



Welcome to

# Mastering Vue.js



# Hello!

# I am Jason Taylor



codingflow.net



[github.com/jasongt](https://github.com/jasongt)



@jasongtau @thiagospassos



# Hello!

# I am Thiago Passos



[passos.com.au](http://passos.com.au)



[github.com/thiagospassos](https://github.com/thiagospassos)



@jasongtau @thiagospassos



# Our Goal?

- For you to have fun
- To hit the ground running
- Learn the process from start to finish
- For you to be as excited about Vue.js as we are



@jasongtau @thiagospassos

# About You?

- What's your name?
- What's your background?
- What do you expect from it?
- What's your main focus?



@jasongtau @thiagospassos

# Agenda

**Getting Started**

**Styling the UI**

**Beyond the Basics**

**Consuming APIs**

**Dynamic Forms and Validation**

**State Management**

**Automated Testing**

**Packaging and Deploying**

# Agenda

**Getting Started**

**Styling the UI**

**Beyond the Basics**

**Consuming APIs**

**Dynamic Forms and Validation**

**State Management**

**Automated Testing**

**Packaging and Deploying**

# Agenda

**Getting Started**

**Styling the UI**

**Beyond the Basics**

**Consuming APIs**

**Dynamic Forms and Validation**

**State Management**

**Automated Testing**

**Packaging and Deploying**

# Agenda

**Getting Started**

**Styling the UI**

**Beyond the Basics**

**Consuming APIs**

**Dynamic Forms and Validation**

**State Management**

**Automated Testing**

**Packaging and Deploying**

# Agenda

**Getting Started**

**Styling the UI**

**Beyond the Basics**

**Consuming APIs**

**Dynamic Forms and Validation**

**State Management**

**Automated Testing**

**Packaging and Deploying**

# Agenda

**Getting Started**

**Styling the UI**

**Beyond the Basics**

**Consuming APIs**

**Dynamic Forms and Validation**

**State Management**

**Automated Testing**

**Packaging and Deploying**

# Agenda

**Getting Started**

**Styling the UI**

**Beyond the Basics**

**Consuming APIs**

**Dynamic Forms and Validation**

**State Management**

**Automated Testing**

**Packaging and Deploying**

# Agenda

**Getting Started**

**Styling the UI**

**Beyond the Basics**

**Consuming APIs**

**Dynamic Forms and Validation**

**State Management**

**Automated Testing**

**Packaging and Deploying**

# Agenda

**Getting Started**

**Styling the UI**

**Beyond the Basics**

**Consuming APIs**

**Dynamic Forms and Validation**

**State Management**

**Automated Testing**

**Packaging and Deploying**



# Mastering Vue.js

## Getting Started

# Agenda

Background

Working with Vue

Core Features

# Agenda

Background

Working with Vue

Core Features

# History

- Dynamic HTML 1996
- jQuery 2006
- Knockout.js 2010
- AngularJS 2010
- Backbone.js 2010



@jasongtau @thiagospassos

# Evan You

I figured, what if I could just extract the part that I really liked about Angular and build something really lightweight.



@jasongtau @thiagospassos

# What's Vue?

- A progressive JavaScript framework
- Incrementally adoptable, core is focused on view only
- Additional libraries support more sophisticated features



@jasongtau @thiagospassos

# Why Choose Vue?

## Approachable

Already know HTML, CSS and JavaScript?  
Read the guide and start building things in no time!

## Versatile

An incrementally adoptable ecosystem that scales between a library and a full-featured framework

## Performant

20KB min+gzip Runtime, Blazing Fast Virtual DOM, Minimal Optimization Efforts



@jasongtau @thiagospassos

# Features

- Reactivity
- Data Binding
- Components
- Events Handling
- Animation / Transition
- Computed Properties
- Templates
- Directives
- Routing
- State Management
- Lightweight
- Much more!



@jasongtau @thiagospassos

# Agenda

Background

Working with Vue

Core Features

# Essential Skills

HTML



JS



# Recommended Tools



Visual Studio Code



Vue VSCode  
Snippets



Vue  
Dev Tools



@jasongtau @thiagospassos

# Demo: Hello, World!

- It would be completely irresponsible if we didn't start here



@jasongtau @thiagospassos

# Demo: Todo App

- Create a simple app for managing a list of todo items



@jasongtau @thiagospassos

# Agenda

Background

Working with Vue

Core Features

# The Vue Instance

Started with a root Vue instance

Passed in an options object:

- el
- data
- methods
- computed

Additional options

```
var app = new Vue({  
  el: '#app',  
  data: {  
  },  
  methods: {  
  },  
  computed: {  
  },  
})
```



@jasongtau @thiagospassos

# Template Syntax

Text

`{{ expression }}`

Attributes

`v-bind:id="..." or :id="..."`

Events

`v-on:click="..." or @click="..."`

Modifiers

`@submit.prevent="..."`



@jasongtau @thiagospassos

# Computed Properties

Reactive and Cached

```
computed: {  
  itemCount: function() {  
    return this.items.length;  
  }  
}
```

Normal binding

```
{{ itemCount }}
```



@jasongtau @thiagospassos

# Class and Style Bindings

Class

```
:class="{'done': t.done}"
```

Multiple Classes

```
:class="{'done': t.done, ...}"
```

Styles

```
:style="{'color': todoColor}"
```

Style object

```
:style="todoStyleObject"
```



@jasongtau @thiagospassos

# Conditional Rendering

v-if

```
<div v-if="items.length > 1">  
  {{ remaining }} item(s) remaining.  
</div>
```

v-else-if

```
<div v-else-if="items.length === 1">  
  Last item, get it done!  
</div>
```

v-else

```
<div v-else>  
  All done!  
</div>
```



@jasongtau @thiagospassos

# List Rendering

v-for

```
<a v-for="filter in filters">  
  {{ filter }}  
</a>  
<a v-for="(filter, index) in filters">  
  {{ index }} - {{ filter }}  
</a>
```



@jasongtau @thiagospassos

# Event Handling

v-on or @

```
<a @click="activeFilter = filter">  
  {{ filter }}  
</a>
```



@jasongtau @thiagospassos

# Form Binding

v-model

```
<input v-model="todo" />
```



@jasongtau @thiagospassos

# Your Turn!

Follow the instructions to complete  
the code exercises and challenges

 Workshop Guide  
[bit.ly/masteringvuejs](https://bit.ly/masteringvuejs)

 Workshop Code  
[bit.ly/vuejs-minnesota-2019](https://bit.ly/vuejs-minnesota-2019)



@jasongtau @thiagospassos



# Mastering Vue.js Beyond the Basics

# Agenda

Components

New App with the CLI

Views, Routes and more...

Debugging and Style Guide

# Agenda

Components

New App with the CLI

Views, Routes and more...

Debugging and Style Guide



# Agenda

Components

New App with the CLI

Views, Routes and more...

Debugging and Style Guide

# What do you need?



@jasongtau @thiagospassos

# What do we suggest?



@jasongtau @thiagospassos

```
λ npm i -g @vue/cli
λ vue create app-name
λ vue ui
```



@jasongtau @thiagospassos



50

# Agenda

Components

New App with the CLI

Views, Routes and more...

Debugging and Style Guide



# Agenda

Components

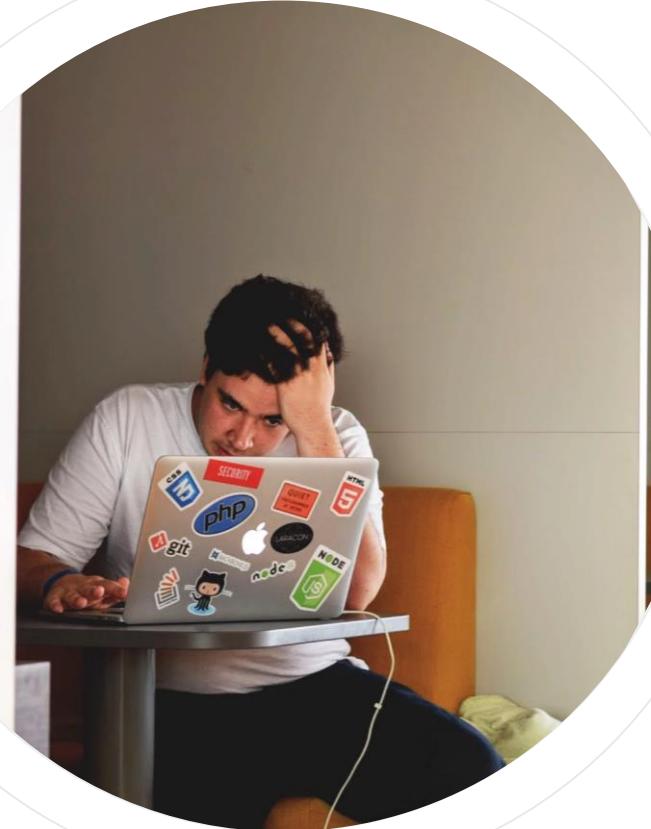
New App with the CLI

Views, Routes and more...

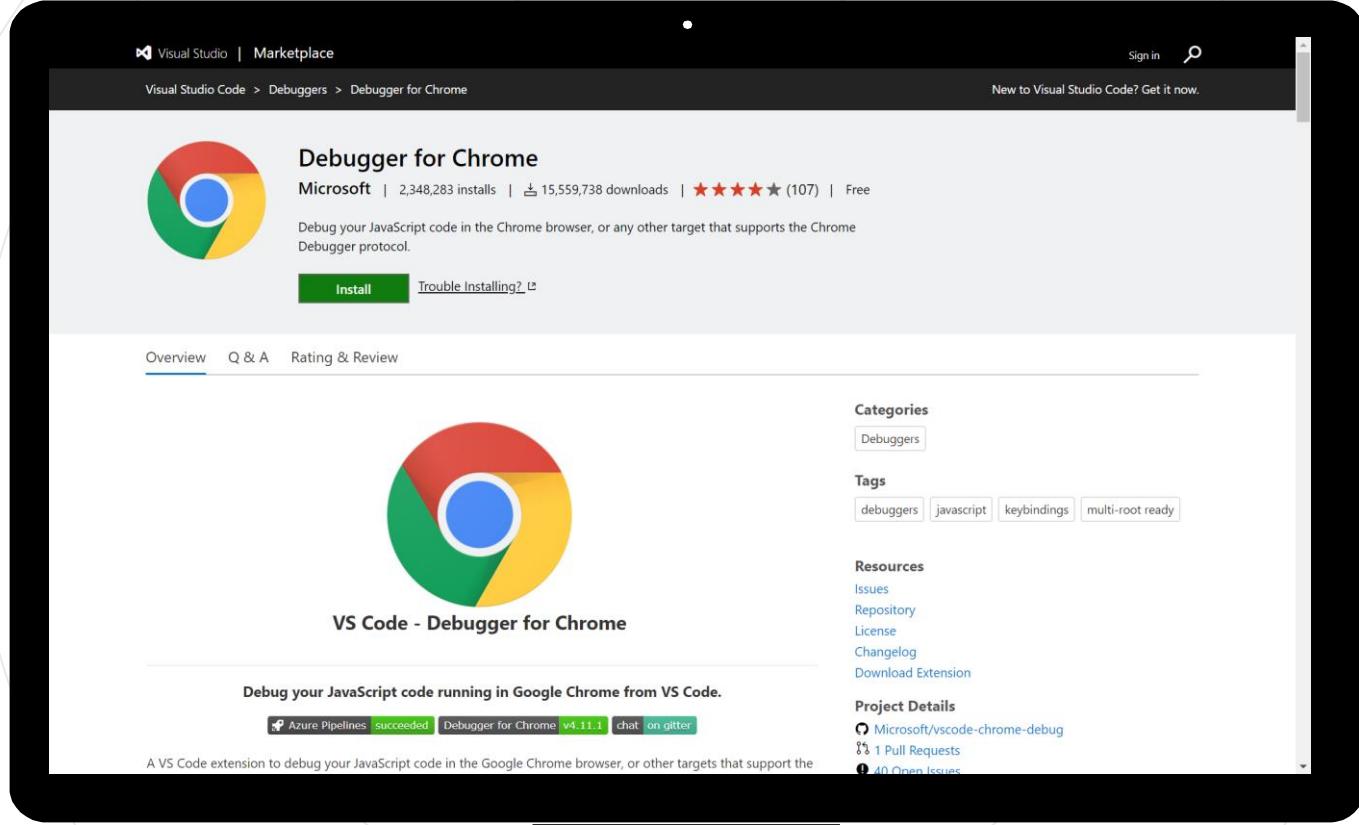
Debugging and Style Guide

# Debugging

[bit.ly/vue-debugging](https://bit.ly/vue-debugging)



@jasongtau @thiagospassos



@jasongtau @thiagospassos

## Vue.config.js

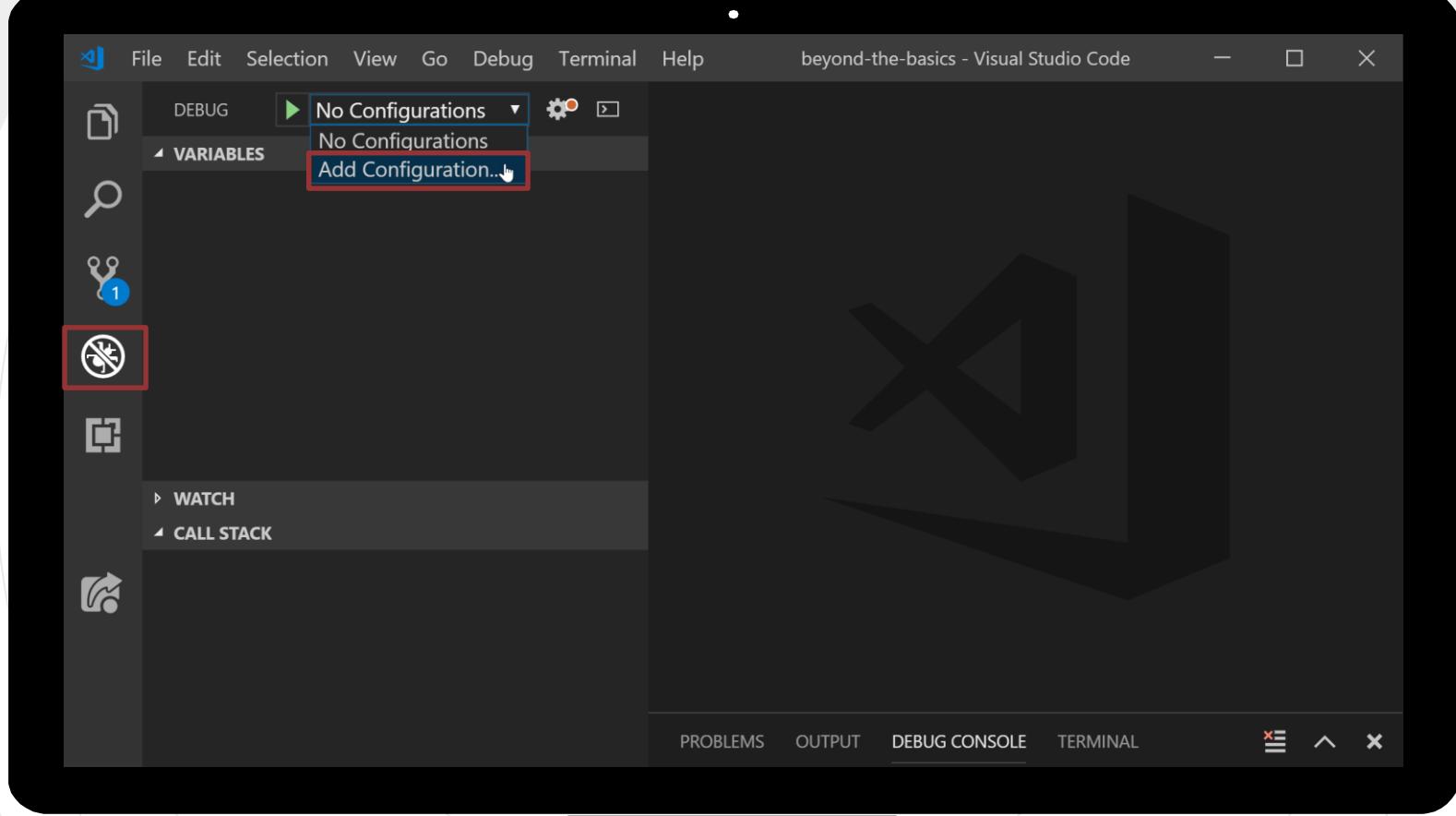
A screenshot of the Visual Studio Code interface. The title bar reads "masteringvuejs-code-demo - Visual Studio Code". The left sidebar shows icons for file operations like Open, Save, Find, and Undo. The main editor area contains the following code:

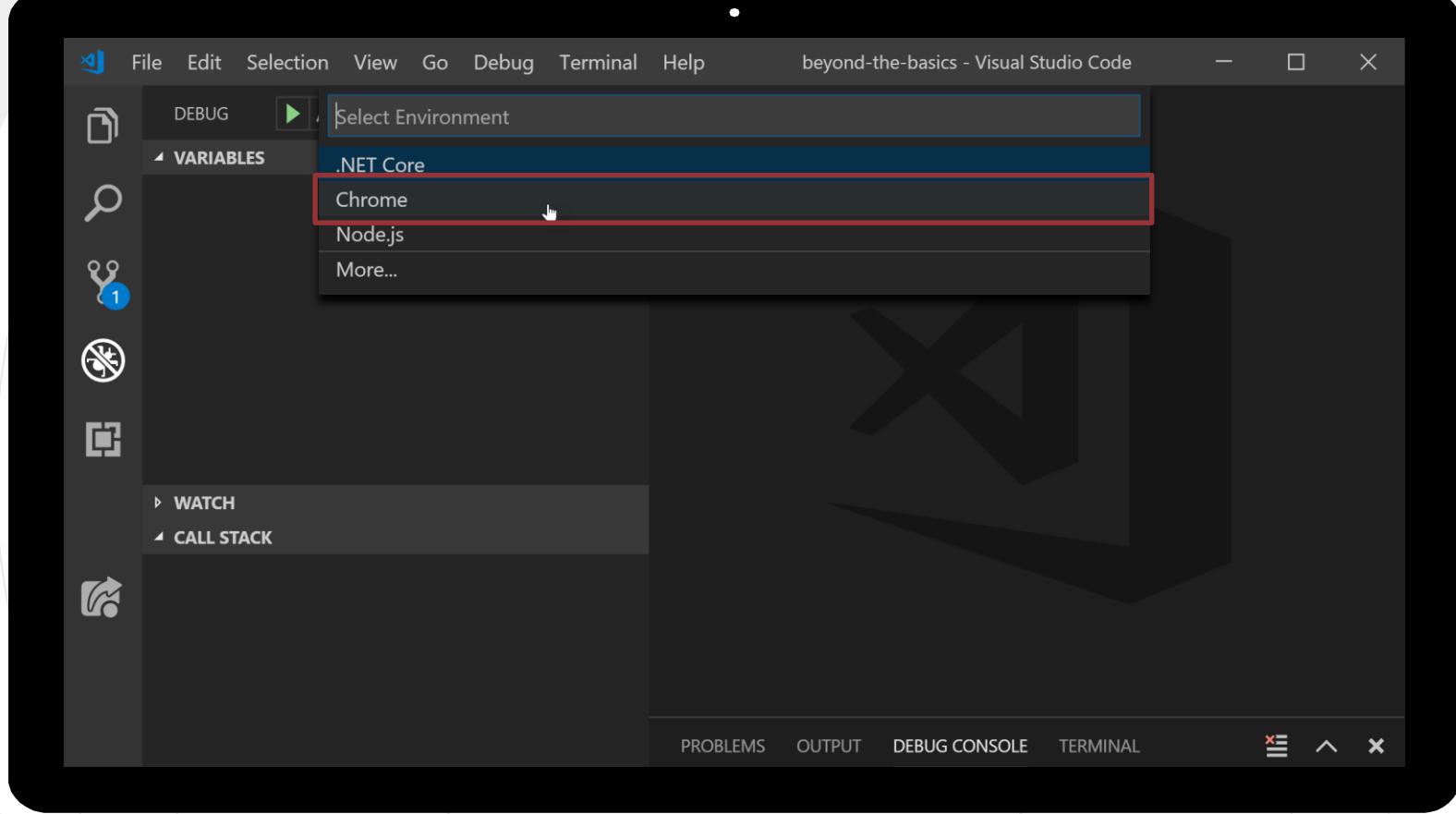
```
module.exports = {
  configureWebpack: {
    devtool: 'source-map'
  }
}
```

The status bar at the bottom shows the current branch is "master\*", with 0 changes, 0 warnings, and 0 errors. It also displays "Live Share" and "Prettier" status indicators.



@jasongtau @thiagospassos





A screenshot of the Visual Studio Code interface. The title bar shows "File Edit Selection View Go Debug Terminal Help launch.json - beyond-the-basics - Visual Studio C...". The left sidebar has icons for file, search, and a task list with one item. The main editor area contains the following JSON code:

```
1  [
2      // Use IntelliSense to learn about possible attributes.
3      // Hover to view descriptions of existing attributes.
4      // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
5      "version": "0.2.0",
6      "configurations": [
7          {
8              "type": "chrome",
9              "request": "launch",
10             "name": "Launch Chrome against localhost",
11             "url": "http://localhost:8080",
12             "webRoot": "${workspaceFolder}"
13         }
14     ]
15 ]
```

The bottom right of the editor has a blue button labeled "Add Configuration...". Below the editor are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE (which is selected), and TERMINAL. The bottom right corner of the interface shows a circular badge with the number 58.





File Edit Selection View Go Debug Terminal Help

DEBUG Launch C ⚙️ 🛡️

**VARIABLES**

- Local
  - this: undefined
  - format: "YYYY"
  - value: Sat May 04 2019 16:12:41
- Closure
- Global

**WATCH**

You, 4 hours ago | 1 author (You)

```
import moment from 'moment'  
export default function(value, format) {  
  if (format) {  
    return moment(value).format(format)  
  }  
  return value  
}
```



@jasongtau @thiagospassos



60

# Style Guide

[bit.ly/vue-style-guide](https://bit.ly/vue-style-guide)



@jasongtau @thiagospassos

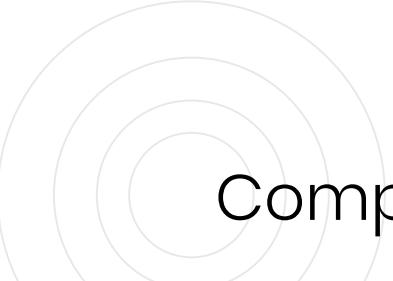
```
Vue.component('todo-item', {  
    // ...  
})  
  
export default {  
    name: 'TodoItem',  
    // ...  
}
```



Multi-word component names

```
Vue.component('some-comp', {  
  data: function () {  
    return {  
      foo: 'bar'  
    }  
  }  
})
```

```
export default {  
  data () {  
    return {  
      foo: 'bar'  
    }  
  }  
}
```



Component data must be a function

```
props : {  
    status: String  
}
```

```
props: {  
    status:{  
        type: String,  
        required: true,  
        validator: function(value){  
            // ...  
        }  
    }  
}
```



Props should be detailed

```
<ul>
  <li
    v-for="todo in todos"
    :key="todo.id">
      {{ todo.text }}
    </li>
</ul>
```



Always use key for v-for

```
<ul v-if="shouldShowUsers">
  <li
    v-for="user in users"
    :key="user.id">
      {{ user.name }}
    <li>
  </ul>
```

Avoid v-if with v-for in the same element

```
<style scoped>
  .button {
    border: none;
    border-radius: 2px;
  }

  .button-close {
    background-color: red;
  }
</style>
```

## Component style scoping

# Your Turn!

Follow the instructions to complete  
the code exercises and challenges

 Workshop Guide  
[bit.ly/masteringvuejs](https://bit.ly/masteringvuejs)

 Workshop Code  
[bit.ly/vuejs-minnesota-2019](https://bit.ly/vuejs-minnesota-2019)



@jasongtau @thiagospassos



# Mastering Vue.js Dynamic Forms & Validation

# Agenda

Best Practices

Inline Forms & Validation

Complex Forms & Validation

# Agenda

Best Practices

Inline Forms & Validation

Complex Forms & Validation

# Best Practices

- Forms and validation are arguably the most important interaction between your application and your users
- Let's examine some best practices

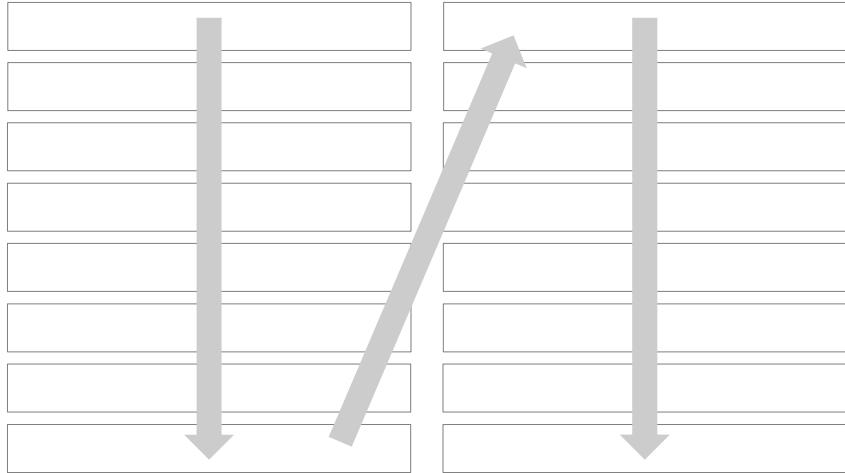


# 1. Single Column Layout

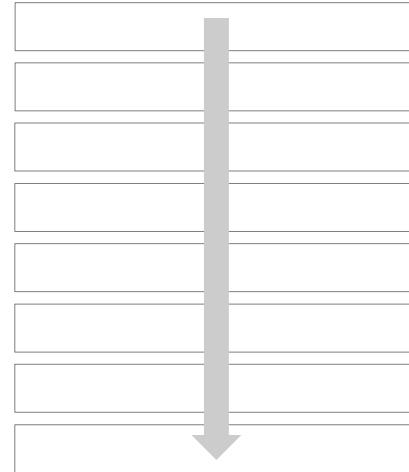
- Prefer single column layouts over multi column layouts
- Single column layouts have a clear path to completion
- Multi column layouts require the user to scan the form to determine the path to completion



Bad Example 



Good Example 



@jasongtau @thiagospassos

## 2. Organise Fields Logically

- Order fields logically from the user's perspective, not the application or database perspective
- Order fields from easiest to hardest.  
Don't make it hard just to get started!



## Bad Example



Address

Full Name

Email

## Good Example



Full Name

Email

Address


@jasongtau @thiagospassos

## Bad Example



Credit Card Number

Expiry

CCV

Address

Full Name

## Good Example



Full Name

Address

Credit Card Number

Expiry

CCV



@jasongtau @thiagospassos

# 3. Group Related Fields

- Grouping related fields into logical sets will improve layout and user comprehension



@jasongtau @thiagospassos

## Bad Example



Full Name

Address

Credit Card Number

Expiry

CCV

## Good Example



### Personal

Full Name

### Address

### Payment

Credit Card Number

Expiry

CCV



@jasongtau @thiagospassos

# 4. Inline Validation

- Use inline validation to provide fast and effective feedback
- For complex forms, avoid waiting until the user submits the form
- Ideally validate after input, not on each keystroke



## Bad Example



Full Name

Email

Credit Card Number

Expiry

CCV

You really messed up!

- Full Name is required.
- Email is invalid.
- Credit card number is required.
- Expiry is required.
- CCV is required.

## Good Example



Full Name

Required field

Email

Credit Card Number

Expiry

CCV



# 5. Clear Actions

- Ensure that the form actions are clearly visible and well-defined
- Emphasis should be placed on the primary action



## Bad Example



Email

Password

Register

Cancel

*(evil example)*

Email

Password

Cancel

Register

## Good Example



Email

Password

Register

Cancel

*(or)*

Register

Cancel



@jasongtau @thiagospassos

# Agenda

Best Practices

Inline Forms & Validation

Complex Forms & Validation

# Demo: Inline Forms

- Create a form for inline editing of categories



# Demo: Simple Validation

- Add validation support to the new categories form



@jasongtau @thiagospassos

# Agenda

Best Practices

Inline Forms & Validation

Complex Forms & Validation

# Demo: Complex Forms

- Create a form for editing products



@jasongtau @thiagospassos

# Demo: Advanced Validation

- Implement complex validation rules with Vuelidate



# Your Turn!

Follow the instructions to complete  
the code exercises and challenges

 Workshop Guide  
[bit.ly/masteringvuejs](https://bit.ly/masteringvuejs)

 Workshop Code  
[bit.ly/vuejs-minnesota-2019](https://bit.ly/vuejs-minnesota-2019)



@jasongtau @thiagospassos



# Mastering Vue.js Automated Testing

# Agenda

why automated testing?

Test types and frameworks

Testing our components

Testing our solution

# Agenda

why automated testing?

Test types and frameworks

Testing our components

Testing our solution

# Why automated testing?



@jasongtau @thiagospassos

A large white circle is centered on the slide. Inside it, a smaller gray circle overlaps from the top-left. To the right of the center, there is a series of concentric white circles on a black background.

1

**Improves  
confidence and  
remove fear of  
changing code**

The background features a large white circle on a black base. Inside it, there's a smaller gray circle with a white outline. To the right, a series of concentric circles in white outlines are visible against the black background.

2

# Code Quality

“Test a single concept in each test function” (Uncle Bob)



**3**

# **Act as documentation**

# Agenda

why automated testing?

Test types and frameworks

Testing our components

Testing our solution

# Unit Tests

```
    it('renders "Hello World" when passed a message', () => {
      const wrapper = shallowMount(HelloWorld, { propsData: { msg: 'Hello World' } })
      expect(wrapper.text()).toMatch('Hello World')
    })
  })
})
```

```
describe('AddressForm.vue', () => {
  it('it should fail if no address is provided', () => {
    const wrapper = shallowMount(AddressForm)
    expect(wrapper.text()).toContain('Address is required')
  })
})
```



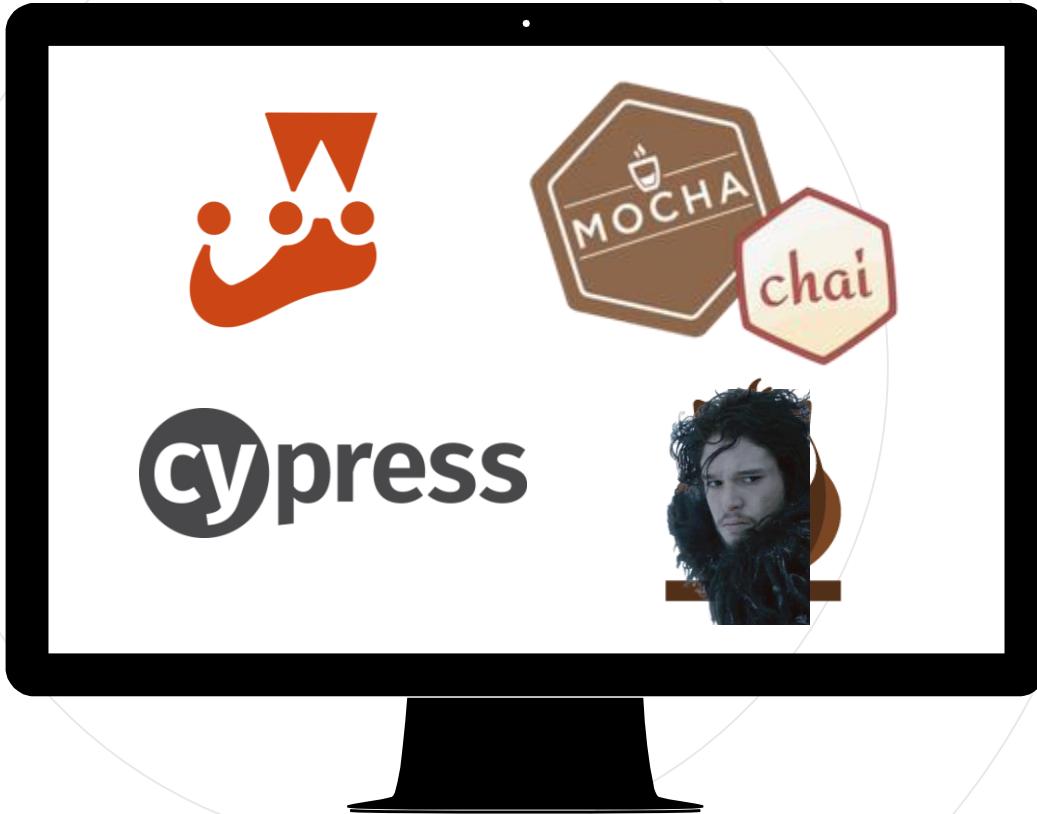
@jasongtau @thiagospassos

# Integration Tests



@jasongtau @thiagospassos

# Tools



@jasongtau @thiagospassos

# Agenda

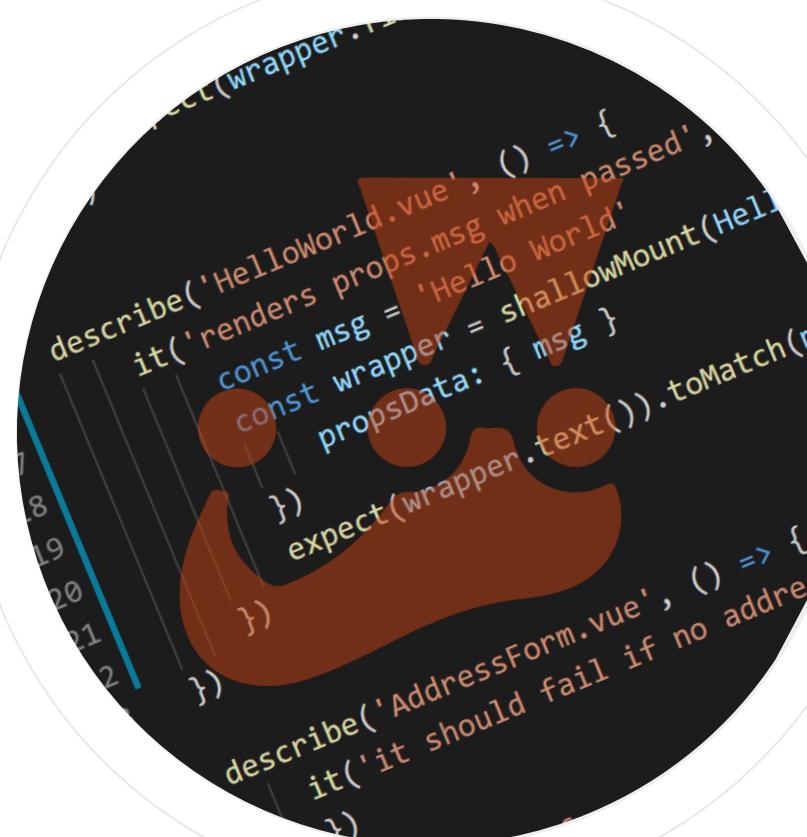
why automated testing?

Test types and frameworks

Testing our components

Testing our solution

# Unit Tests

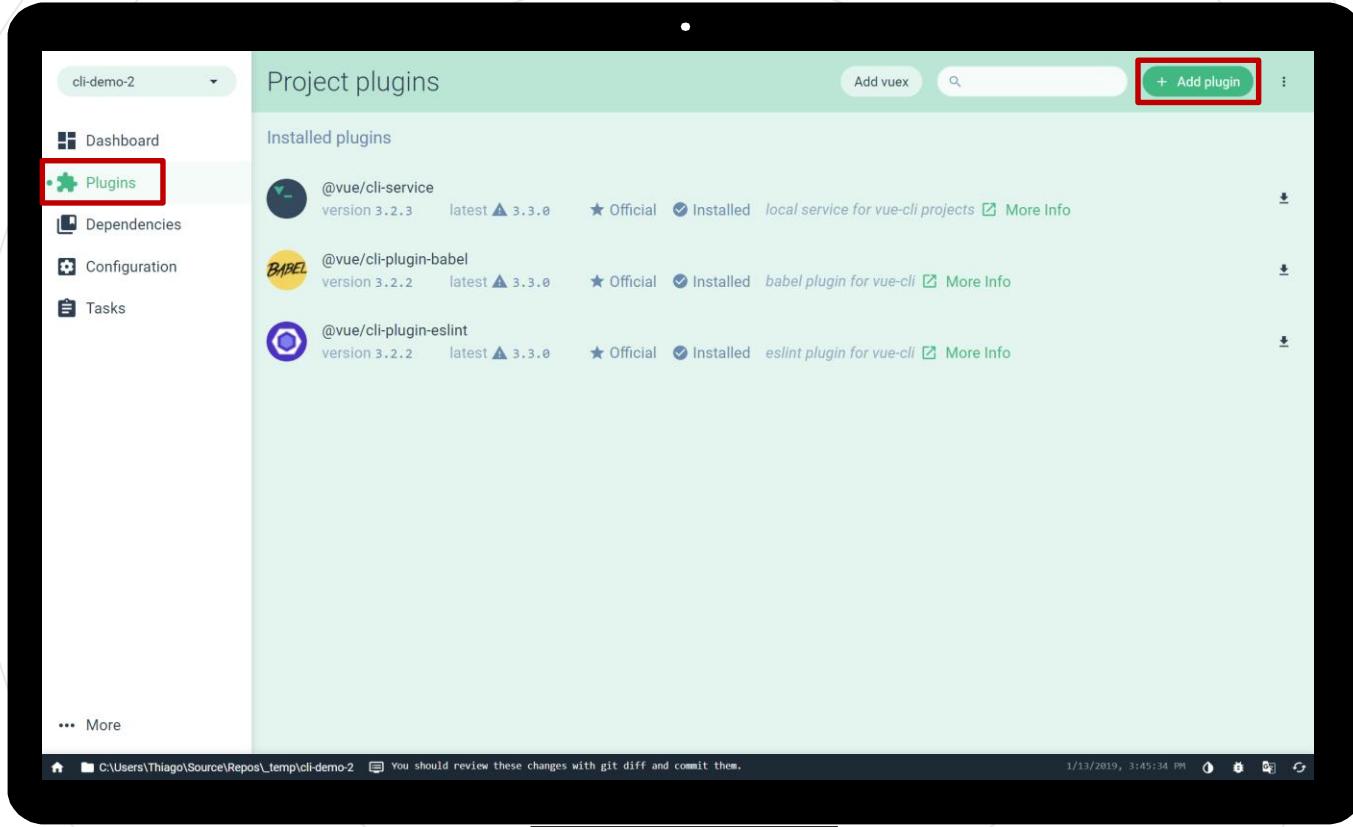


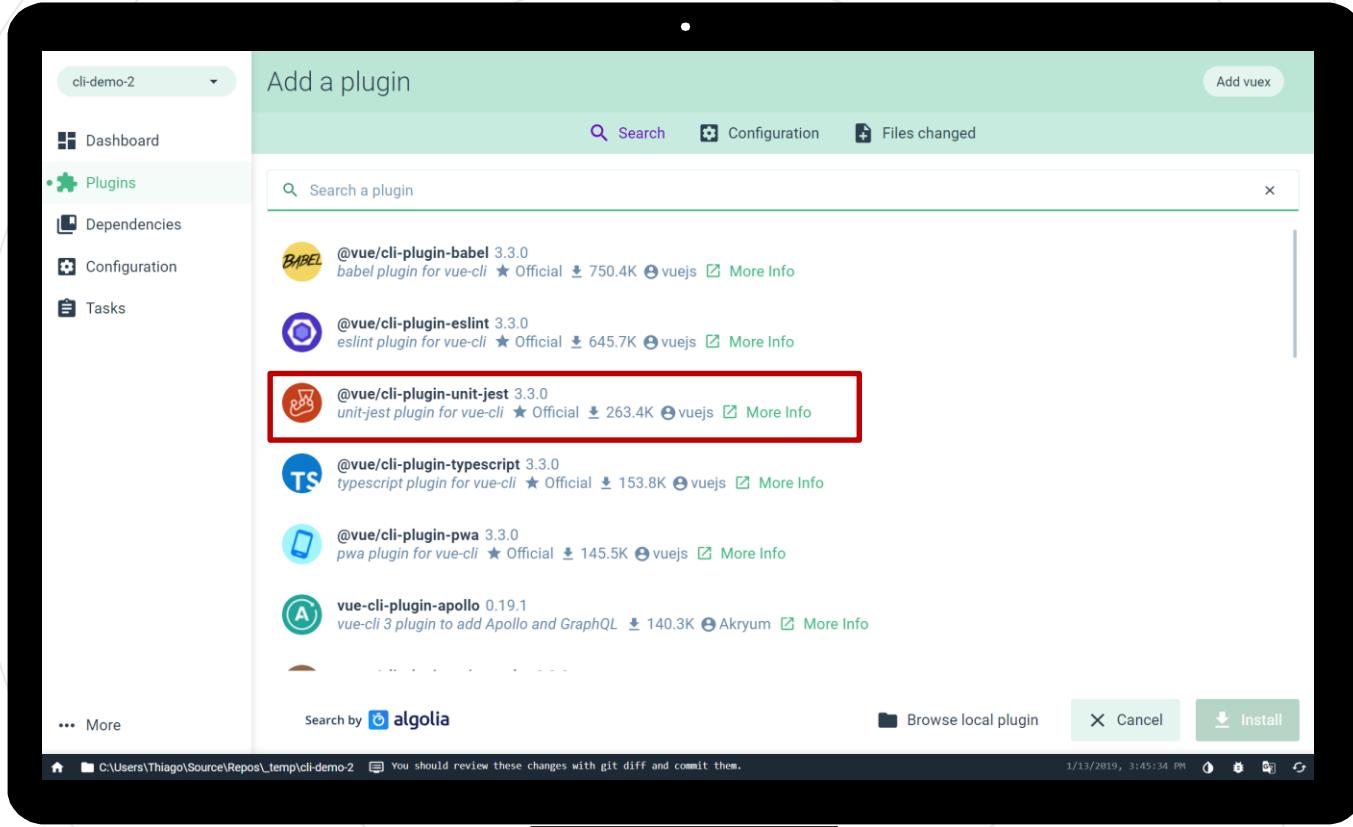
```
describe('HelloWorld.vue', () => {
  it('renders props.msg when passed',
    () => {
      const msg = 'Hello World'
      const wrapper = shallowMount(Hel
        propsData: { msg }
      })
      expect(wrapper.text()).toMatch(
        /Hello World/
      )
    }
  )
})
```

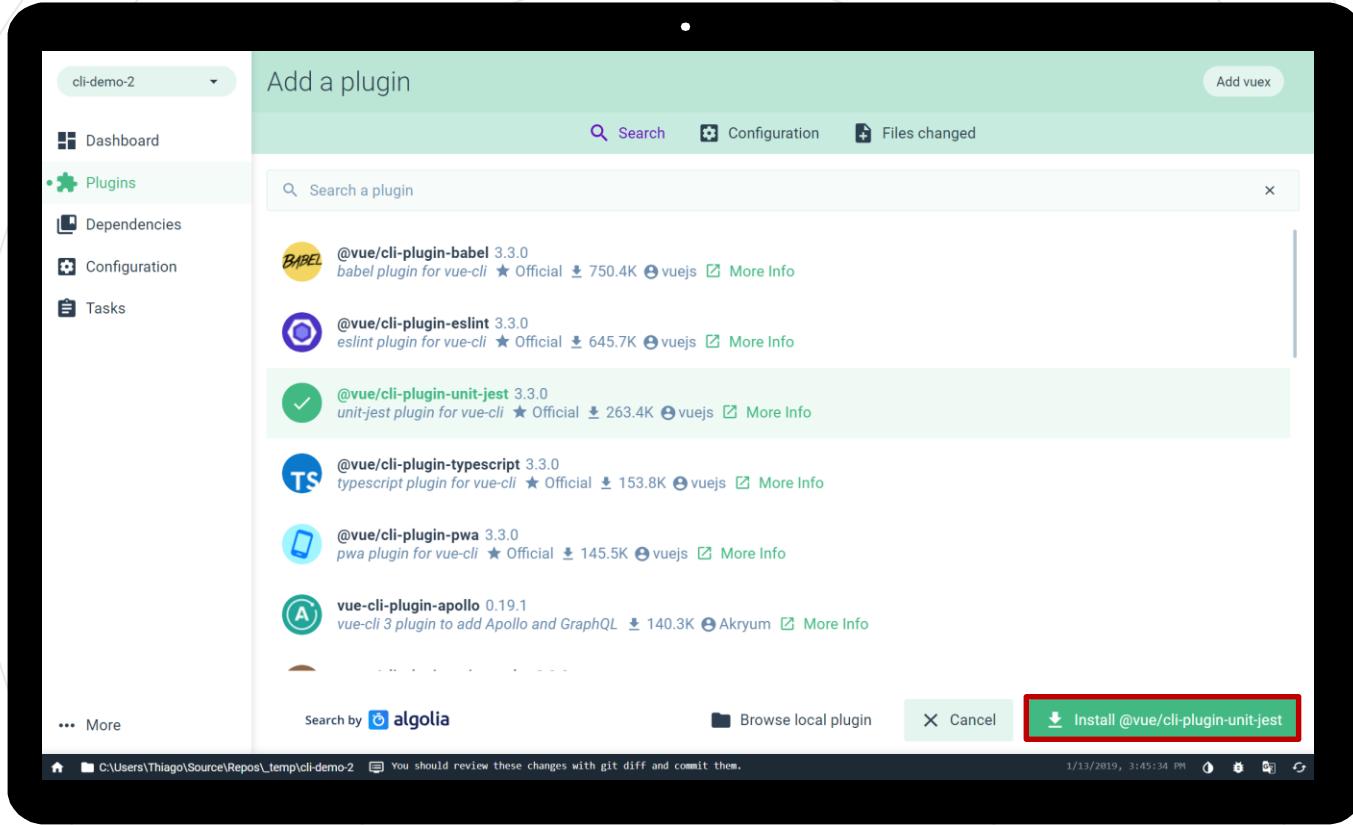
```
describe('AddressForm.vue', () => {
  it('it should fail if no addre
    () => {
      const wrapper = shallowMount(AddressForm)
      expect(wrapper.text()).toContain(
        'Please enter a valid address'
      )
    }
  )
})
```



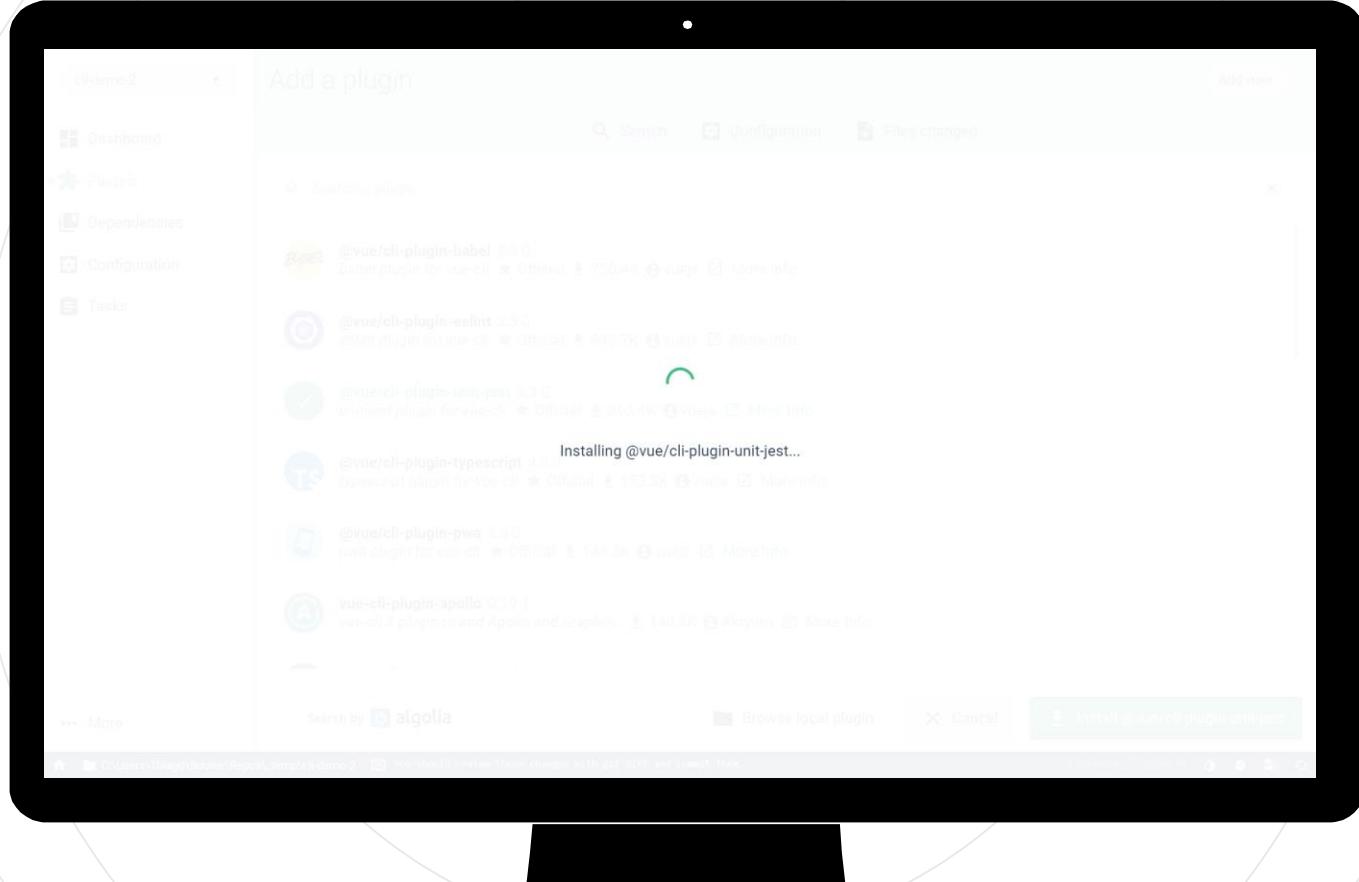
@jasongtau @thiagospassos



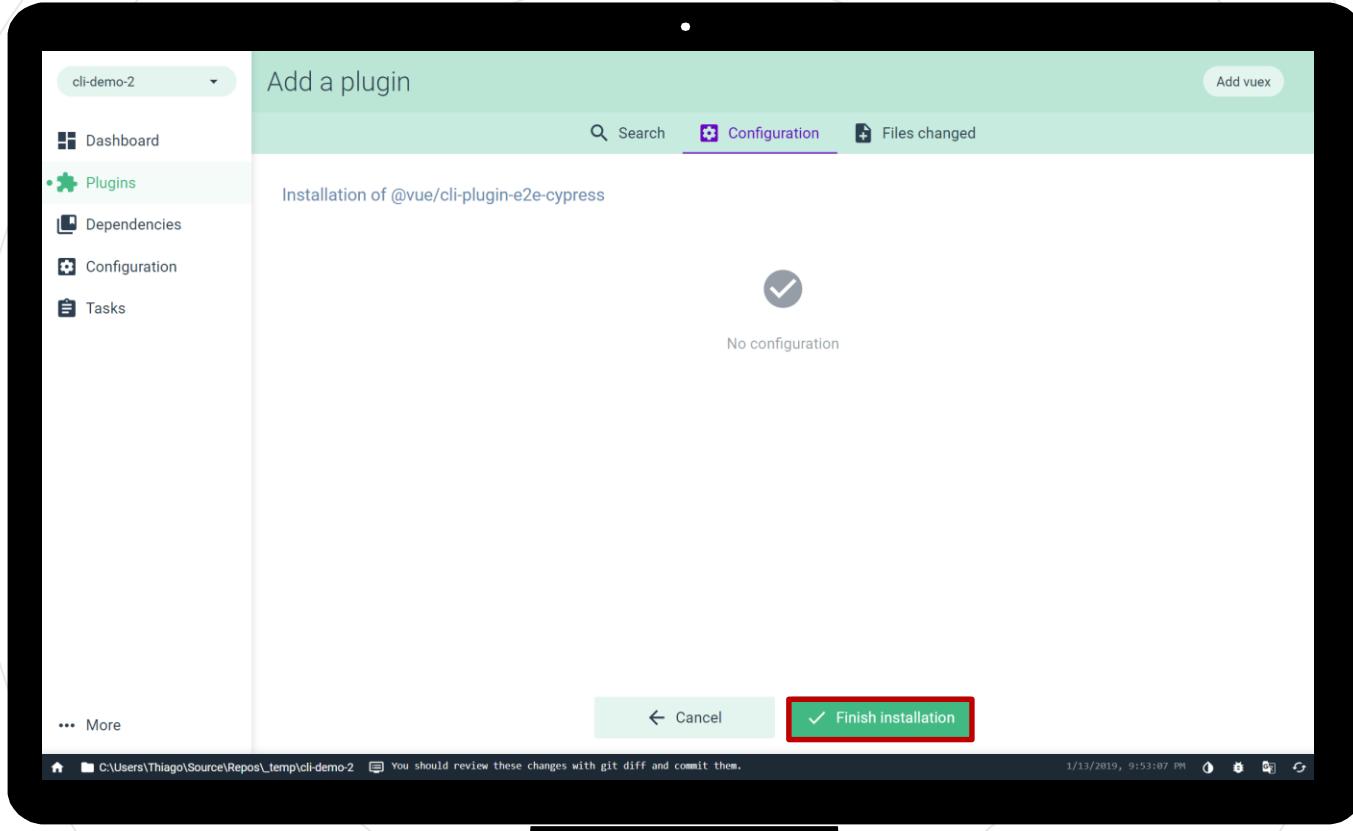


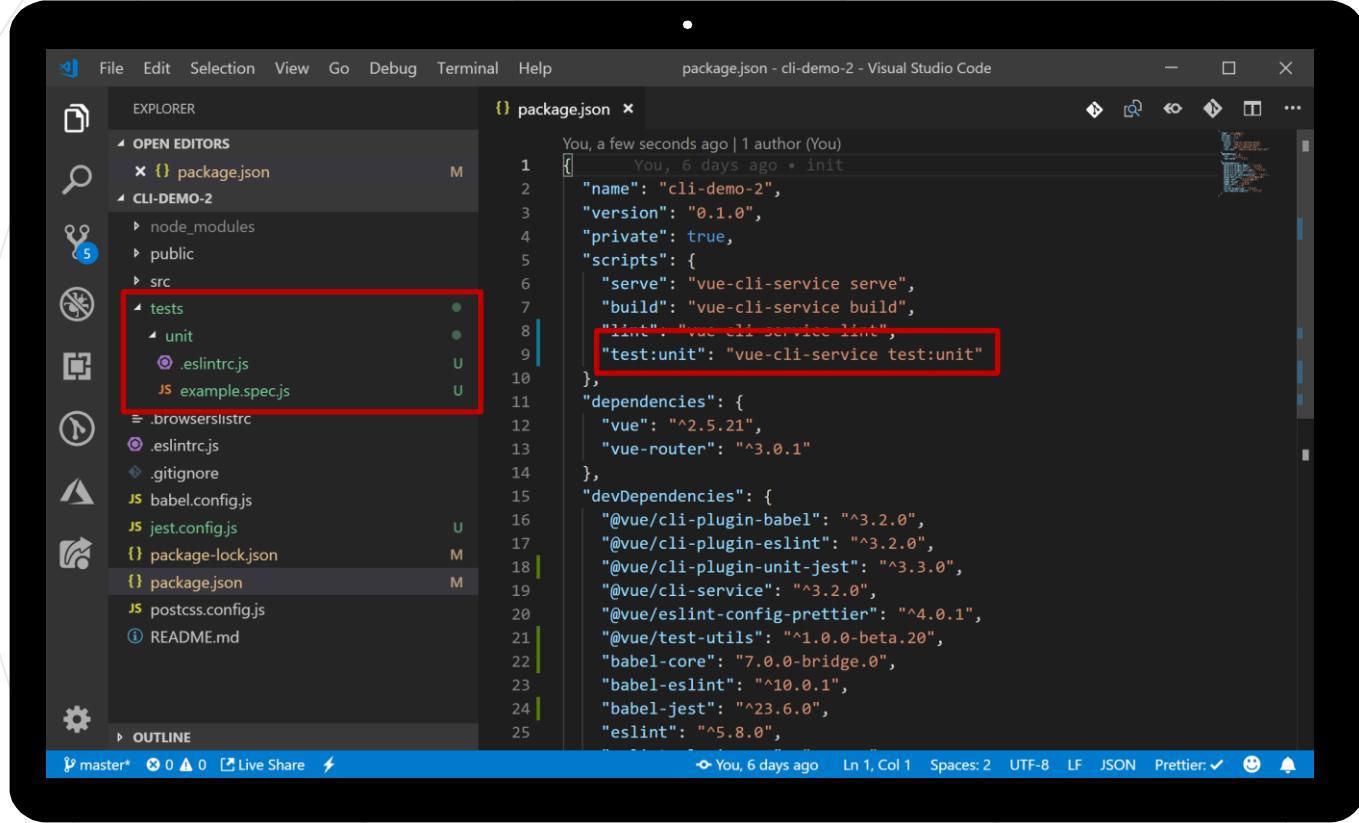


@jasongtau @thiagospassos



@jasongtau @thiagospassos





The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Debug, Terminal, Help
- Editor Bar:** package.json - cli-demo-2 - Visual Studio Code
- Explorer:** Shows the project structure:
  - OPEN EDITORS: package.json (M)
  - CLI-DEMO-2
    - node\_modules
    - public
    - src
      - tests
        - unit
        - .eslintrc.js (U)
        - example.spec.js (U)
  - .browserslistrc
  - .eslintrc.js
  - .gitignore
  - babel.config.js
  - jest.config.js
  - package-lock.json
  - package.json (M)
  - postcss.config.js
  - README.md
- Editor:** package.json (You, a few seconds ago | 1 author (You))

```
1  You, a few seconds ago | 1 author (You)
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
```

You, 6 days ago \* init

```
"name": "cli-demo-2",
"version": "0.1.0",
"private": true,
"scripts": {
  "serve": "vue-cli-service serve",
  "build": "vue-cli-service build",
  "lint": "vue-cli-service lint",
  "test:unit": "vue-cli-service test:unit"
},
"dependencies": {
  "vue": "^2.5.21",
  "vue-router": "^3.0.1"
},
"devDependencies": {
  "@vue/cli-plugin-babel": "^3.2.0",
  "@vue/cli-plugin-eslint": "^3.2.0",
  "@vue/cli-plugin-unit-jest": "^3.3.0",
  "@vue/cli-service": "^3.2.0",
  "@vue/eslint-config-prettier": "^4.0.1",
  "@vue/test-utils": "^1.0.0-beta.20",
  "babel-core": "7.0.0-bridge.0",
  "babel-eslint": "^10.0.1",
  "babel-jest": "^23.6.0",
  "eslint": "^5.8.0",
}
```
- Bottom Status Bar:** master\* 0 0 Live Share ⚡ You, 6 days ago Ln 1, Col 1 Spaces: 2 UTF-8 LF JSON Prettier: ✓ 😊 🔔



@jasongtau @thiagospassos



109

# Agenda

why automated testing?

Test types and frameworks

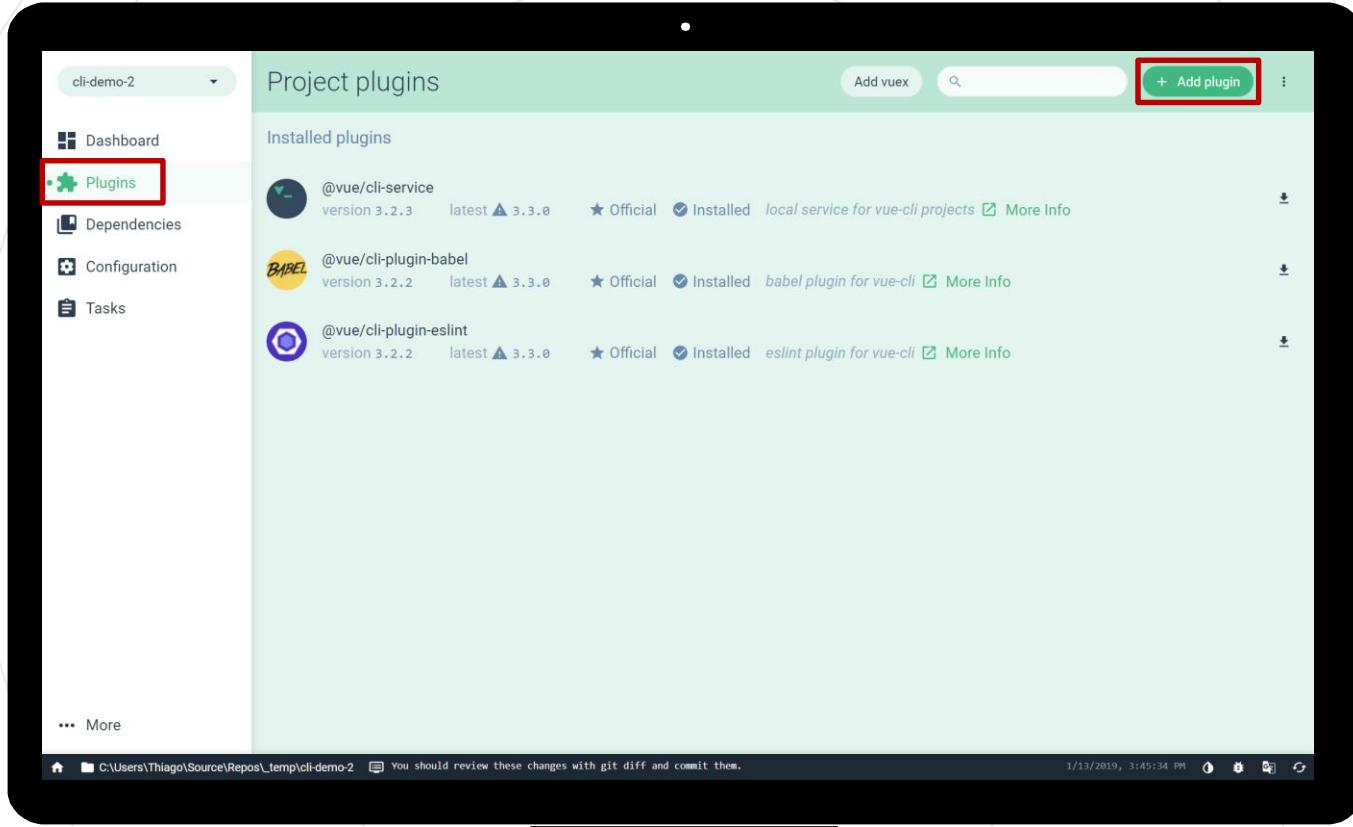
Testing our components

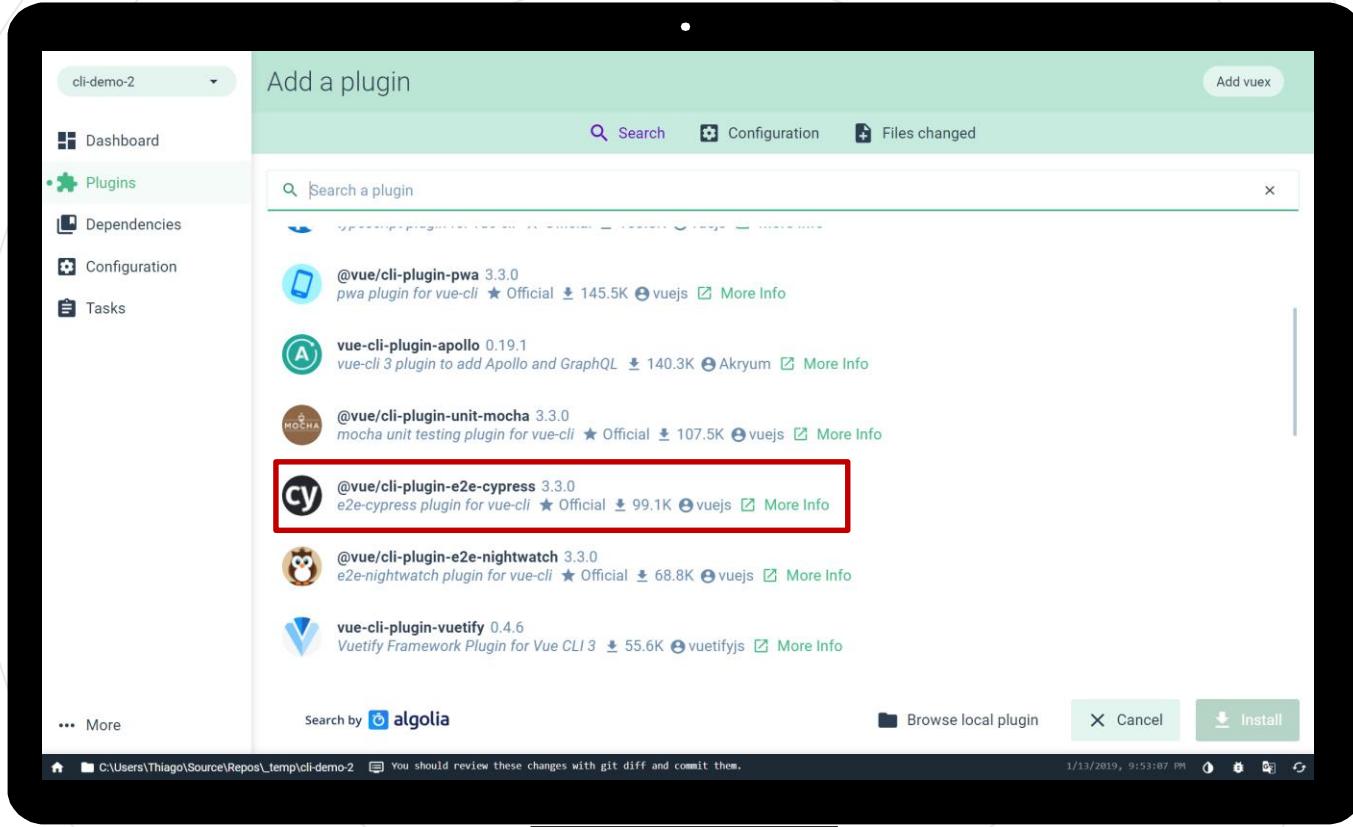
Testing our solution

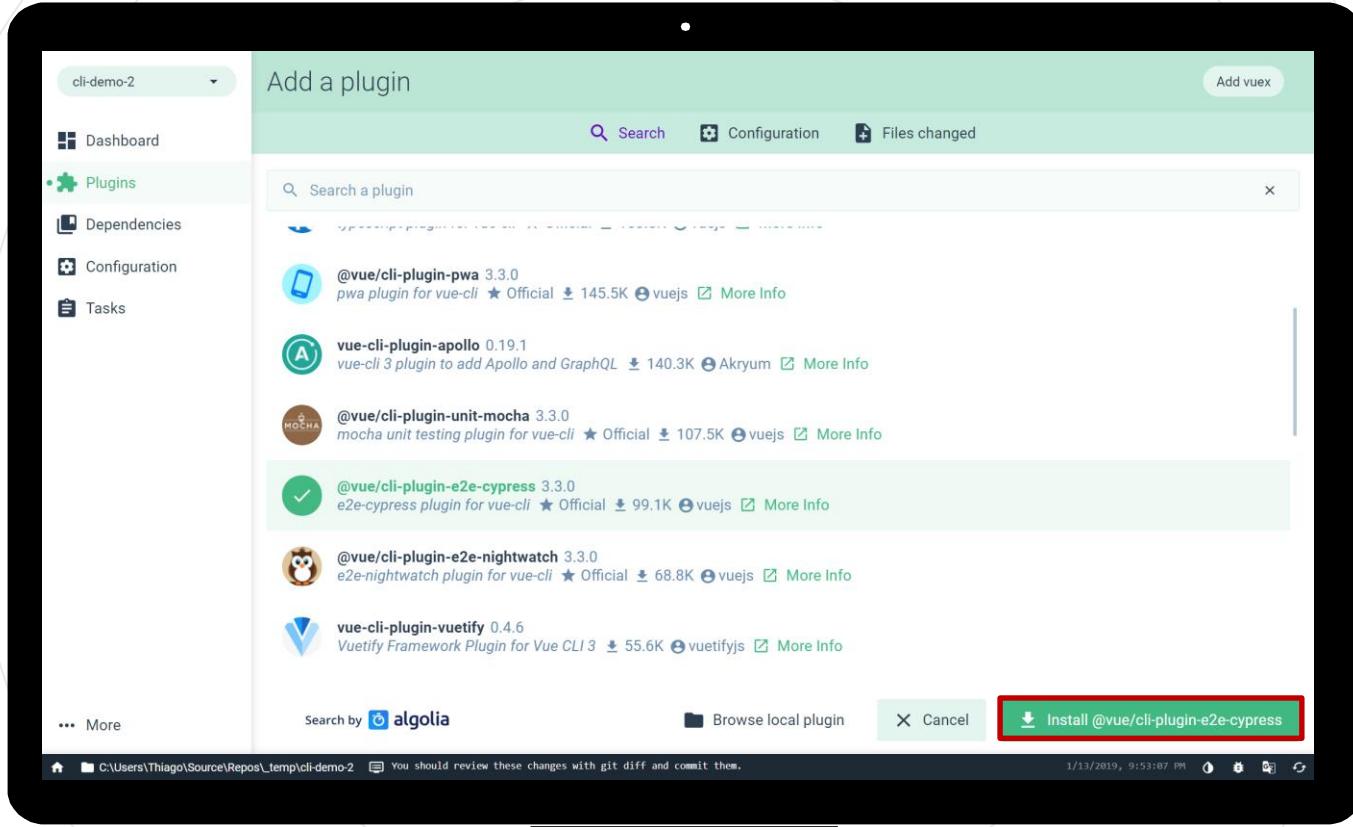
# Integration Tests

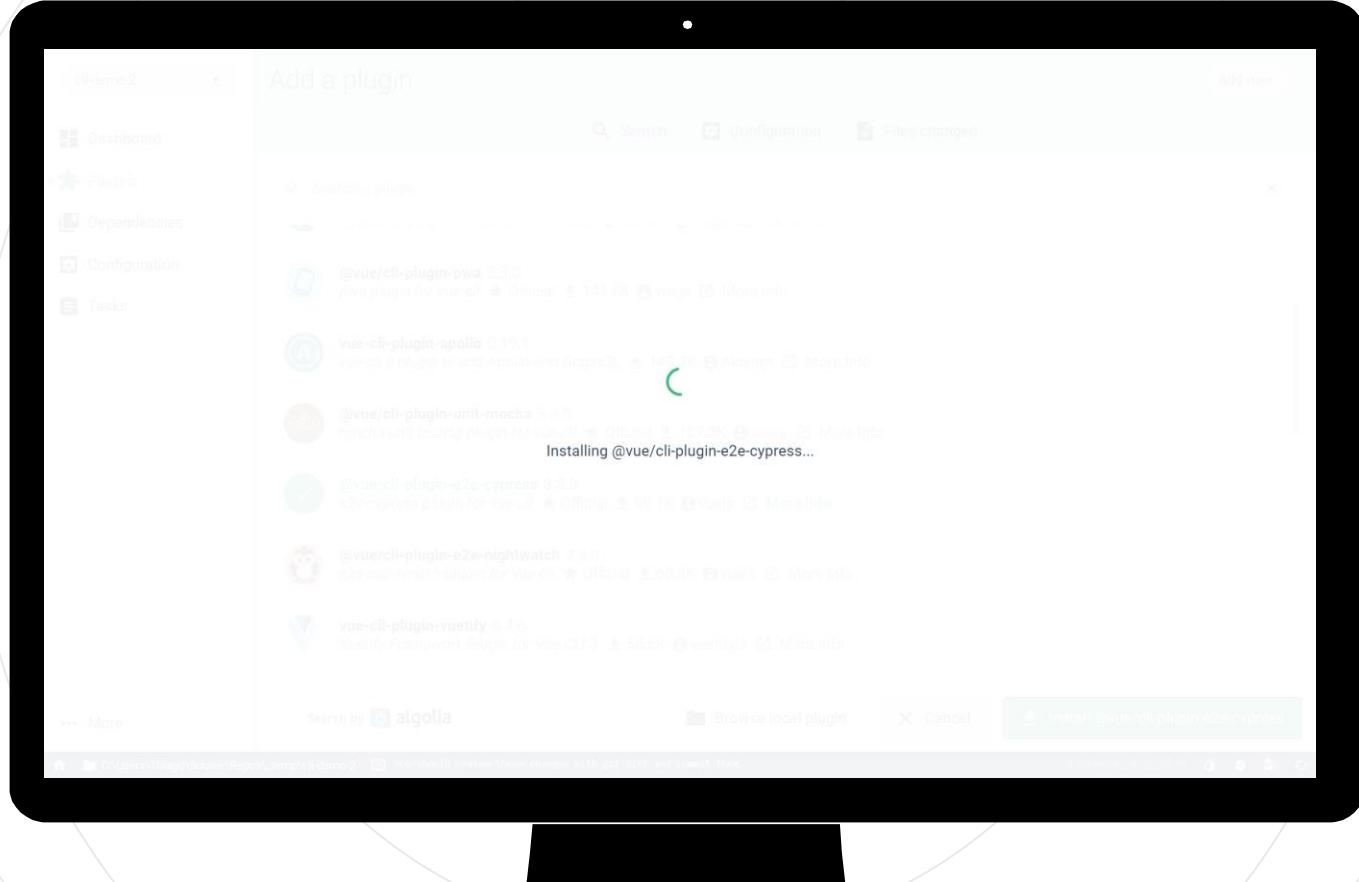


@jasongtau @thiagospassos

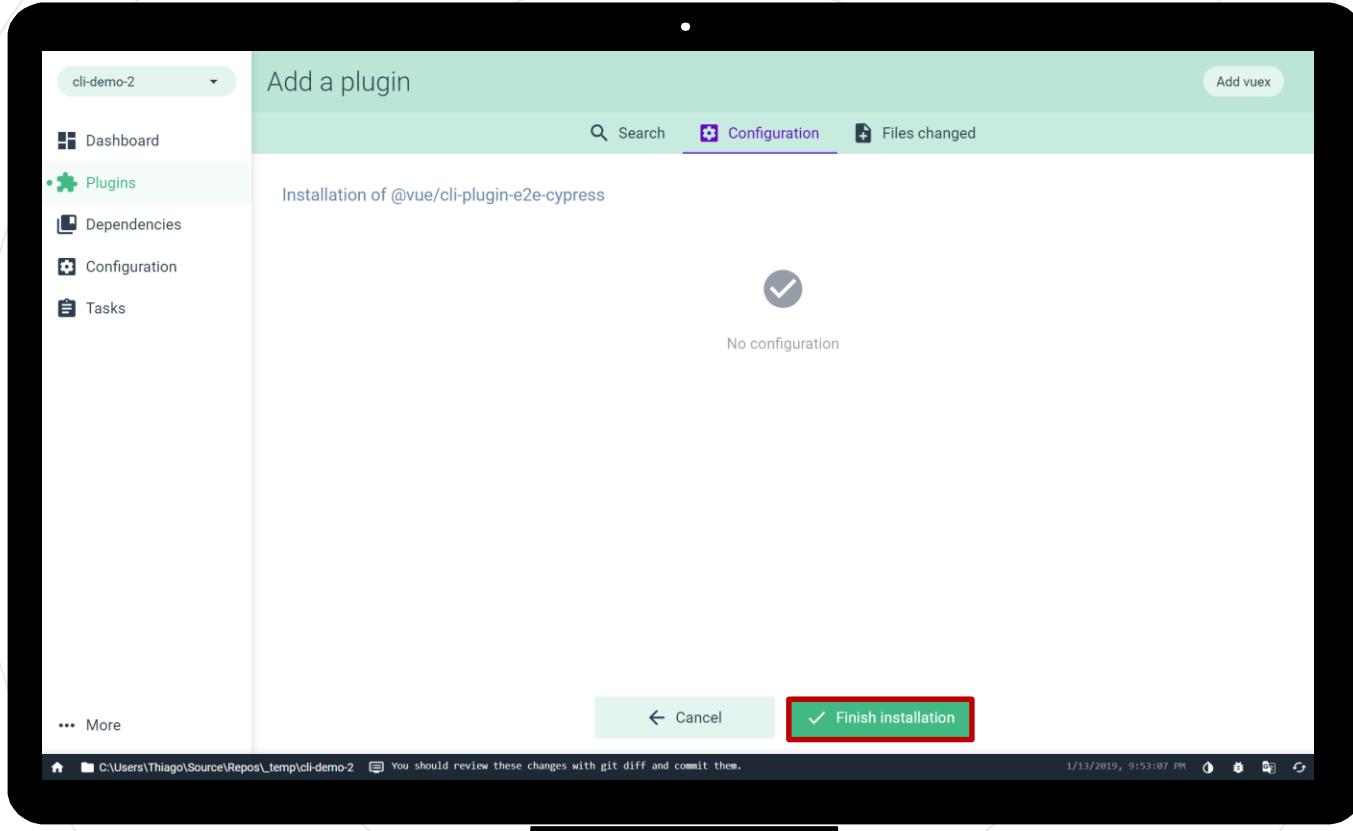








@jasongtau @thiagospassos



A screenshot of the Visual Studio Code interface. The left sidebar shows the file structure of a project named 'CLI-DEMO-2'. A red box highlights the 'e2e' folder under the 'tests' directory. The right pane displays the contents of the 'package.json' file. A second red box highlights the 'test:e2e' script in the 'scripts' section.

```
You, a few seconds ago | 1 author (You)
{
  "name": "cli-demo-2",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "serve": "vue-cli-service serve",
    "build": "vue-cli-service build",
    "lint": "vue-cli-service lint",
    "test:e2e": "vue-cli-service test:e2e",
    "test:unit": "vue-cli-service test:unit"
  },
  "dependencies": {
    "vue": "^2.5.21",
    "vue-router": "^3.0.1"
  },
  "devDependencies": {
    "@vue/cli-plugin-babel": "^3.2.0",
    "@vue/cli-plugin-e2e-cypress": "^3.3.0",
    "@vue/cli-plugin-eslint": "^3.2.0",
    "@vue/cli-plugin-unit-jest": "^3.3.0",
    "@vue/cli-service": "^3.2.0",
    "@vue/eslint-config-prettier": "^4.0.1",
    "@vue/test-utils": "^1.0.0-beta.20",
    "babel-core": "7.0.0-bridge.0",
    "babel-eslint": "^10.0.1",
  }
}
```



@jasongtau @thiagospassos



# Your Turn!

Follow the instructions to complete  
the code exercises and challenges

 Workshop Guide  
[bit.ly/masteringvuejs](https://bit.ly/masteringvuejs)

 Workshop Code  
[bit.ly/vuejs-minnesota-2019](https://bit.ly/vuejs-minnesota-2019)



@jasongtau @thiagospassos



# Mastering Vue.js styling the UI

# Agenda

Frameworks

Icons

Themes

Web Design

# Agenda

Frameworks

Icons

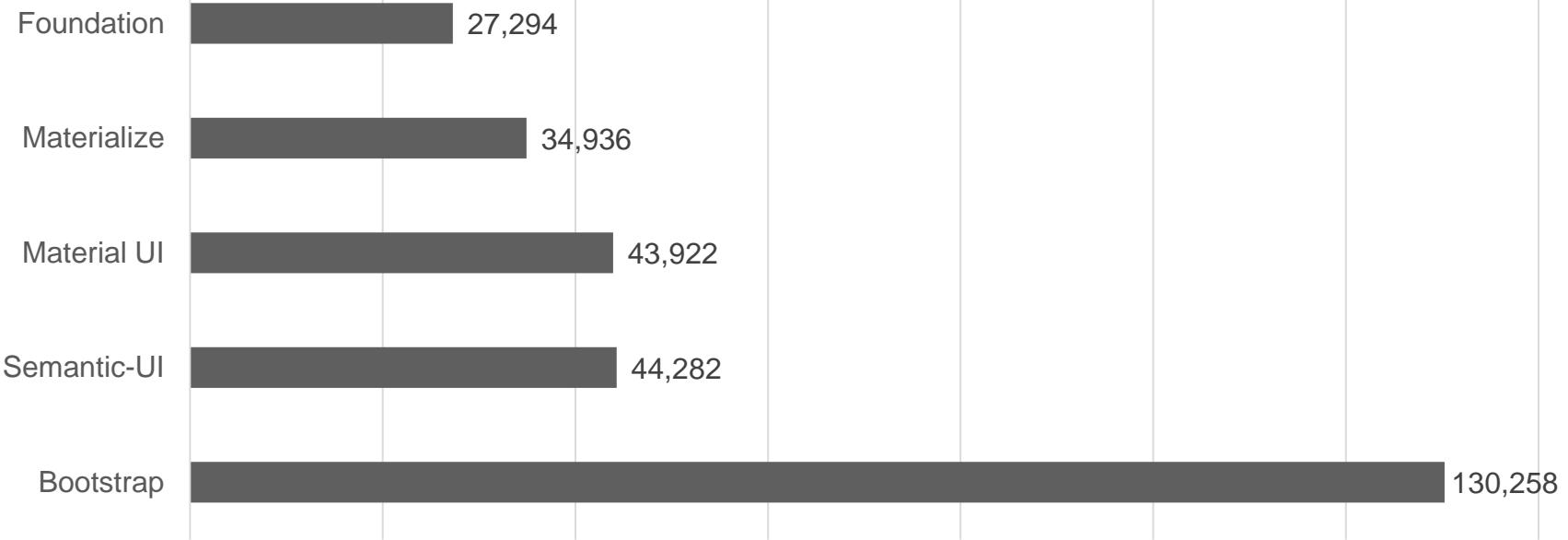
Themes

Web Design

# Frameworks

- Quick and easy to get started
- Professional and consistent appearance
- Many free and paid themes
- Responsive design
- Cross-browser compatibility
- Help is available





Popular Frameworks

# Why Bootstrap?

## Pros

- Responsive and mobile-first
- Extensive components
- Awesome documentation
- Massive community
- Open source

## Cons

- Large library (187KB)
- Steep learning curve



```
<link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css" rel="stylesheet" type="text/css" />
<nav class="navbar navbar-expand-md navbar-dark bg-dark">...</nav>
<main class="container">...</main>
<todo-form>...</todo-form>
<hr>
<todo-list ref="list">...</todo-list>
<h2>Tasks:</h2>
<ul ref="todos" class="list-group">...</ul>
<todo-task ref="task-1517176192142" id="task-1517176192142">...</todo-task>
<todo-task ref="task-151717632897" id="task-151717632897">...</todo-task>
<todo-task ref="task-1517176329096" id="task-1517176329096">...</todo-task>
<todo-task ref="task-1517176334849" id="task-1517176334849">...</todo-task>
```



# Bootstrap Content

- Includes default styles, e.g. headings, paragraphs, lists, tables, forms, ...
- Utilises native fonts
- Consistent across browsers
- Easy to customise



@jasongtau @thiagospassos

# Typography

## Heading 1

## Heading 2

## Heading 3

## Heading 4

## Heading 5

## Heading 6

## Heading with muted text

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor.

## Blockquotes

  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat a ante.

  — Someone famous in *Source Title*

## Example body text

Nullam quis risus eget [urna mollis ornare](#) vel eu leo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

Nullam id dolor id nibh ultricies vehicula.

This line of text is meant to be treated as fine print.

The following is **rendered as bold text**.

The following is *rendered as italicized text*.

An abbreviation of the word attribute is [attr](#).

## Emphasis classes

Fusce dapibus, tellus ac cursus commodo, tortor mauris nibh.

[Nullam id dolor id nibh ultricies vehicula ut id elit.](#)

Pellentesque ornare sem lacinia quam venenatis vestibulum.

[Etiam porta sem malesuada magna mollis euismod.](#)

[Donec ullamcorper nulla non metus auctor fringilla.](#)

Duis mollis, est non commodo luctus, nisi erat porttitor ligula.

[Maecenas sed diam eget risus varius blandit sit amet non magna.](#)

[Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat a ante.](#)

— Someone famous in *Source Title*



# Bootstrap Layout

- Includes two basic layout elements:
  - Containers
  - Grid system
- Responsive breakpoints, adapts according to device capabilities



@jasongtau @thiagospassos

Navbar Home Link Disabled Dropdown

Search

Hello, world!

This is a template for a simple marketing or informational website. It includes a large callout called a jumbotron and three supporting pieces of content. Use it as a starting point to create something more unique.

Learn more »

**Heading**

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

**Heading**

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

**Heading**

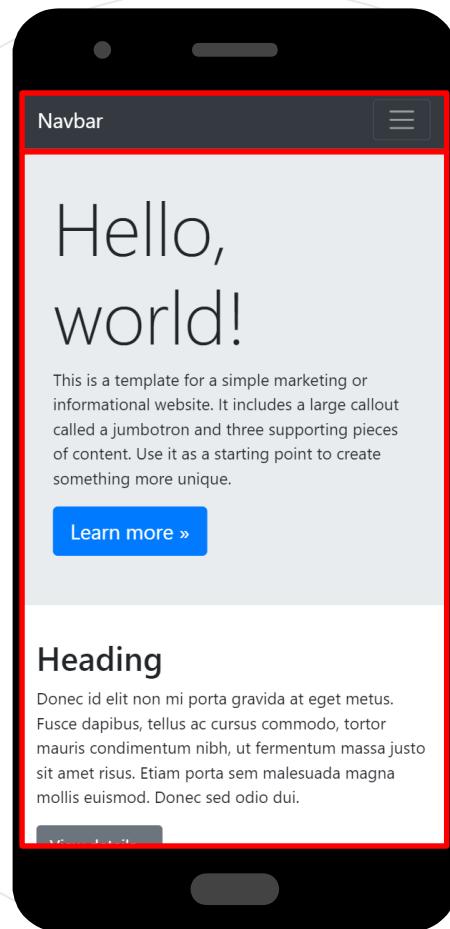
Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.

[View details »](#)

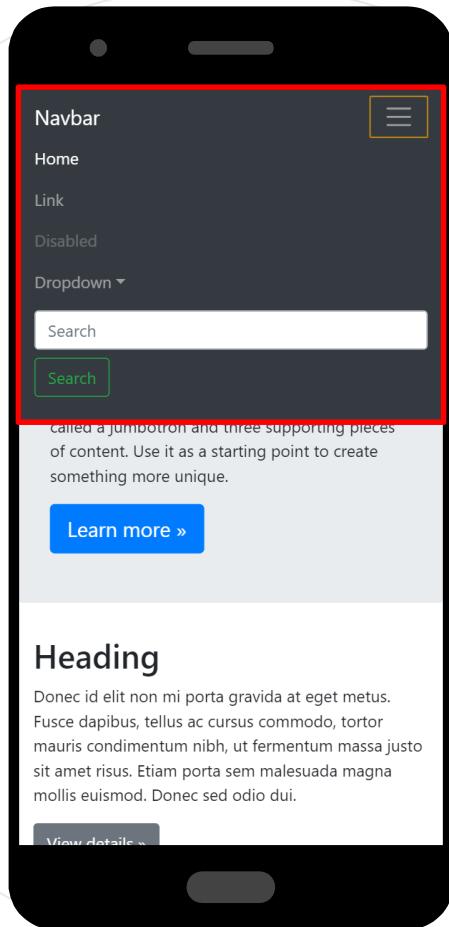
© Company 2017-2018



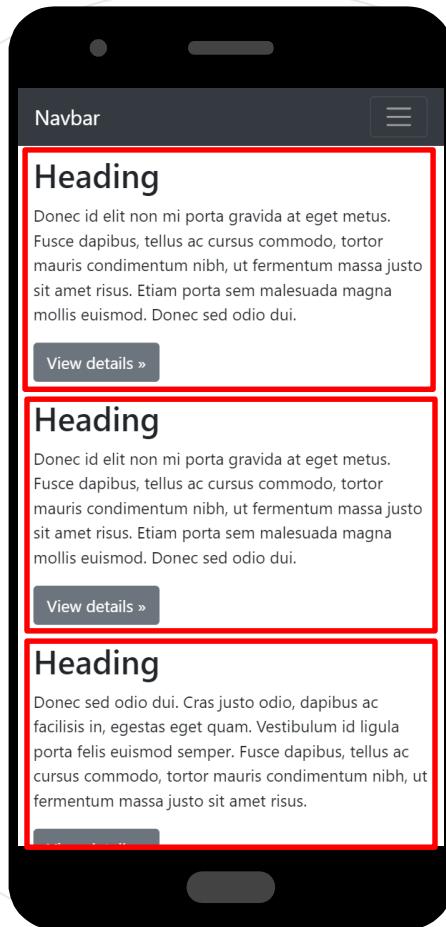
@jasongtau @thiagospassos



@jasongtau @thiagospassos



@jasongtau @thiagospassos



@jasongtau @thiagospassos

# Bootstrap Components

- Wide range of components
- Including
  - Navigation
  - Forms
  - Tables
  - Toasts
  - Modals
- So much more!



@jasongtau @thiagospassos

# Tables

Type	Column heading	Column heading	Column heading
Active	Column content	Column content	Column content
Default	Column content	Column content	Column content
Primary	Column content	Column content	Column content
Secondary	Column content	Column content	Column content
Success	Column content	Column content	Column content
Danger	Column content	Column content	Column content
Warning	Column content	Column content	Column content
Info	Column content	Column content	Column content
Light	Column content	Column content	Column content
Dark	Column content	Column content	Column content

# Forms

## Legend

Email email@example.com

Email address

 ✉

We'll never share your email with anyone else.

Password

 ?

Example select

 ▼

Example multiple select

- 1
- 2
- 3
- 4

▼

Example textarea

File input

Disabled input

Disabled input here...

Readonly input

Readonly input here...

Valid input

 ✓

Success! You've done it.

Invalid input

 ✗

Sorry, that username's taken. Try another?

Large input

 .form-control-lg

Default input

 .form-control

Small input

 .form-control-sm

Input addons



@jasongtau @thiagospassos

# Navs

## Tabs

Home Profile Disabled Dropdown ▾

Raw denim you probably haven't heard of them jean shorts Austin. Nesciunt tofu stumptown aliqua, retro synth master cleanse. Mustache cliche tempor, williamsburg carles vegan helvetica. Reprehenderit butcher retro keffiyeh dreamcatcher synth. Cosby sweater eu banh mi, qui irure terry richardson ex squid. Aliquip placeat salvia cillum iphone. Seitan aliquip quis cardigan american apparel, butcher voluptate nisi qui.

## Pills

Active Dropdown ▾ Link Disabled

Active

Dropdown ▾

Link

Disabled

## Breadcrumbs

Home

Home / Library

Home / Library / Data

## Pagination

« 1 2 3 4 5 »

« 1 2 3 4 5 »

« 1 2 3 4 5 »



@jasongtau @thiagospassos

## Cards

Header

### Primary card title

Some quick example text to build on the card title and make up the bulk of the card's content.

Header

### Secondary card title

Some quick example text to build on the card title and make up the bulk of the card's content.

Header

### Success card title

Some quick example text to build on the card title and make up the bulk of the card's content.

Header

### Danger card title

Header

### Primary card title

Some quick example text to build on the card title and make up the bulk of the card's content.

Header

### Secondary card title

Some quick example text to build on the card title and make up the bulk of the card's content.

Header

### Success card title

Some quick example text to build on the card title and make up the bulk of the card's content.

Header

### Danger card title

## Card header

### Special title treatment

Support card subtitle

Image

Some quick example text to build on the card title and make up the bulk of the card's content.

Cras justo odio

Dapibus ac facilisis in

Vestibulum at eros

[Card link](#) [Another link](#)

2 days ago



# Bootstrap Vue

- Bootstrap components require the use of JavaScript
- You could import required libraries, i.e. jQuery and Popper
- A better choice is to use Bootstrap Vue
- Bootstrap Vue provides components and directives for Bootstrap



@jasongtau @thiagospassos

# Demo: Bootstrap Vue

- Installing & Configuring
- Working with Components



@jasongtau @thiagospassos

# Agenda

Frameworks

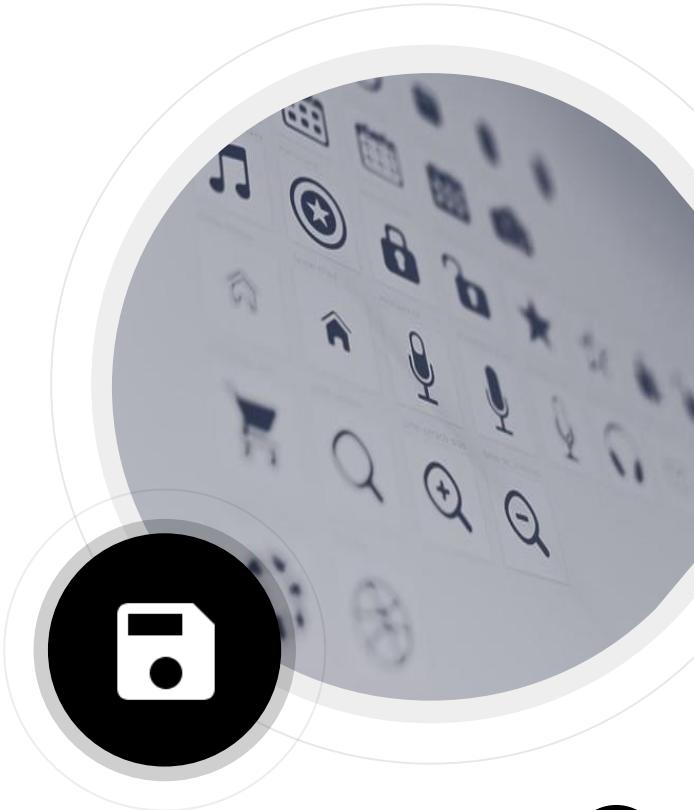
Icons

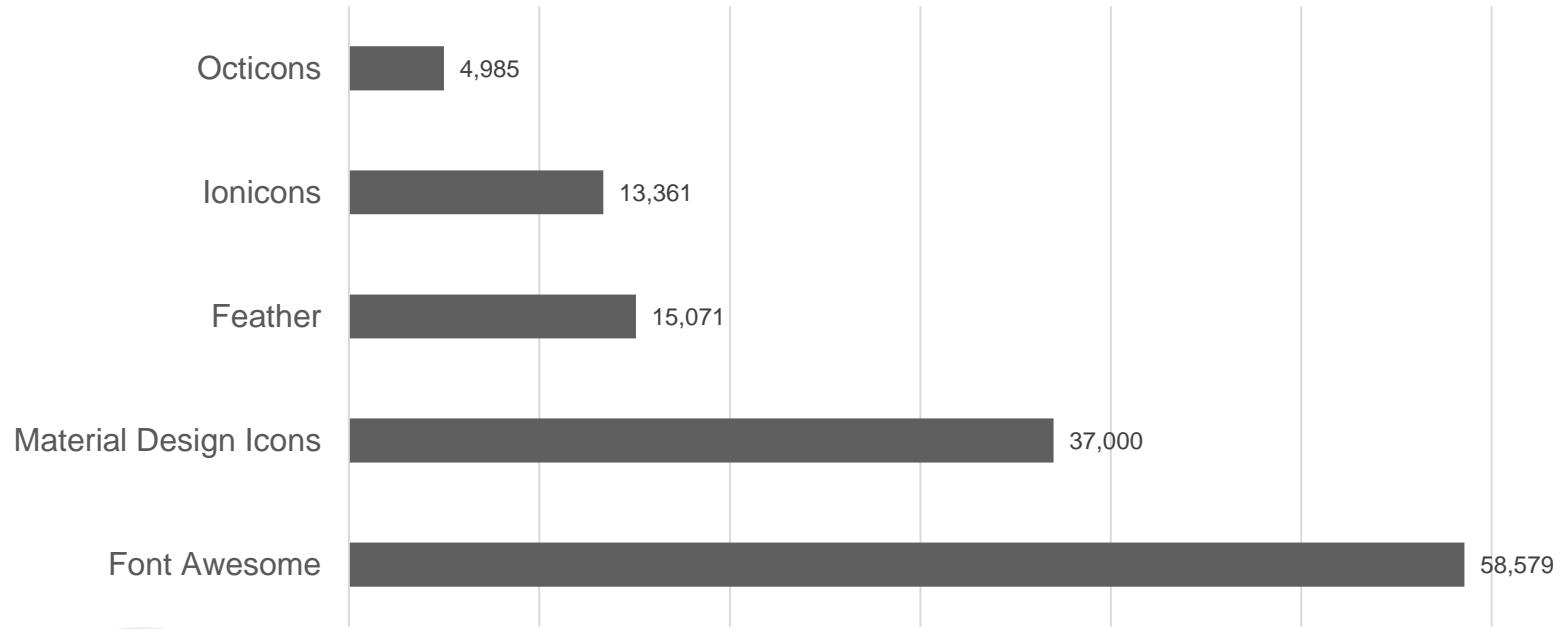
Themes

Web Design

# Icons

- Support, or completely replace text
- Use sparingly, and thoughtfully
- Stick to convention
- Know advanced features





Popular Icon Libraries

# Demo: Icons

- Install Feather Icons
- Updating the UI



@jasongtau @thiagospassos

# Agenda

Frameworks

Icons

Themes

Web Design

# Themes

- Sites built on Bootstrap look great
- Unfortunately, they look the same as every other Bootstrap site...
- Improve your site's appearance by customising or replacing the theme



# Sass

- CSS with superpowers
- Use features that don't exist in CSS
- Features such as:
  - Variables
  - Nesting
  - Partials
  - Mixins
- A couple of examples...



# Sass Variables

- Store information you want to use throughout your stylesheet
- Such as colours and fonts
- Use the \$ symbol for variables

```
$font-stack: Helvetica, sans-serif;  
$primary-color: #333;  
  
body {  
  font: 100% $font-stack;  
  color: $primary-color;  
}
```



# Sass Nesting

- Nest CSS selectors to provide a clear visual hierarchy

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
  }  
  
  li { display: inline-block; }  
}
```



# Demo: Working with Sass

- Enabling Sass
- Customising the site theme



@jasongtau @thiagospassos

# Finding Themes

- Upgrade the sites appearance with a new theme
- Free themes
  - [bootswatch.com](http://bootswatch.com)
- Paid themes
  - [themes.getbootstrap.com](http://themes.getbootstrap.com)
  - [wrapbootstrap.com](http://wrapbootstrap.com)



# Demo: Themes

- Replacing the default theme



# Agenda

Frameworks

Icons

Themes

Web Design

# Web Design

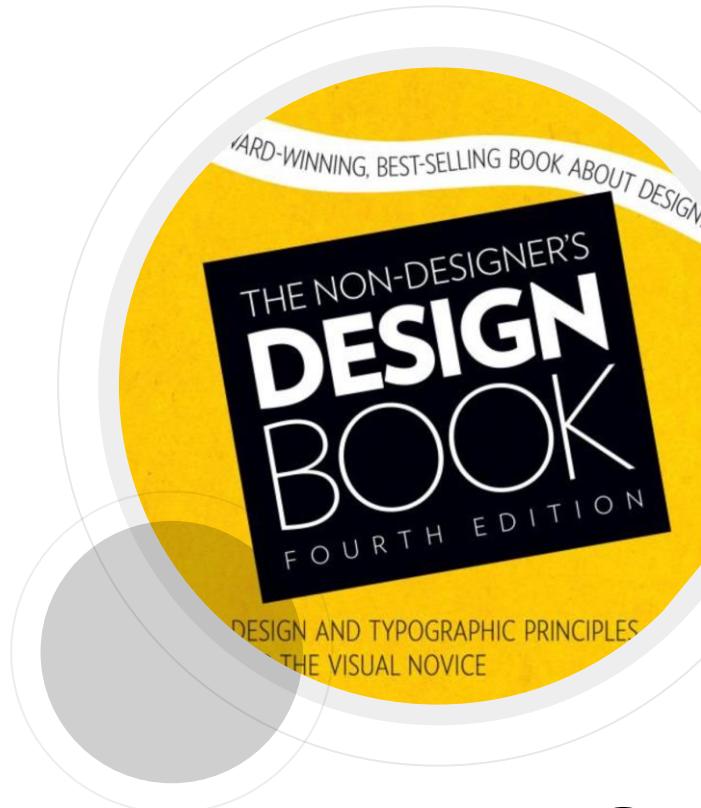
- Developers are not web designers
- Knowledge of frameworks, themes, and icons is a great start
- If you're ready, you can improve your skills with the follow resources:



# The Non-Designer's Design Book

- Design and typographic principles for the visual novice

By Robin Williams



@jasongtau @thiagospassos

# Don't Make Me Think!

- A Common Sense Approach to Web Usability

By Steve Krug



@jasongtau @thiagospassos

# Your Turn!

Follow the instructions to complete  
the code exercises and challenges

 Workshop Guide  
[bit.ly/masteringvuejs](https://bit.ly/masteringvuejs)

 Workshop Code  
[bit.ly/vuejs-minnesota-2019](https://bit.ly/vuejs-minnesota-2019)



@jasongtau @thiagospassos



# Vue.js

## Consuming APIs

# Agenda

Getting Started

Configuration

Interceptors

Put all together

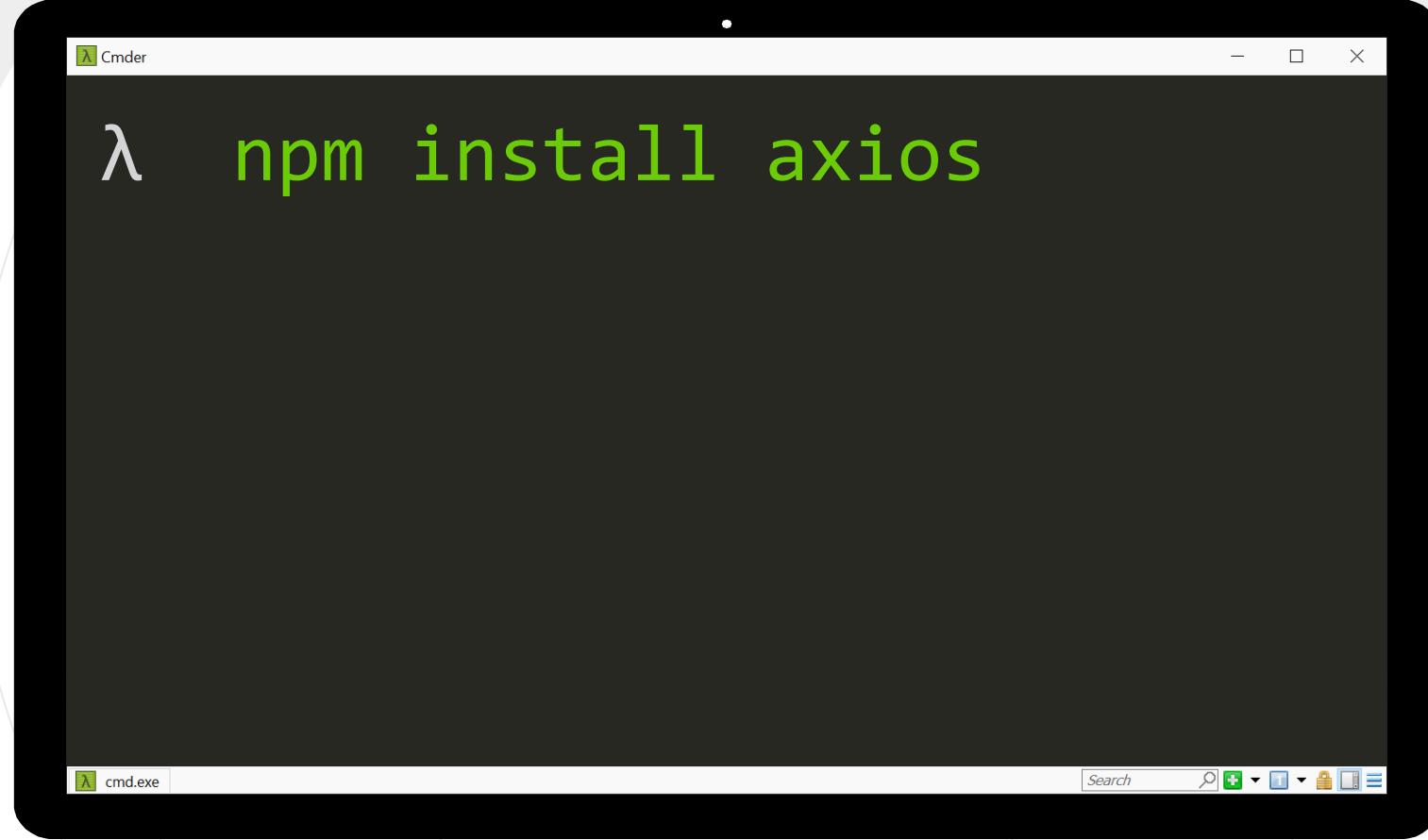
# Agenda

Getting Started

Configuration

Interceptors

Put all together



@jasongtau @thiagospassos

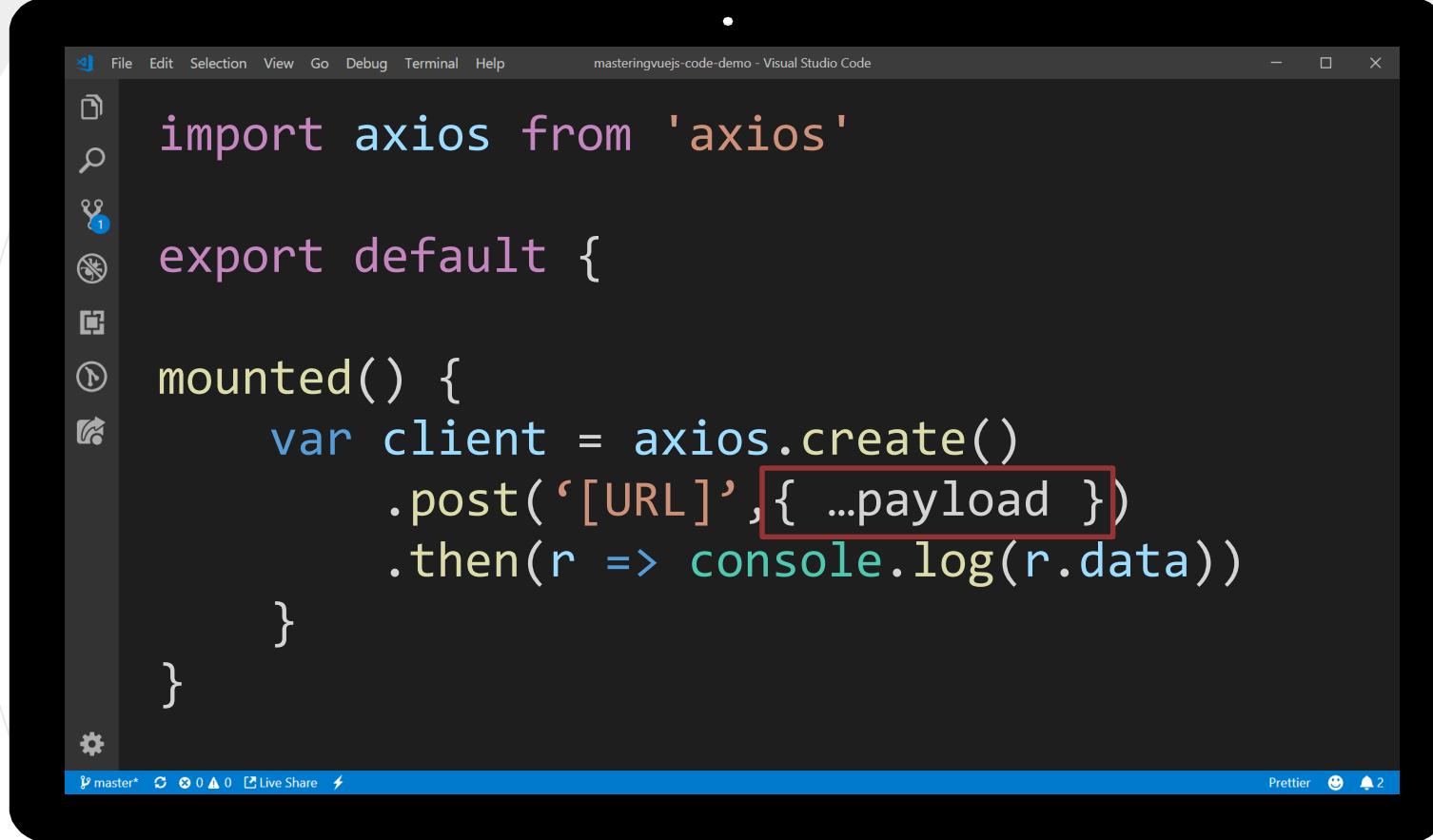
```
import axios from 'axios'

export default {

  mounted() {
    var client = axios.create()
      .get('[URL]')
      .then(r => console.log(r.data))
  }
}
```



@jasongtau @thiagospassos

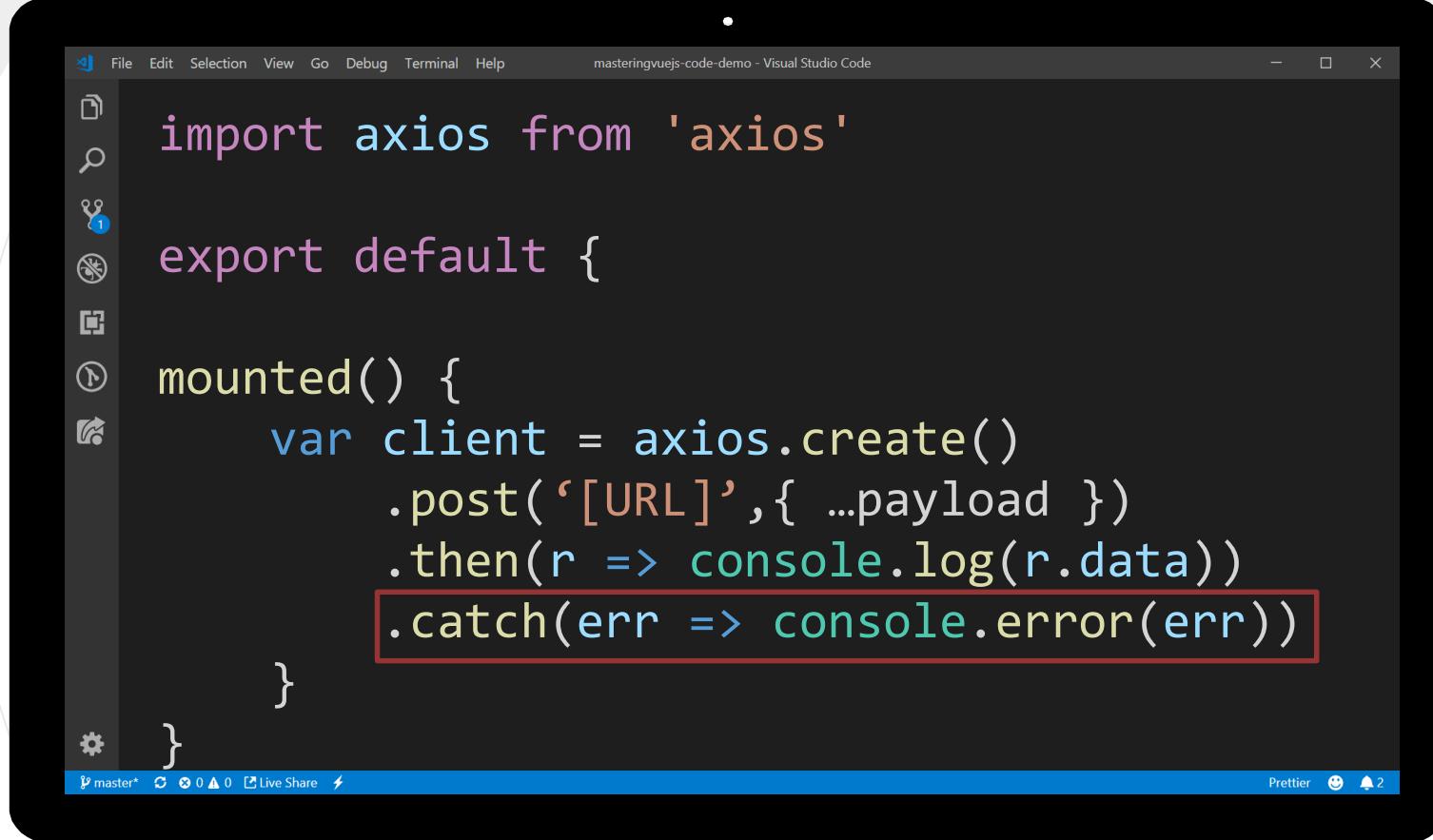


```
import axios from 'axios'

export default {

  mounted() {
    var client = axios.create()
      .post('[URL]', { ...payload })
      .then(r => console.log(r.data))
  }
}
```





```
import axios from 'axios'

export default {

  mounted() {
    var client = axios.create()
      .post('[URL]', { ...payload })
      .then(r => console.log(r.data))
      .catch(err => console.error(err))
  }
}
```



# Agenda

Getting Started

Configuration

Interceptors

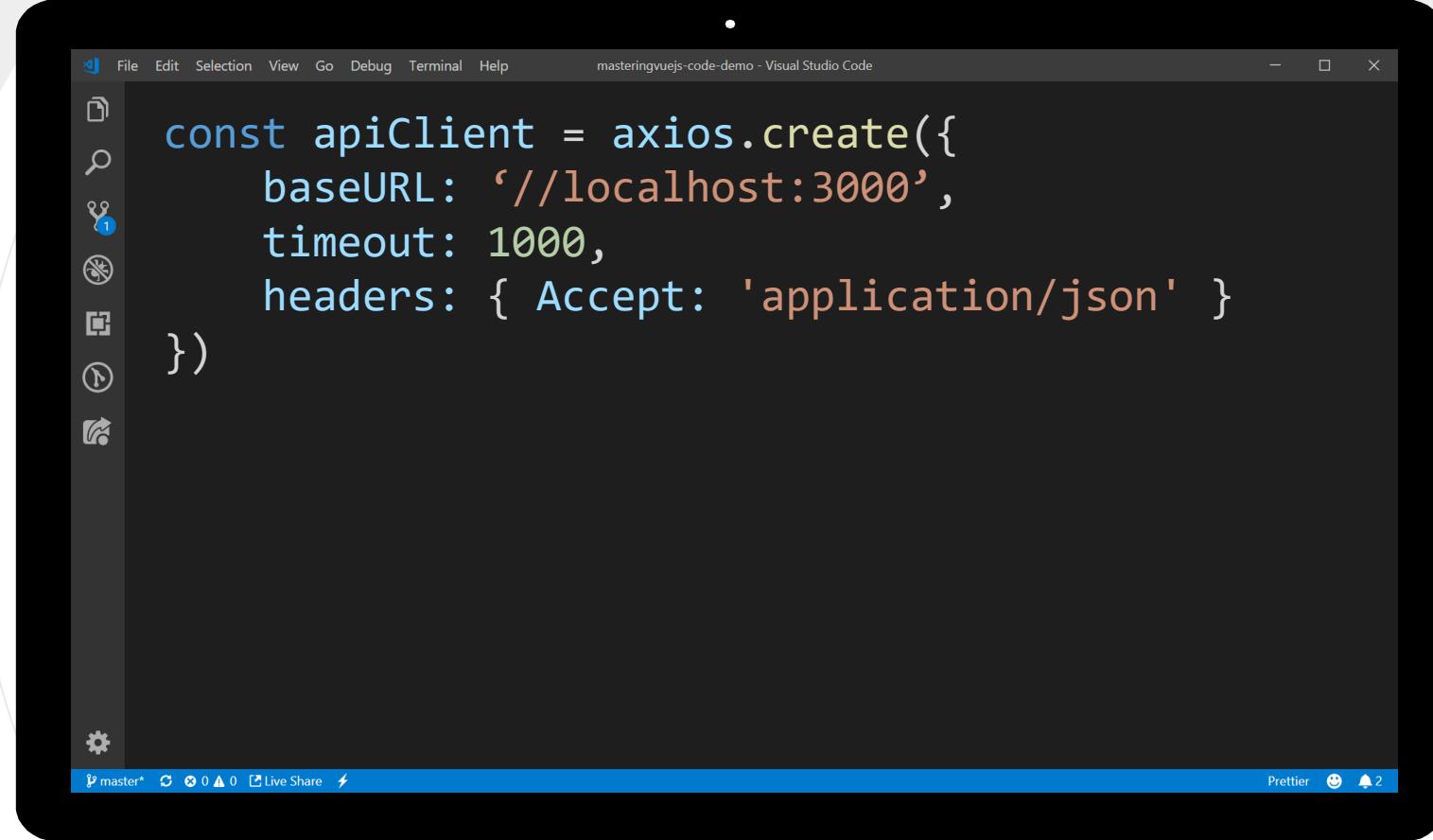
Put all together

```
const apiClient = axios.create({
  baseURL: '//localhost:3000',
})

export const SupplierService = {
  getAll() {
    return apiClient.get('/suppliers')
  },
  get(id) {
    return apiClient.get('/suppliers/' + id)
  },
}
```



@jasongtau @thiagospassos



A screenshot of Visual Studio Code with a dark theme. The main area displays the following JavaScript code:

```
const apiClient = axios.create({
  baseURL: '//localhost:3000',
  timeout: 1000,
  headers: { Accept: 'application/json' }
})
```

The sidebar on the left contains several icons: a file icon, a search icon, a preview icon with a '1' notification, a refresh icon, a copy icon, a circular arrow icon, and a settings gear icon.

The bottom status bar shows the current branch as 'master\*', file statistics (0 changes, 0 additions, 0 deletions), a 'Live Share' icon, and extension status for 'Prettier' (formatting applied).



@jasongtau @thiagospassos

```
File Edit Selection View Go Debug Terminal Help  
masteringvuejs-code-demo - Visual Studio Code  
axios.defaults.baseUrl = '//localhost:3000'  
axios.defaults.headers  
  .common['Authorization'] = AUTH_TOKEN;  
  
const apiClient = axios.create({  
  timeout: 1000,  
  headers: { Accept: 'application/json' }  
})
```

master\* 0 0 0 Live Share Prettier 2



@jasongtau @thiagospassos

```
File Edit Selection View Go Debug Terminal Help  
masteringvuejs-code-demo - Visual Studio Code  
axios.defaults.baseUrl = '//localhost:3000'  
  
const apiClient = axios.create({  
    timeout: 1000,  
    headers: { Accept: 'application/json' }  
})  
  
apiClient.defaults.headers  
.common['Authorization'] = AUTH_TOKEN;
```



@jasongtau @thiagospassos

# Agenda

Getting Started

Configuration

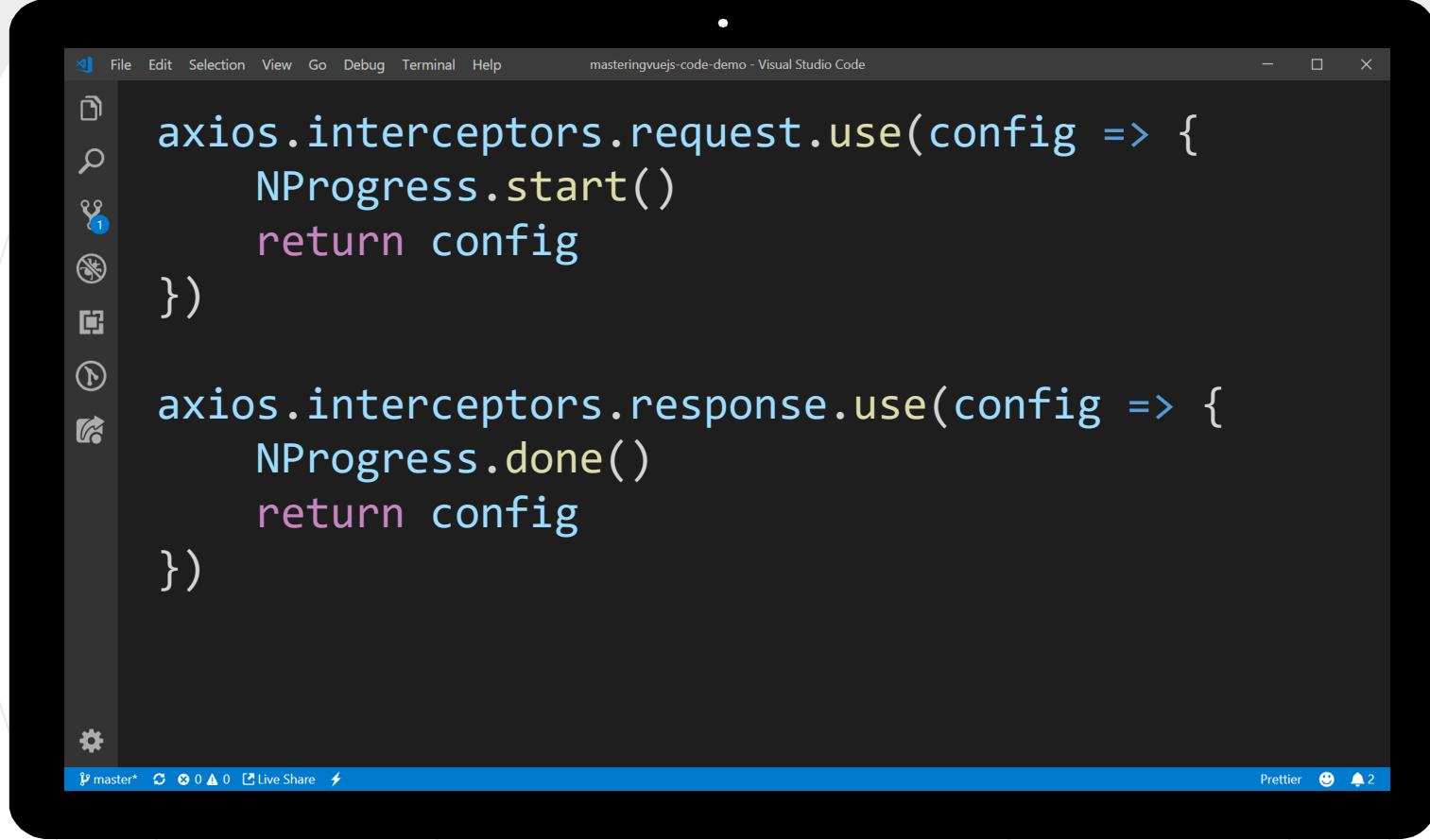
Interceptors

Put all together

```
File Edit Selection View Go Debug Terminal Help  
masteringvuejs-code-demo - Visual Studio Code  
axios.interceptors.request.use(config => {  
    // TODO  
    return config  
})  
  
axios.interceptors.response.use(config => {  
    // TODO  
    return config  
})
```



@jasongtau @thiagospassos



```
File Edit Selection View Go Debug Terminal Help
masteringvuejs-code-demo - Visual Studio Code
axios.interceptors.request.use(config => {
  NProgress.start()
  return config
})

axios.interceptors.response.use(config => {
  NProgress.done()
  return config
})
```

master\* 0 0 0 Live Share Prettier 2



@jasongtau @thiagospassos

A screenshot of Visual Studio Code with a dark theme. The main area displays the following JavaScript code:

```
axios.interceptors.request.use(config => {
  if (AuthService.token()) {
    config.headers.authorization =
      'Bearer ' + AuthService.token()
  }
  return config
})
```

The code uses the Axios library's interceptors to add a Bearer token to the Authorization header of every request if an authentication service has provided one.



@jasongtau @thiagospassos

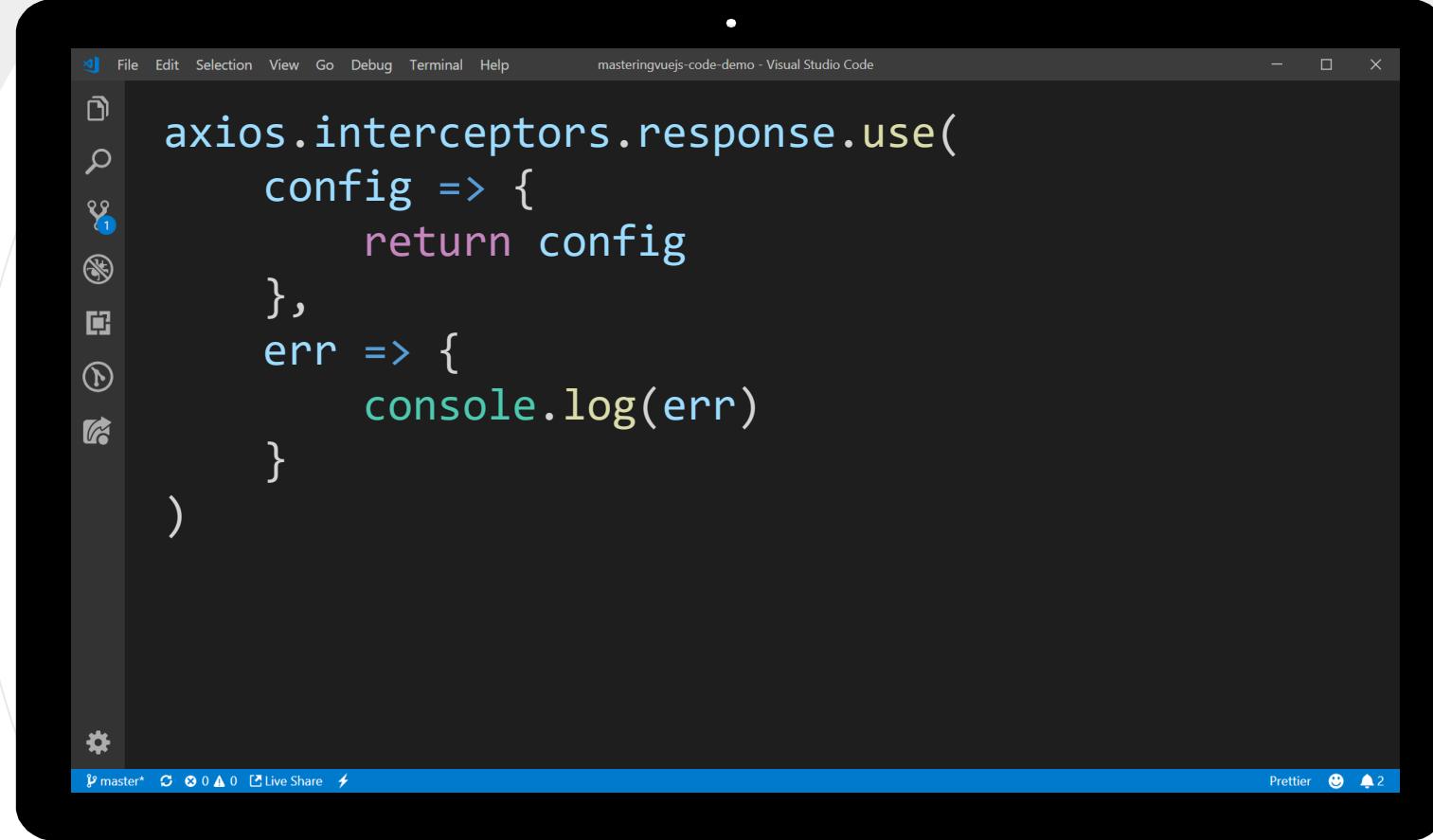
A screenshot of the Visual Studio Code interface. The title bar reads "masteringvuejs-code-demo - Visual Studio Code". The left sidebar contains icons for file operations, search, and other tools. The main editor area displays the following code:

```
axios.interceptors.request.use(  
    config => {  
        return config  
    },  
    err => {  
        console.log(err)  
    }  
)
```

The status bar at the bottom shows "master\*" and "Live Share" on the left, and "Prettier" and a notification icon on the right. A gear icon is also visible in the bottom-left corner of the editor area.



@jasongtau @thiagospassos

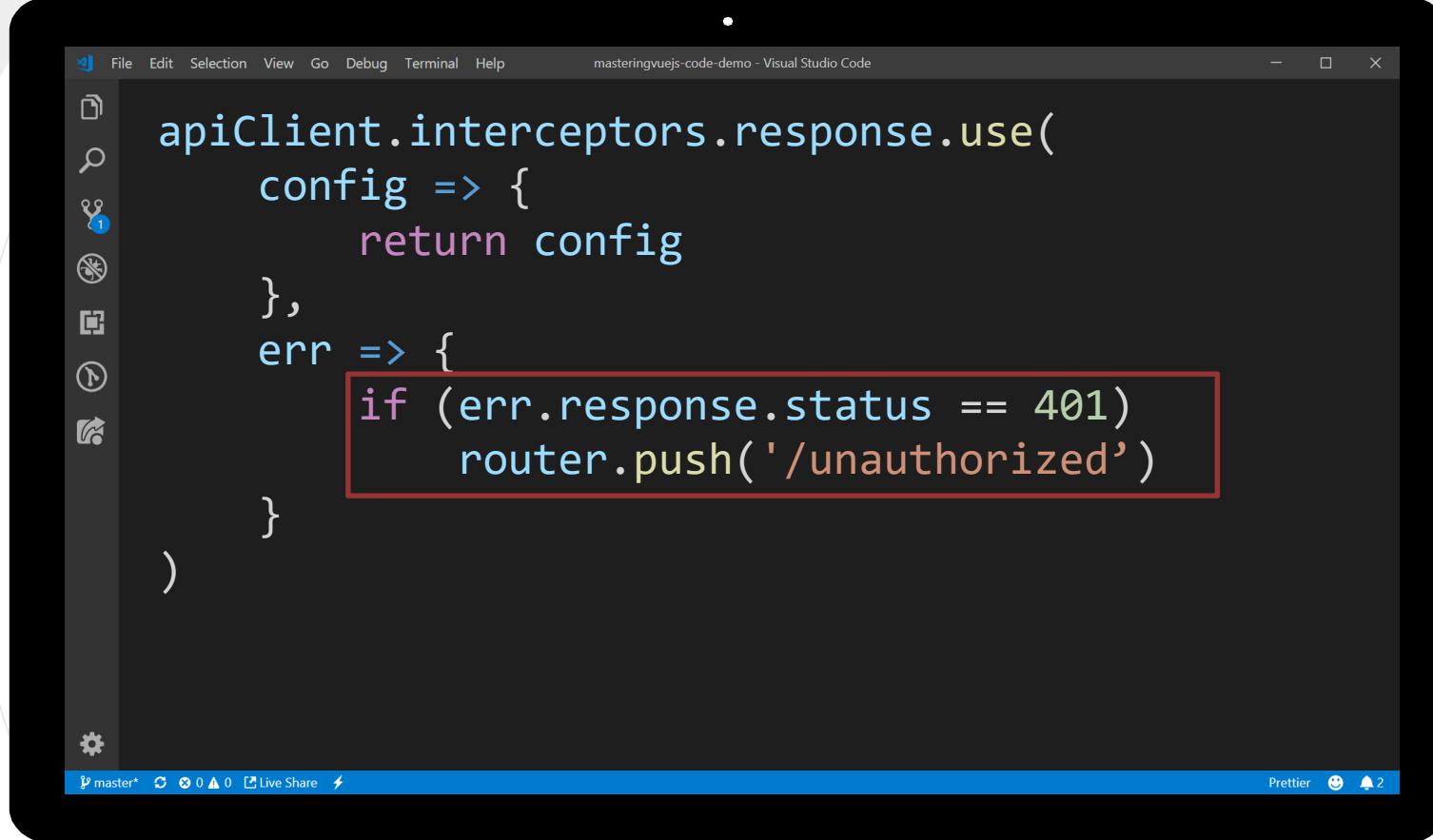


A screenshot of Visual Studio Code running on a Mac. The interface is in dark mode. The top menu bar includes File, Edit, Selection, View, Go, Debug, Terminal, and Help. The title bar says "masteringvuejs-code-demo - Visual Studio Code". The left sidebar contains icons for file operations like Open, Save, Find, and others, with a blue badge showing '1'. The main editor area contains the following JavaScript code:

```
axios.interceptors.response.use(  
    config => {  
        return config  
    },  
    err => {  
        console.log(err)  
    }  
)
```

The bottom status bar shows the current branch is "master\*", with 0 changes, 0 issues, and 0 warnings. It also displays "Live Share" and "Prettier" status. There are also icons for GitHub, Twitter, and a bell.





```
apiClient.interceptors.response.use(
  config => {
    return config
  },
  err => {
    if (err.response.status == 401)
      router.push('/unauthorized')
  }
)
```



# Agenda

Getting Started

Configuration

Interceptors

Put all together



# Your Turn!

Follow the instructions to complete  
the code exercises and challenges

 Workshop Guide  
[bit.ly/masteringvuejs](https://bit.ly/masteringvuejs)

 Workshop Code  
[bit.ly/vuejs-minnesota-2019](https://bit.ly/vuejs-minnesota-2019)



@jasongtau @thiagospassos



# Mastering Vue.js State Management

# Agenda

What is Vuex?

When to use Vuex?

Accessing State

Changing State

# Agenda

What is Vuex?

When to use Vuex?

Accessing State

Changing State

# What is Vuex?

- Reactive state management pattern and library
- Serves as a centralised store for all components
- State changes tracked and predictable
- Leverages reactivity system for simple and efficient updates
- Supports advanced features



@jasongtau @thiagospassos

# Agenda

What is Vuex?

When to use Vuex?

Accessing State

Changing State

# When to use Vuex?

- Sharing data between more than two components
- Sharing data between components with no direct parent-child relationship
- Advanced debugging features such as time-travel
- Short-term vs. long-term productivity?



# Vuex Big Picture

Components

Actions

Mutations

State

External API

Vuex

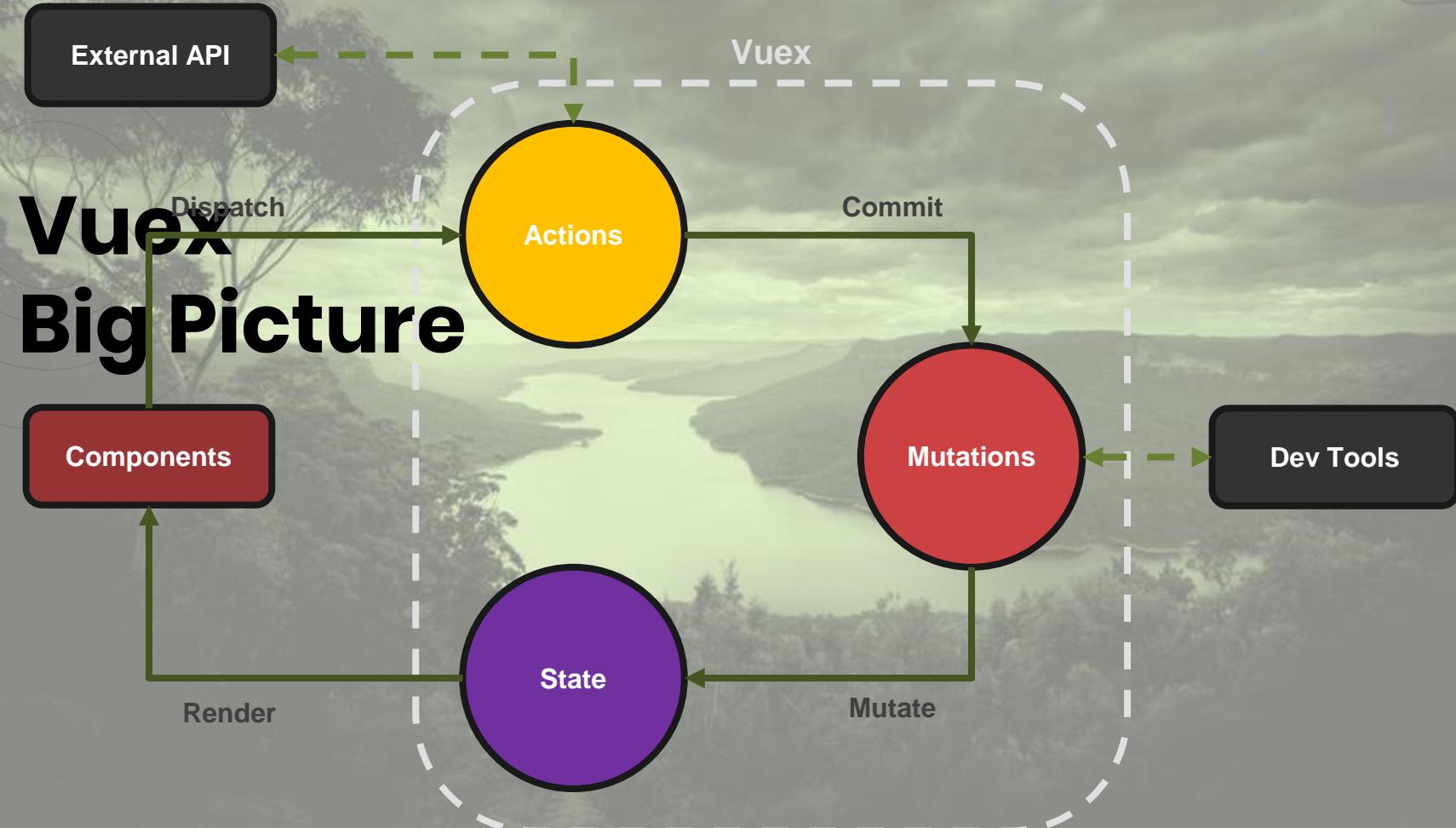
Dev Tools

Render

Mutate

Commit

Dispatch



# Agenda

What is Vuex?

When to use Vuex?

Accessing State

Changing State

# Demo: Accessing State

- Installing and configuring Vuex
- Working with Vuex state and getters
- Displaying shared state within components



# Core Concepts: State

- Single object containing all shared state
- Straight-forward to locate state

```
state: {  
  userName: 'Jason',  
  roles: [ 'Administrator', 'Contributor' ]  
}
```



@jasongtau @thiagospassos

# Core Concepts: Getters

- Computed properties for the store
- Result is cached based on dependencies

```
getters: {  
  isAdministrator: (state) => {  
    return state.roles.contains(r => r === 'Administrator')  
  }  
}
```



@jasongtau @thiagospassos

# Agenda

What is Vuex?

When to use Vuex?

Accessing State

Changing State

# Demo: Changing State

- Working with Vuex actions and mutations
- Tracking success and error notifications



@jasongtau @thiagospassos

# Core Concepts: Mutations

- Used to commit and track state changes
- Mutations are synchronous
- Mutations should only be called from actions

```
mutations: {
  addUser: (state, payload) => {
    return state.users.push(payload)
  }
}
```



@jasongtau @thiagospassos

# Core Concepts: Actions

- Contain business logic and commit mutations
- Actions are asynchronous
- Always put mutations within actions

```
actions: {  
  addUser: ({ commit }, user) => {  
    return UserService.addUser(user).then(  
      () => commit('addUser', user)  
    )  
  }  
}
```



# Vuex Big Picture

Components

Actions

Mutations

State

External API

Vuex

Dev Tools

Render

Mutate

Commit

Dispatch

# Key Points

- Serves as a centralised store for all components
- Share application level state across many components
- Efficiently access derived state from multiple components
- ...



@jasongtau @thiagospassos

# Key Points

- Effectively encapsulate logic to manage state changes
- Easily debug state changes with snapshots and time-travel



@jasongtau @thiagospassos

# Your Turn!

Follow the instructions to complete  
the code exercises and challenges

 Workshop Guide  
[bit.ly/masteringvuejs](https://bit.ly/masteringvuejs)

 Workshop Code  
[bit.ly/vuejs-minnesota-2019](https://bit.ly/vuejs-minnesota-2019)



@jasongtau @thiagospassos



# Vue.js

## Packaging and deploying

# Agenda

Environment variables

Tooling

Continuous Integration

Continuous Deployment

# Agenda

Environment variables

Tooling

Continuous Integration

Continuous Deployment

# What are my options?

Update variables at build time with the Vue.js' recommended approach

Build once and deploy many updating variables at deployment time and reading them at runtime

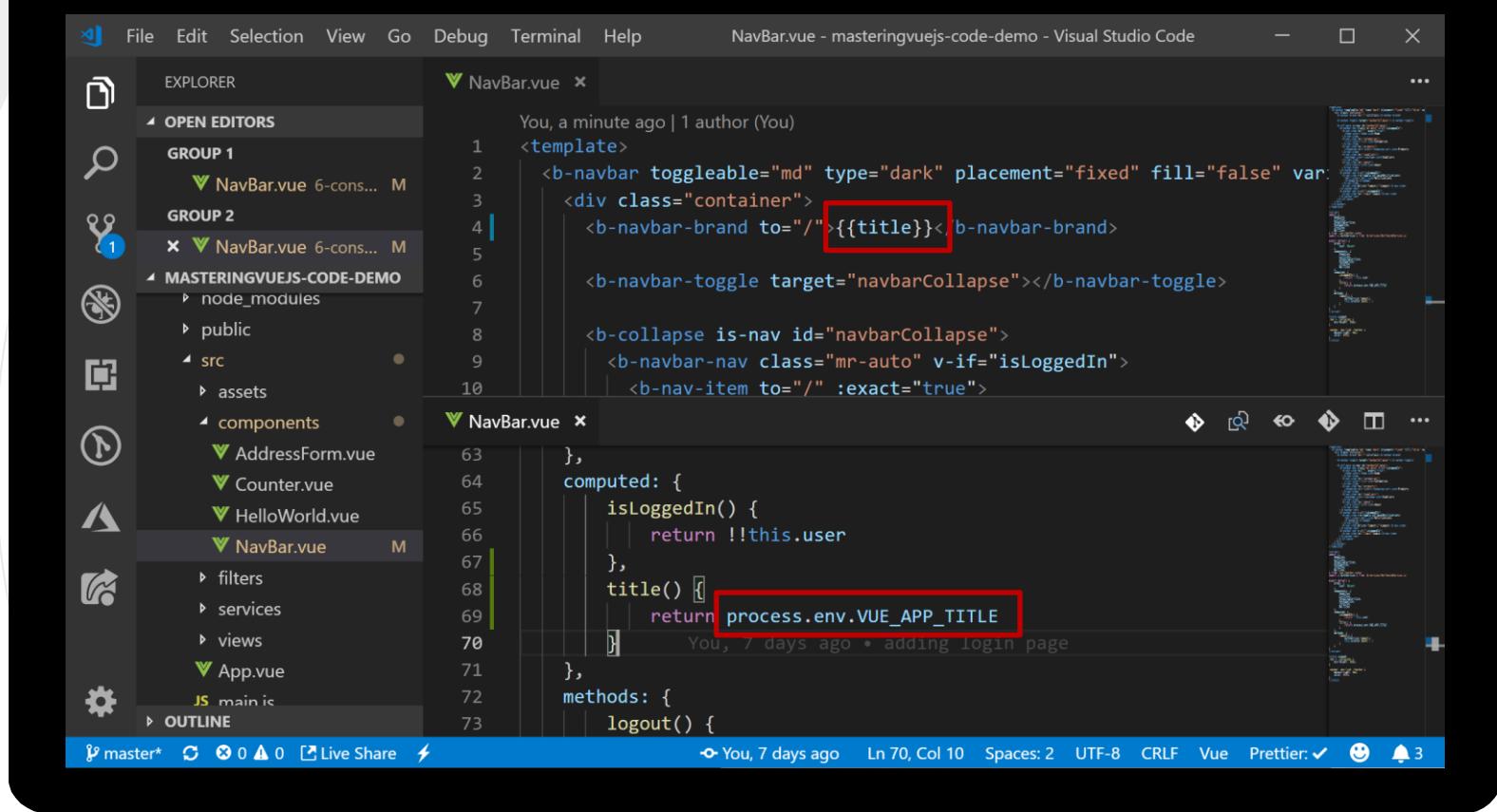


The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Debug, Terminal, Help
- Title Bar:** .env - masteringvuejs-code-demo - Visual Studio Code
- Explorer Bar:** Shows the project structure with files like .env, .env.production, .eslintrc.js, .gitignore, .prettierrc.js, auth-json-server.js, babel.config.js, cypress.json, db.json, jest.config.js, package-lock.json, and package.json.
- Editor:** The .env file is open, showing environment variables:

```
You, 21 hours ago | 1 author (you)  
1 VUE_APP_BASE_URL="//localhost:3000"  
2 VUE_APP_TITLE="This is my awesome development enviroment"
```
- Annotations:** A central box labeled "Mode" branches into three boxes:
  - production**: npm run build  
npm run test:e2e
  - development**: npm run serve
  - test**: npm run test:unit
- Bottom Status Bar:** master, 0 ▲ 0, Live Share, You, 21 hours ago, Ln 2, Col 58, Spaces: 4, UTF-8, CRLF, Plain Text, smiley face, bell icon





You, a minute ago | 1 author (You)

```
<template>
  <b-navbar toggleable="md" type="dark" placement="fixed" fill="false" var="NavBar">
    <div class="container">
      <b-navbar-brand to="/">{{title}}</b-navbar-brand>
      <b-navbar-toggle target="navbarCollapse"></b-navbar-toggle>
      <b-collapse is-nav id="navbarCollapse">
        <b-navbar-nav class="mr-auto" v-if="isLoggedIn">
          <b-nav-item to="/" :exact="true">
```

You, / days ago \* adding login page

```
  },
  computed: {
    isLoggedIn() {
      return !!this.user
    },
    title() {
      return process.env.VUE_APP_TITLE
    }
  },
  methods: {
    logout() {
```



A screenshot of the Visual Studio Code interface. The left sidebar shows a file tree with a red box highlighting the 'config.json' file under the 'public' directory. The main editor area displays the JSON content of 'config.json'. The status bar at the bottom shows the file path 'master\*', line count 'Ln 4, Col 1', and other settings.

config.json - masteringvuejs-code-demo - Visual Studio Code

EXPLORER

OPEN EDITORS

MASTERINGVUEJS-CODE-DEMO

node\_modules

public

static

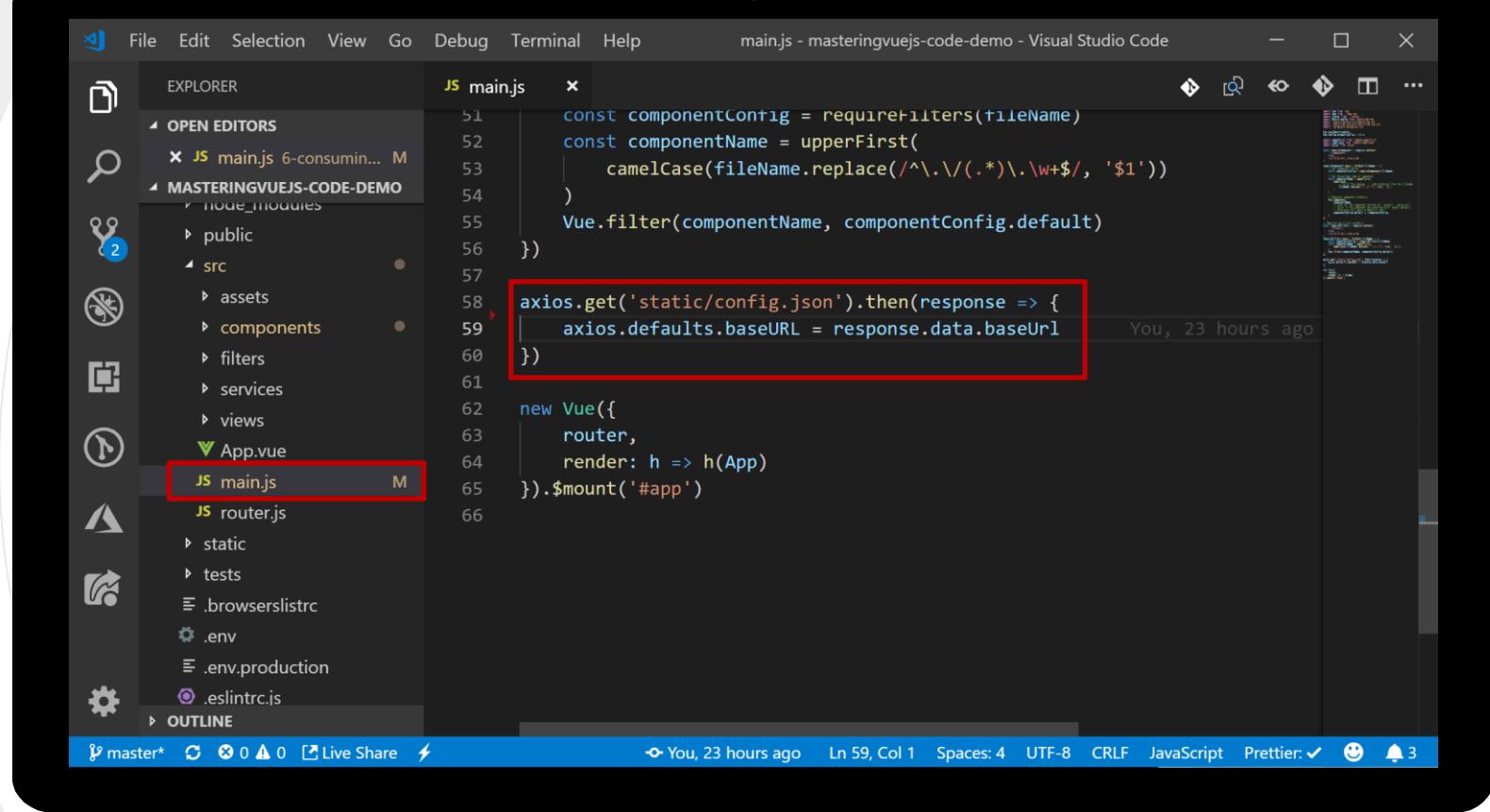
config.json

You, 23 hours ago | 1 author (You)

```
{  
  "baseUrl": "//localhost:3000"  
}
```

Ln 4, Col 1 Spaces: 4 UTF-8 CRLF JSON Prettier: ✓





File Edit Selection View Go Debug Terminal Help main.js - masteringvuejs-code-demo - Visual Studio Code

EXPLORER

- OPEN EDITORS
- JS main.js M
- MASTERINGVUEJS-CODE-DEMO
- node\_modules
- public
- src
  - assets
  - components
  - filters
  - services
  - views
- App.vue
- JS main.js M
- JS router.js
- static
- tests
- .browserslistrc
- .env
- .env.production
- .eslintrc.js
- OUTLINE

main.js

```
const componentConfig = requireFilters(fileName)
const componentName = upperFirst(
  camelCase(fileName.replace(/^\.\.\//(.*)\.\w+$/, '$1'))
)
Vue.filter(componentName, componentConfig.default)

axios.get('static/config.json').then(response => {
  axios.defaults.baseURL = response.data.baseUrl
})

new Vue({
  router,
  render: h => h(App)
}).$mount('#app')
```

You, 23 hours ago

master\* 0 ▲ 0 Live Share You, 23 hours ago Ln 59, Col 1 Spaces: 4 CRLF JavaScript Prettier: ✓ 3



203



@jasongtau @thiagospassos

# Agenda

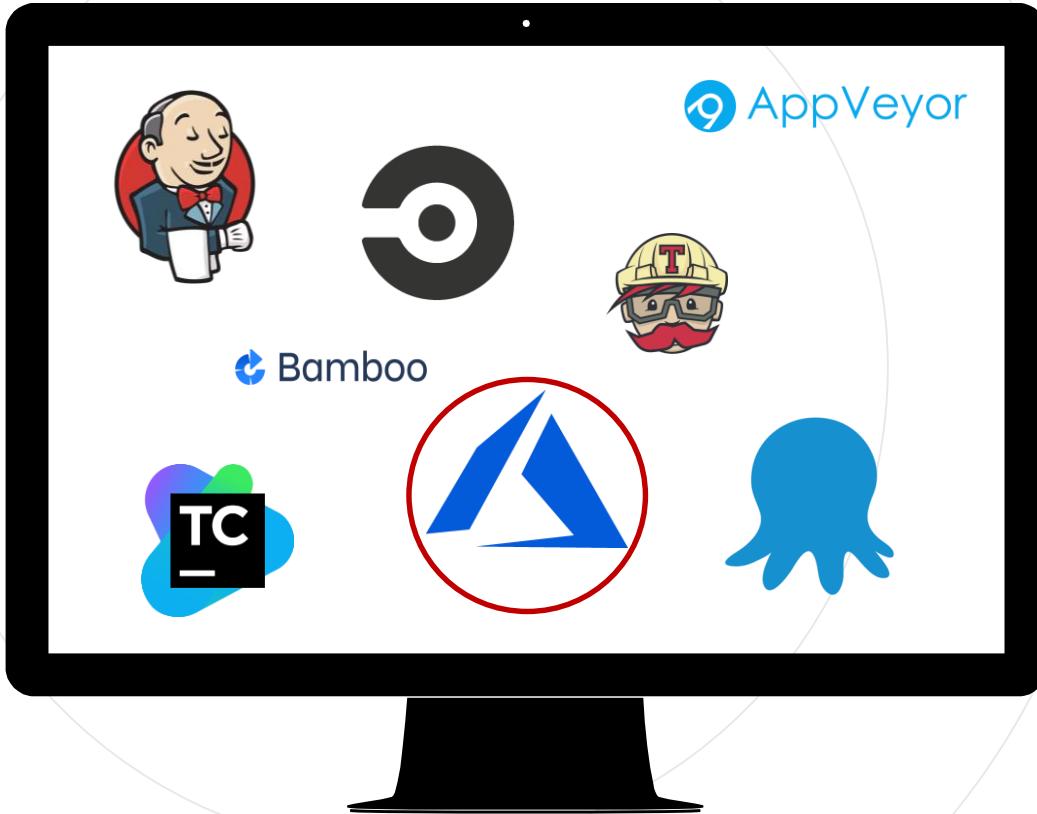
Environment variables

Tooling

Continuous Integration

Continuous Deployment

# Tools



@jasongtau @thiagospassos

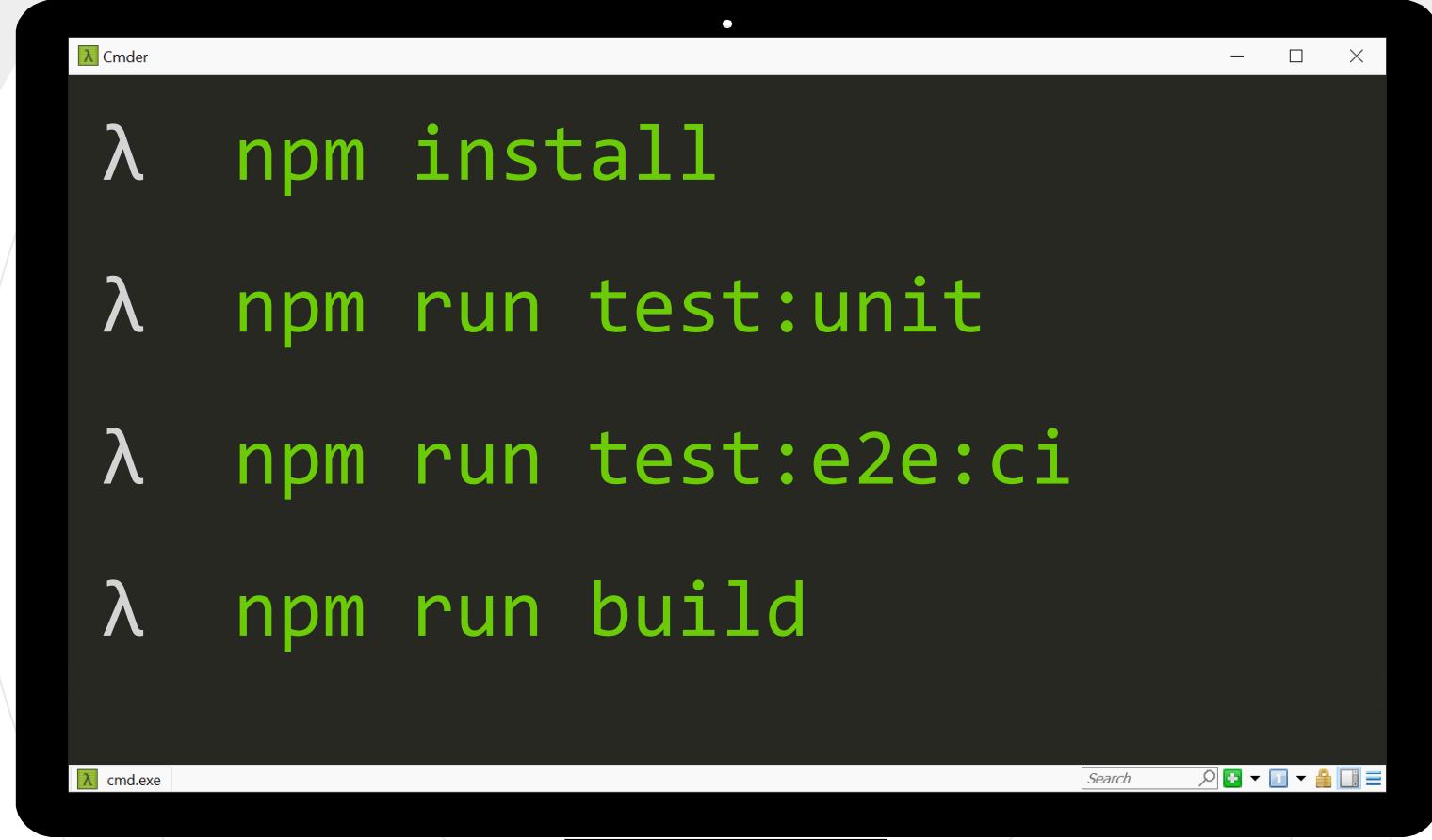
# Agenda

Environment variables

Tooling

Continuous Integration

Continuous Deployment



```
λ npm install
λ npm run test:unit
λ npm run test:e2e:ci
λ npm run build
```



@jasongtau @thiagospassos



207

# Agenda

Environment variables

Tooling

Continuous Integration

Continuous Deployment



# Your Turn!

Follow the instructions to complete  
the code exercises and challenges

 Workshop Guide  
[bit.ly/masteringvuejs](https://bit.ly/masteringvuejs)

 Workshop Code  
[bit.ly/vuejs-minnesota-2019](https://bit.ly/vuejs-minnesota-2019)



@jasongtau @thiagospassos

# Thank you!

The resources for this workshop:

-  Workshop Guide  
[bit.ly/masteringvuejs](https://bit.ly/masteringvuejs)

-  Workshop Code  
[bit.ly/vuejs-minnesota-2019](https://bit.ly/vuejs-minnesota-2019)



@jasongtau @thiagospassos