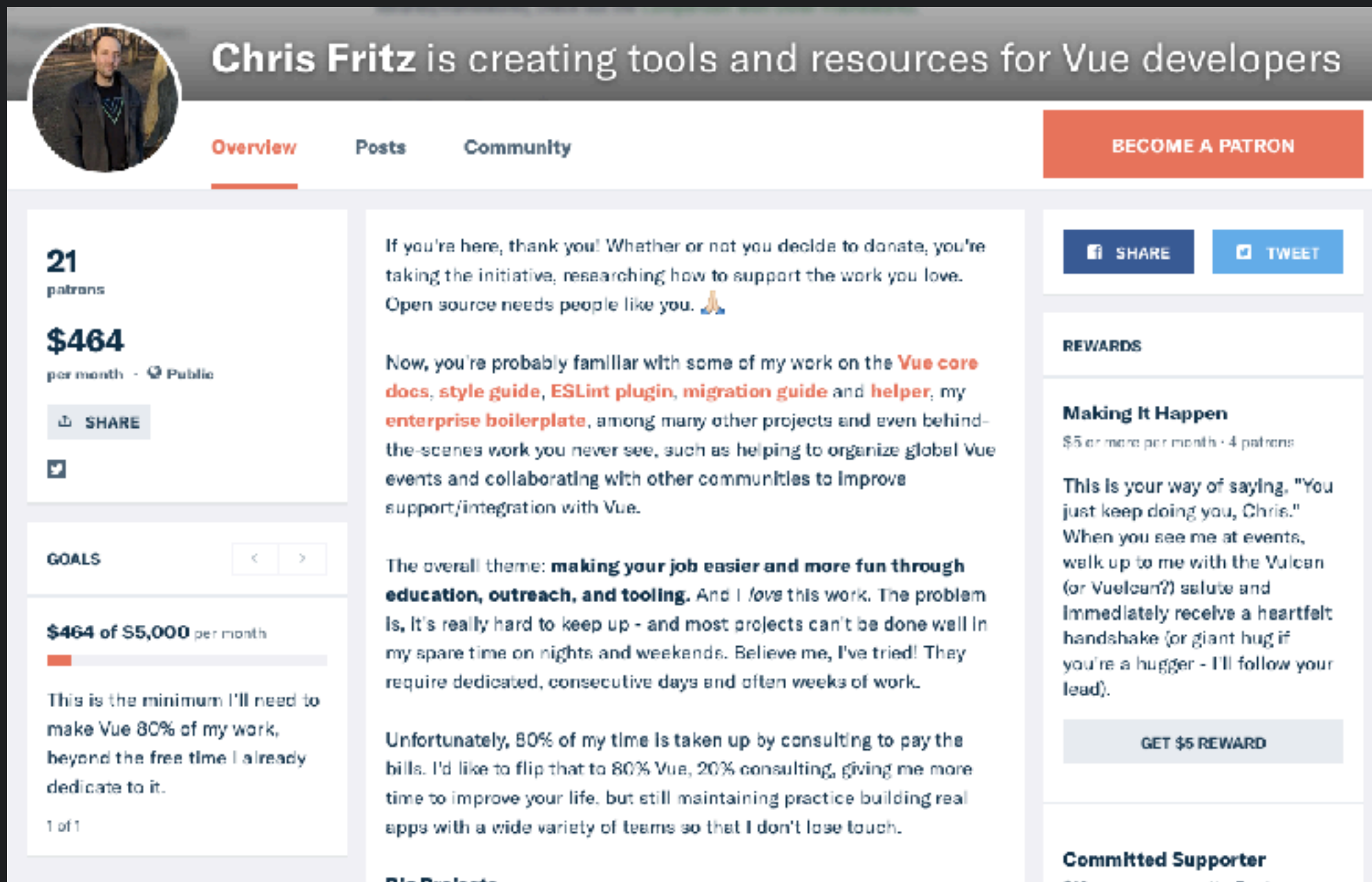
```
<TheaterCurtains :show="showCurtains" />
```

# Can't get enough?

[github.com/chrisvfritz/vue-enterprise-boilerplate](https://github.com/chrisvfritz/vue-enterprise-boilerplate)



# Docs roadmap



The screenshot shows the Patreon profile of Chris Fritz. At the top, a banner reads "Chris Fritz is creating tools and resources for Vue developers". Below this is a navigation bar with "Overview" (selected), "Posts", and "Community". A red button "BECOME A PATRON" is on the right. The left sidebar shows 21 patrons, a current total of \$464 per month, and a goal of \$5,000 per month. The main content area has a post titled "If you're here, thank you!" followed by a paragraph about the importance of open source and a paragraph about his work on Vue core docs, style guide, ESLint plugin, migration guide, and helper. It also mentions his enterprise boilerplate and behind-the-scenes work. The right sidebar shows a "REWARDS" section with a reward for \$5 or more per month, titled "Making It Happen", which includes a description of the reward and a "GET \$5 REWARD" button. Below that is a "Committed Supporter" section.

Chris Fritz is creating tools and resources for Vue developers

Overview Posts Community

BECOME A PATRON

21 patrons

\$464 per month - Public

SHARE

GOALS

\$464 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.

Big Projects

REWARDS

**Making It Happen**  
\$5 or more per month - 4 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 Vulcan?) salute and immediately receive a heartfelt handshake (or giant hug if you're a hugger - I'll follow your lead).

GET \$5 REWARD

Committed Supporter  
\$10 or more per month - 7 patrons

[patreon.com/chrisvuefritz](https://patreon.com/chrisvuefritz)



# Components Rewrite!

## Big Projects

There are a lot of little things I'd like to do with more Vue time, but here are some of the bigger projects that will only ever happen when I can spend more of my time on Vue:

- ~~Reorganize the huge **Components page** into several pages, making it easier to grasp the Essentials, while providing improved examples and deeper dives into each major feature.~~ **PUBLISHED 2018-03-26**
- Provide deeper dives into topics such as component organization, advanced composition, and some powerful component patterns.
- Rewrite the Vuex docs/examples with the same love that I've given the core docs.
- Rewrite the Vue Router docs/examples with the same love that I've given the core docs.
- Research and possibly build a new solution for

Thanks to **Cameron Adams** the Vuesionary!!

[github.com/chrisvfritz/7-secret-patterns](https://github.com/chrisvfritz/7-secret-patterns)

@chrisvfritz