
Welcome to
Mastering Vue.js

Hello!

I am Thiago Passos

passos.com.au

github.com/thiagospassos

@thiagospassos @jasongtau

Hello!

I am Jason Taylor

codingflow.net

github.com/jasongt

@thiagospassos @jasongtau

Our Goals?

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

About You?

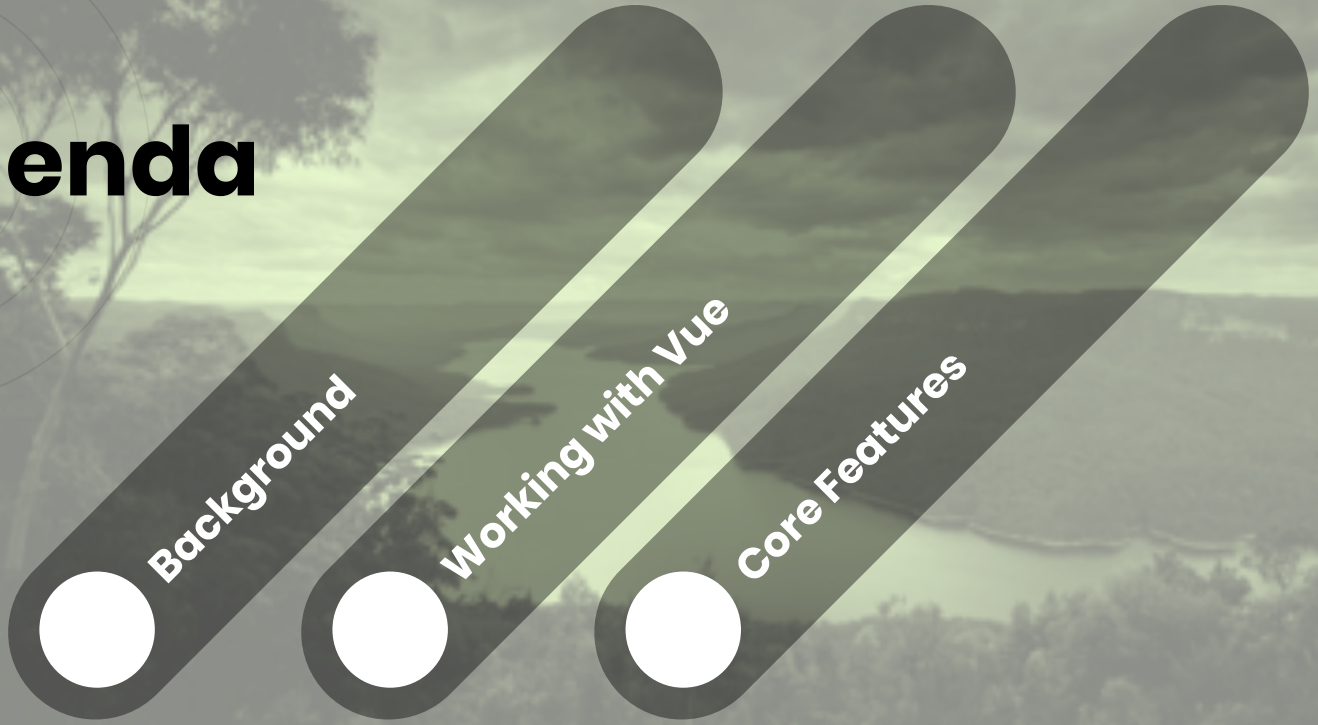
- What's your name?
- What's your background?
- What do you expect from it?
- What's your main focus?

Agenda

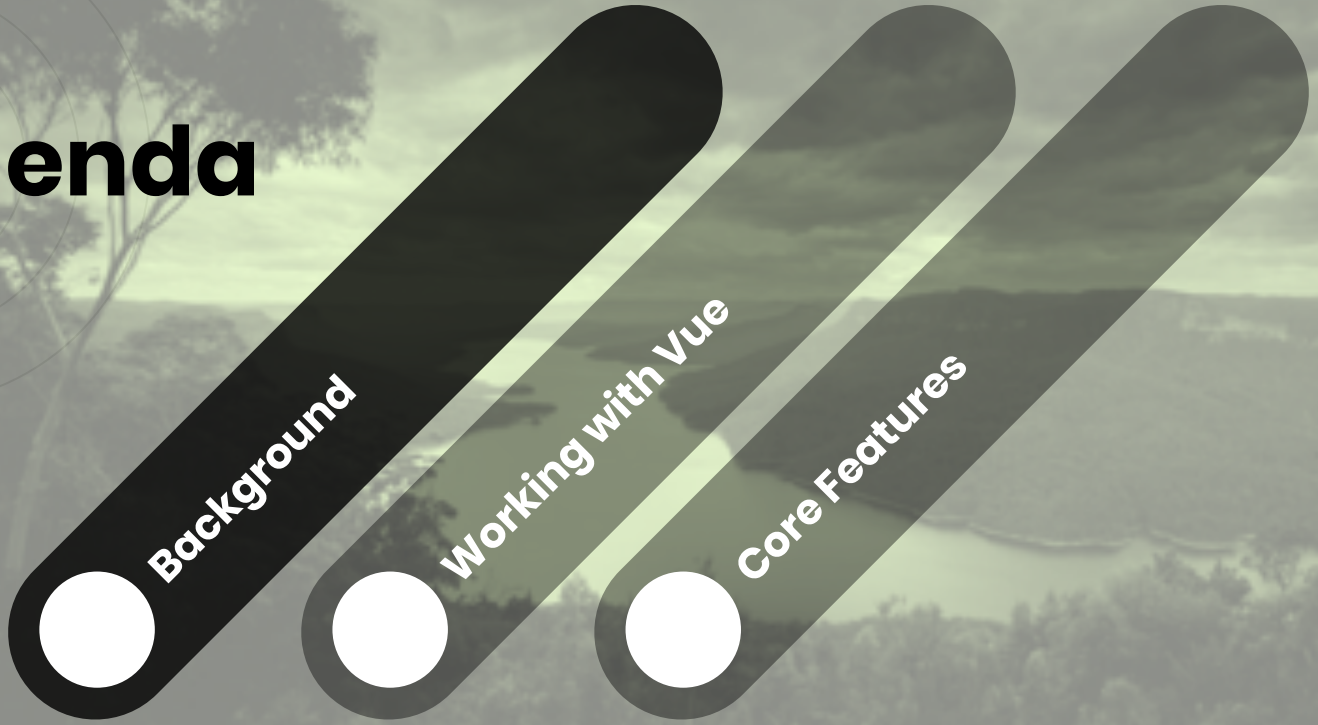
- Getting Started
- Automated Testing
- Beyond the Basics
- Consuming APIs
- Dynamic Forms and Validation
- State Management


Mastering Vue.js
Getting Started

Agenda



Agenda



History

- Dynamic HTML 19
- jQuery 2006
- Knockout.js 2010
- AngularJS 2010
- Backbone.js 2010

Evan You

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

What's Vue?

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

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

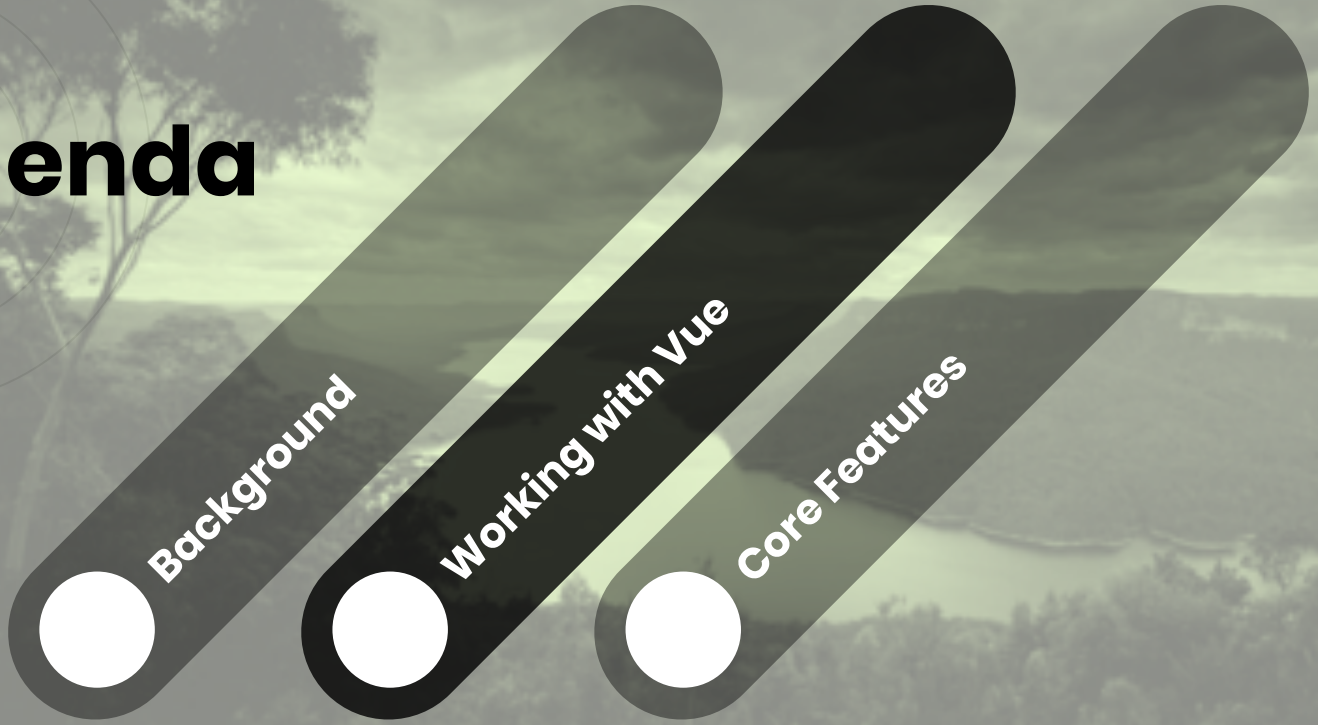
20KB min+gzip Runtime, Blazing Fast Virtual DOM, Minimal Optimization Efforts

@thiagospassos @jasongtau

Features

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

Agenda



Essential Sk

HTML

JS

Recommended Tools

Visual Studio Code

Vue VSCode
Snippets

Vue
Dev Tools

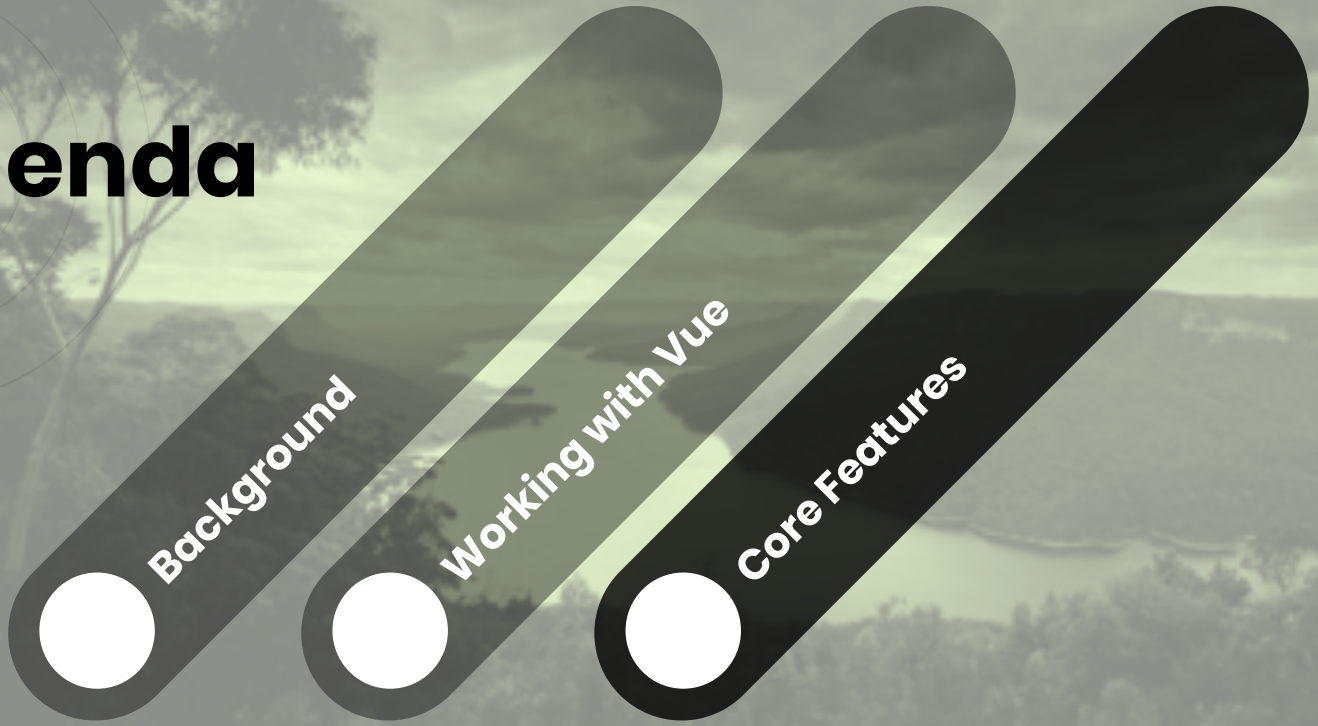
Demo: Hello, World!

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

Demo: Todo App

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

Agenda



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: {
  },
})
```

Template Syntax

Text	{{ message }}
Attributes	v-bind:id="..." or :id="..."
Events	v-on:click="..." or @click="..."
Modifiers	@submit.prevent="..."

Computed Properties

Reactive and Cached

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

Normal binding

```
{{ itemCount }}
```

Class and Style Bindings

Class	:class="{ done: t.done }"
Multiple Classes	:class="{ done: t.done, ... }"
Styles	:style="{ color: todoColor }"
Style object*	{ todo: todoStyles }

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>
```

List Rendering

v-for

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

Event Handling

v-on or @

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

Form Binding

v-model

<input *v-model*="todo“ />

Your Turn!

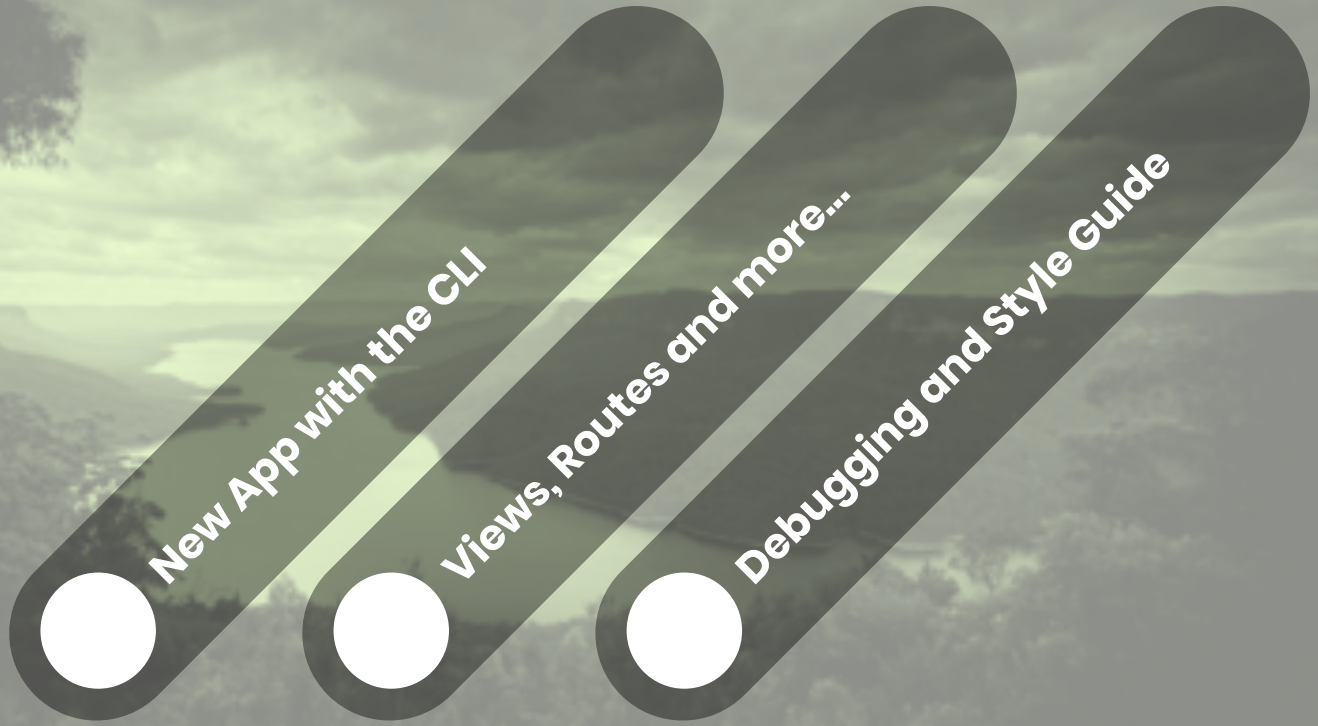
Follow the instructions to complete the code exercises and challenges

Workshop Guide
bit.ly/masteringvuejs

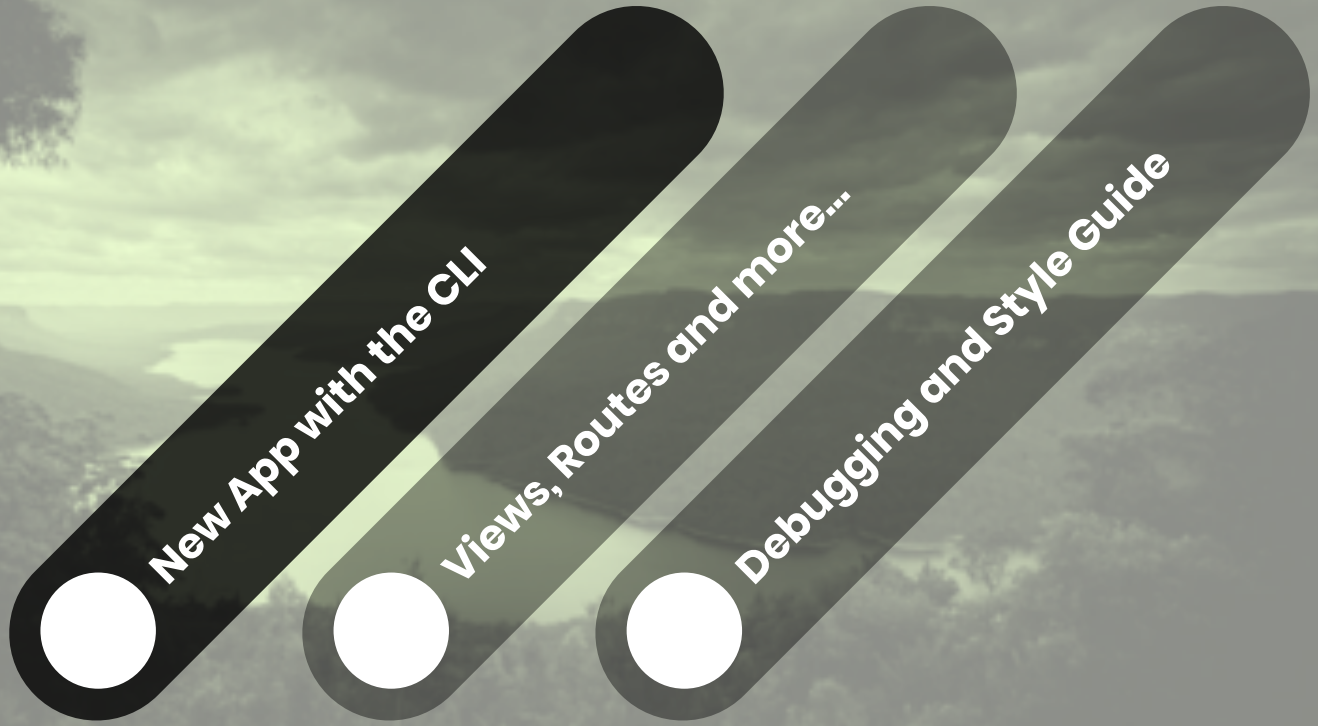
Workshop Code
bit.ly/vuejs-sydney-2019


Mastering Vue.js
Beyond the Basics

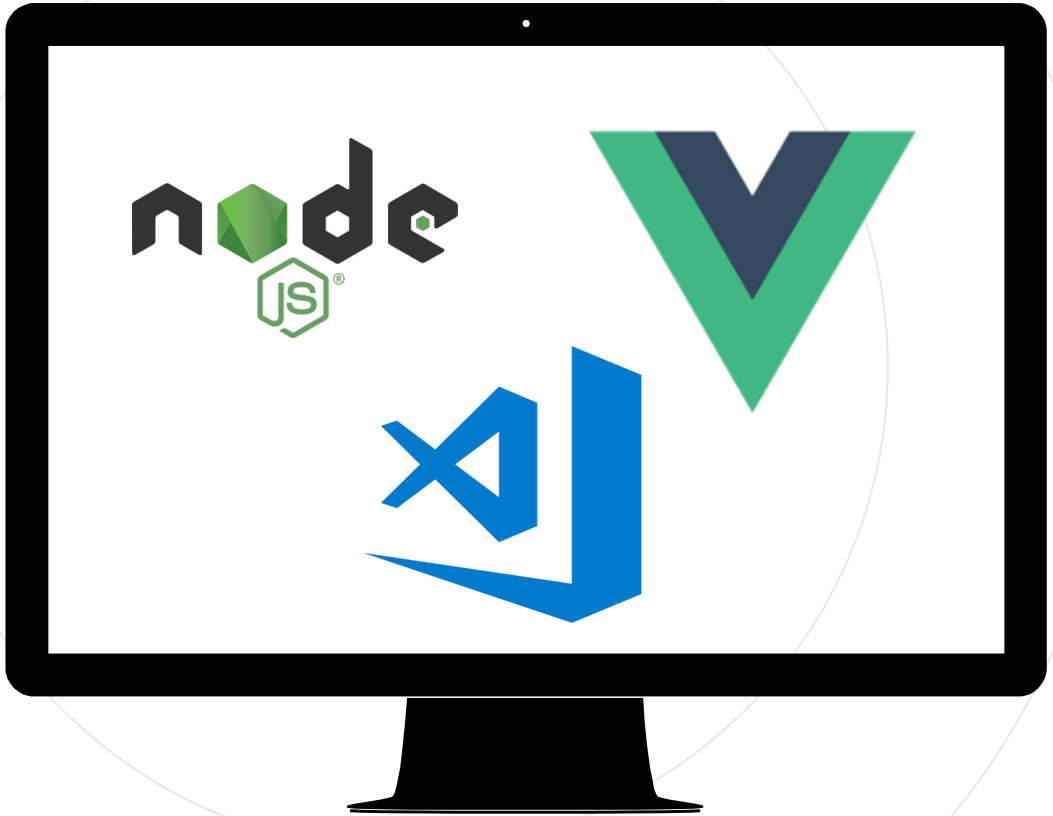
Agenda


New App with the CLI
Views, Routes and more...
Debugging and Style Guide

Agenda


New App with the CLI
Views, Routes and more...
Debugging and Style Guide

What do you need?



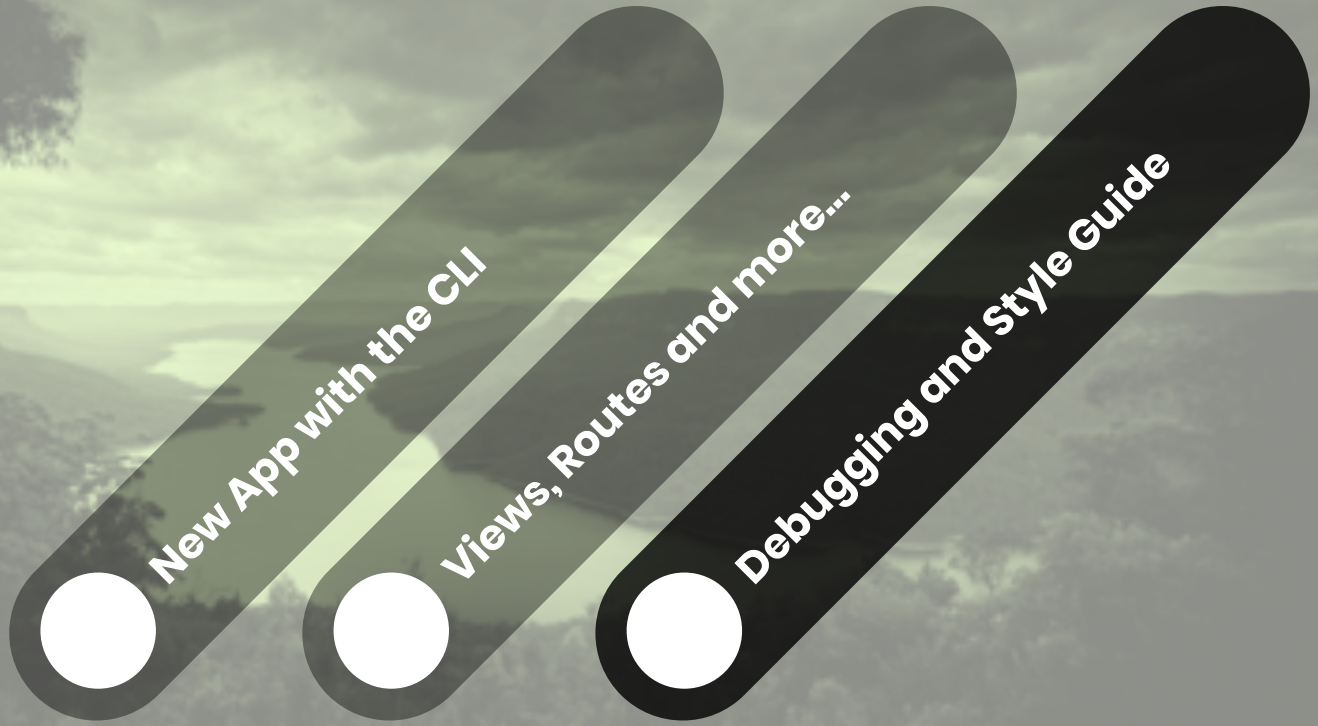
What do we suggest?


Cmder
λ npm i -g @vue/cli
λ vue create app-name
λ vue ui
cmd.exe

Agenda

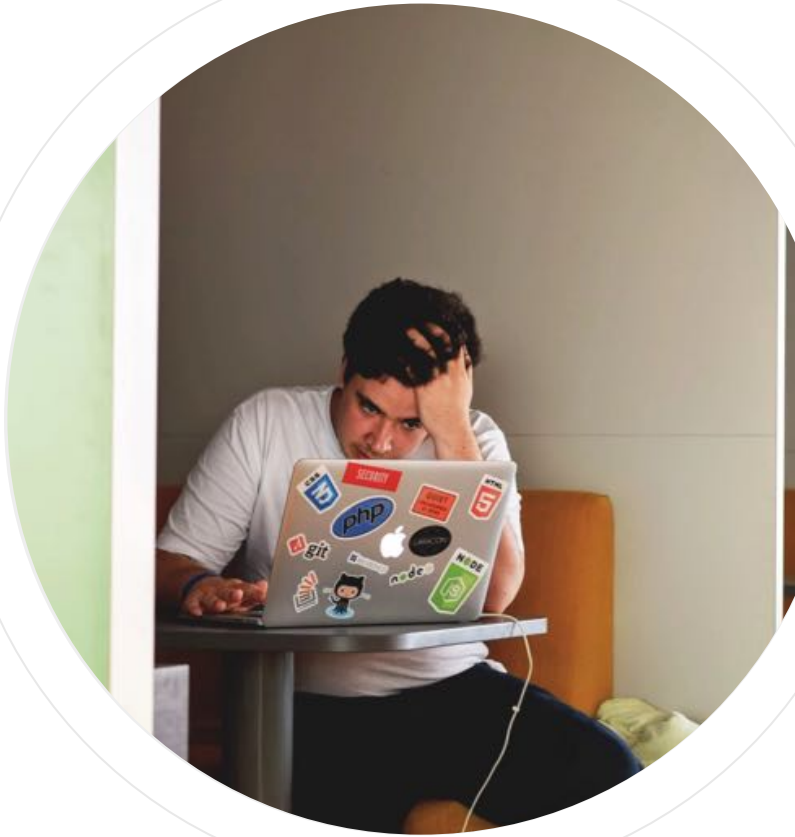
- New App with the CLI
- Views, Routes and more...
- Debugging and Style Guide

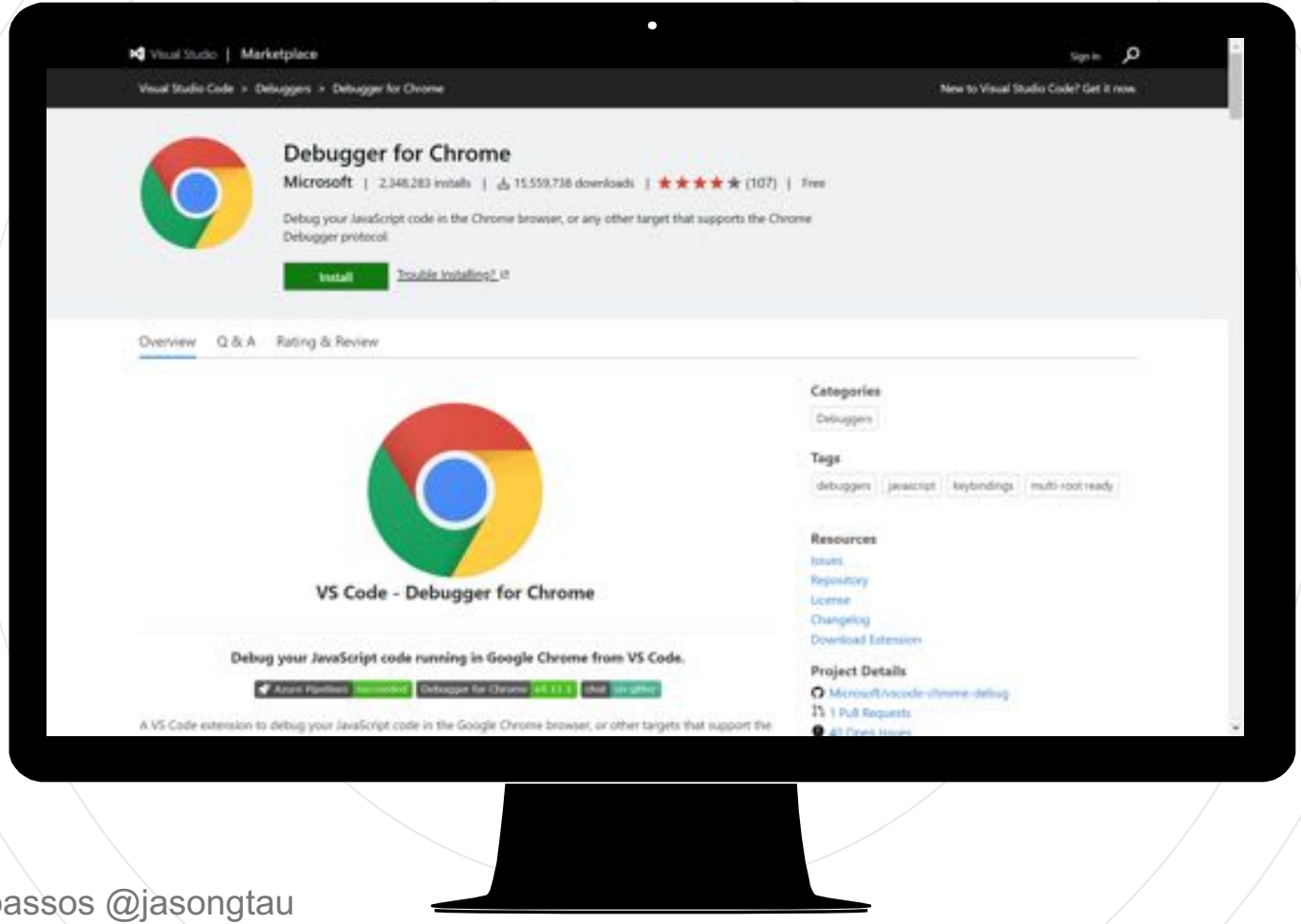
Agenda


New App with the CLI
Views, Routes and more...
Debugging and Style Guide

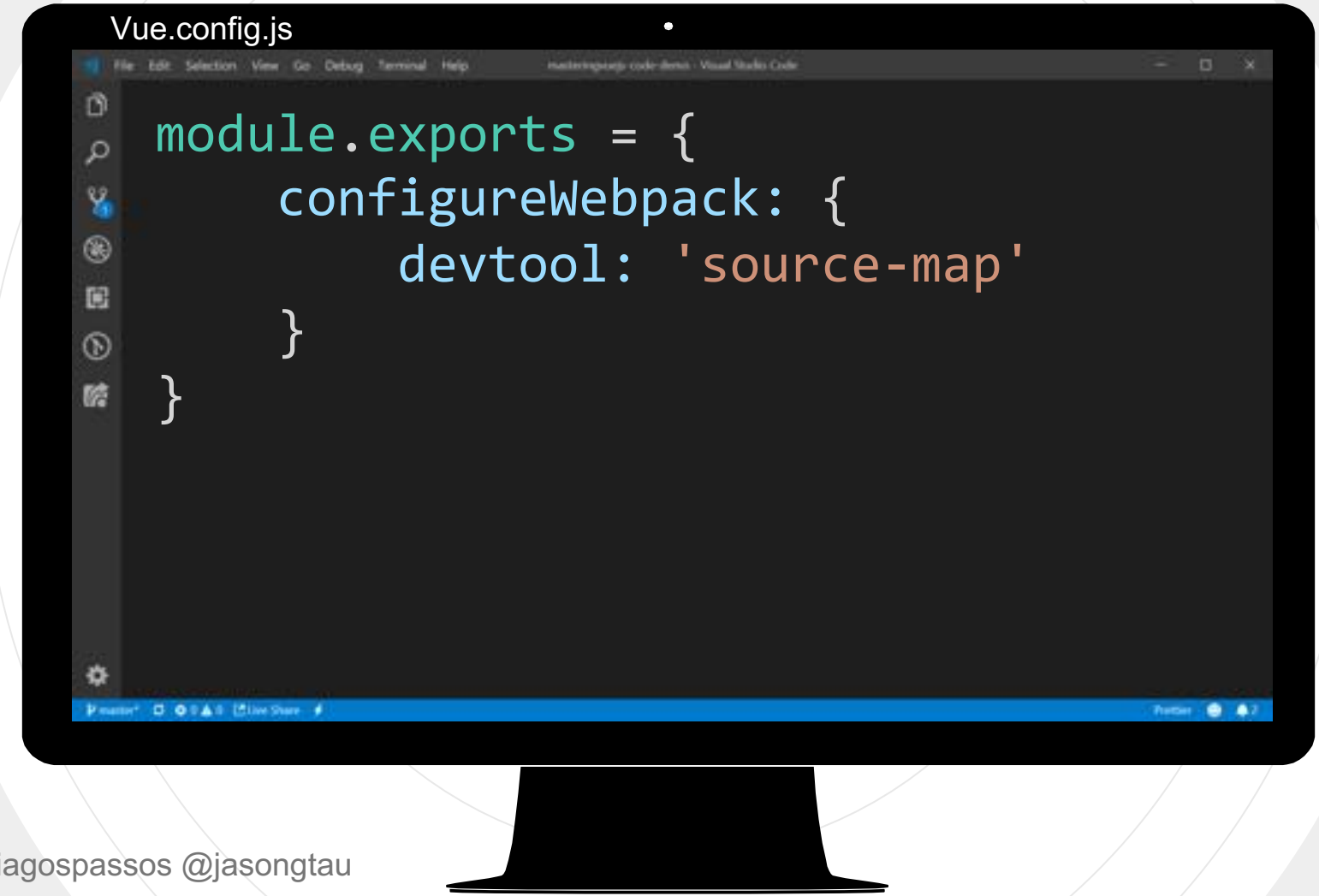
Debugging

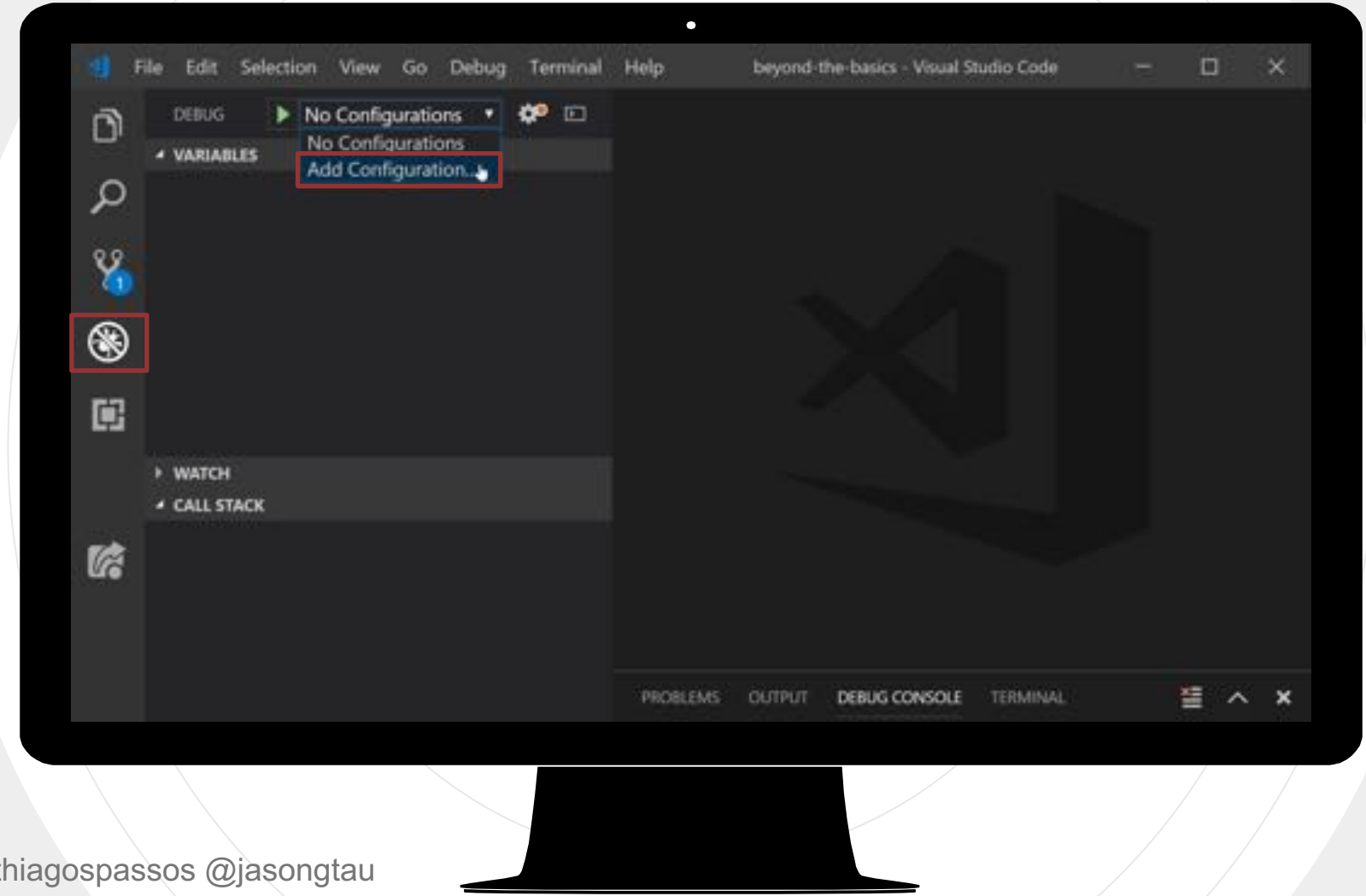
bit.ly/vue-debugging

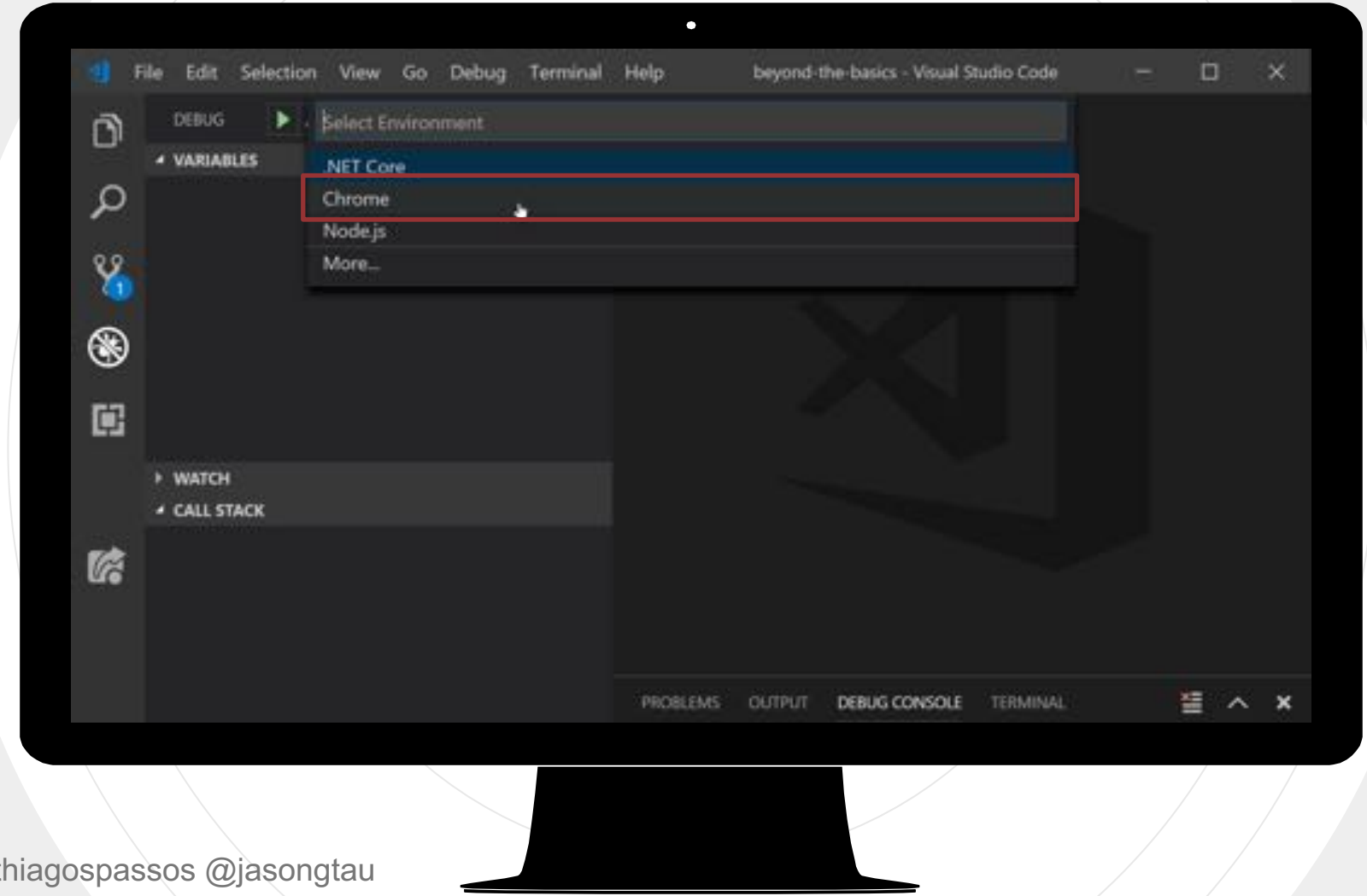


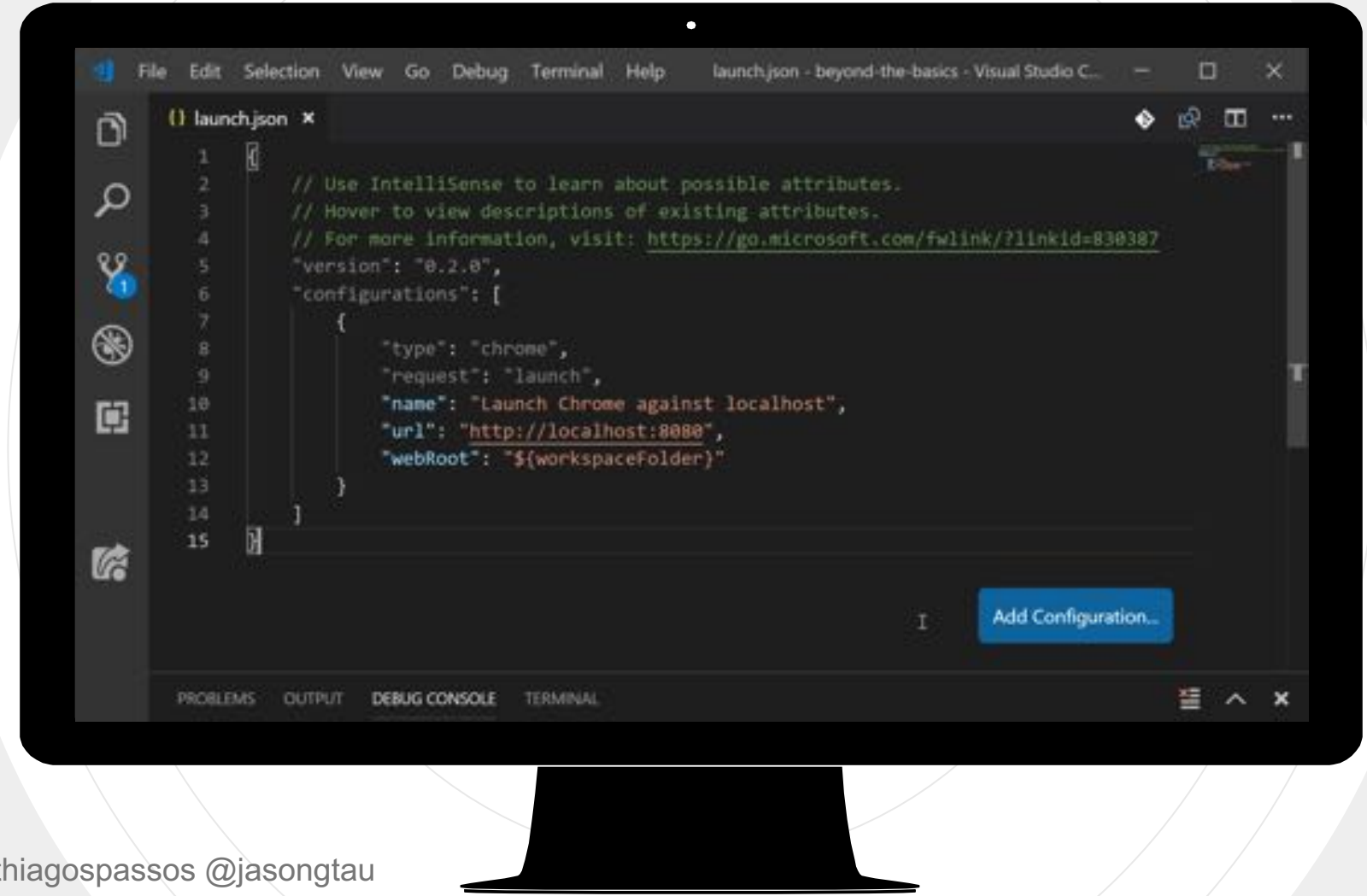

Visual Studio | Marketplace
Sign in
Visual Studio Code > Debuggers > Debugger for Chrome
New to Visual Studio Code? Get it now.
Debugger for Chrome
Microsoft | 2,348,283 installs | 15,559,738 downloads | ★★★★★ (107) | Free
Debug your JavaScript code in the Chrome browser, or any other target that supports the Chrome Debugger protocol
Install
Trouble Installing?
Overview
Q & A
Rating & Review
VS Code - Debugger for Chrome
Debug your JavaScript code running in Google Chrome from VS Code.
Categories
Debuggers
Tags
debuggers
javascript
keybindings
multi-root ready
Resources
Issues
Repository
License
Changelog
Download Extension
Project Details
Microsoft/vscode-chrome-debug
1 Pull Requests

Vue.config.js

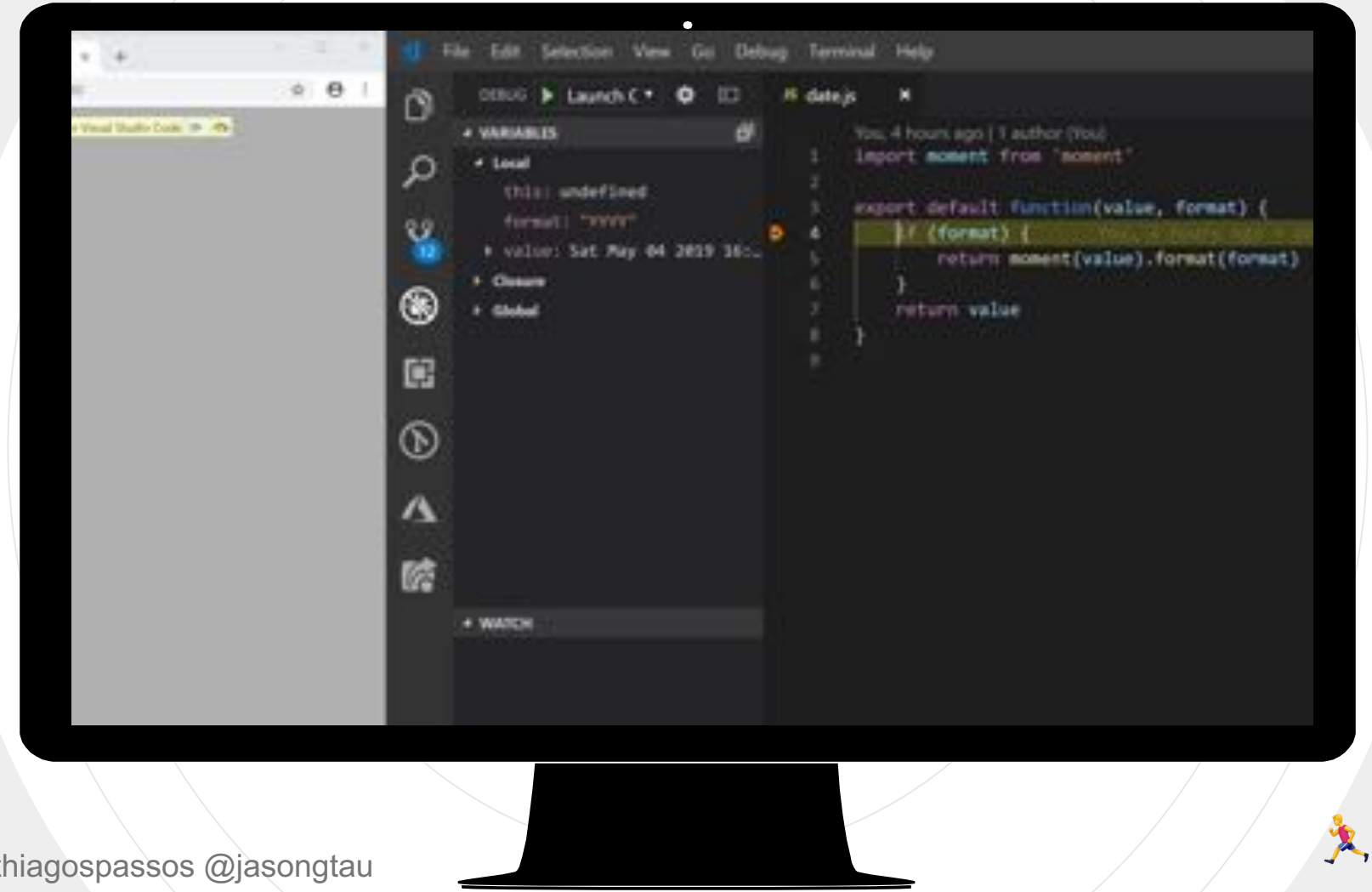



File Edit Selection View Go Debug Terminal Help
beyond-the-basics - Visual Studio Code
DEBUG
No Configurations
No Configurations
Add Configuration...
VARIABLES
WATCH
CALL STACK
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL


File Edit Selection View Go Debug Terminal Help
beyond-the-basics - Visual Studio Code
DEBUG
Select Environment
VARIABLES
.NET Core
Chrome
Node.js
More...
WATCH
CALL STACK
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL


File Edit Selection View Go Debug Terminal Help launch.json - beyond-the-basics - Visual Studio C...
launch.json
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 }
Add Configuration...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL


Cmder
λ npm run serve
cmd.exe


File Edit Selection View Go Debug Terminal Help
DEBUG Launch C
VARIABLES
Local
this: undefined
format: "YYYY"
value: Sat May 04 2019 16:...
Closure
Global
WATCH
date.js
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
}

Style Guide

bit.ly/vue-style-guide

@thiagospassos @jasongtau

```
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

Workshop Code
bit.ly/vuejs-sydney-2019


Mastering Vue.js
Dynamic Forms & Validation

Agenda


Best Practices
Inline Forms & Validation
Complex Forms & Validation

Agenda

- Best Practices
- Inline Forms & Validation
- Complex Forms & Validation

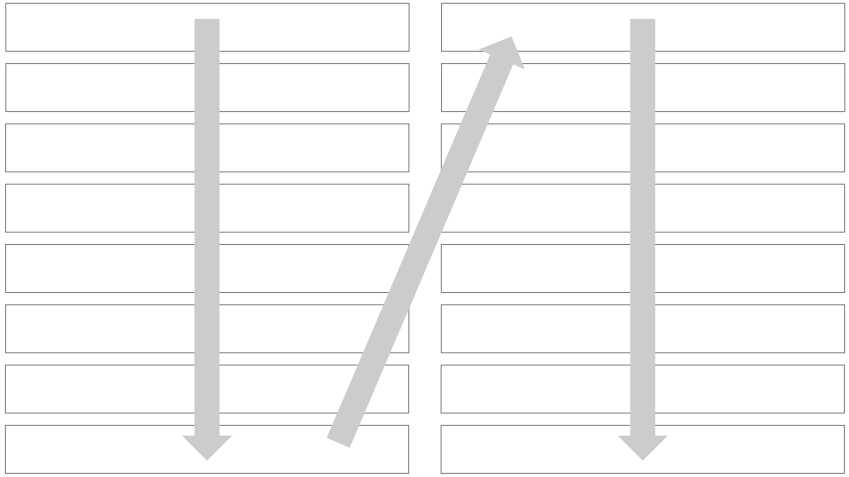
Best Practices

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

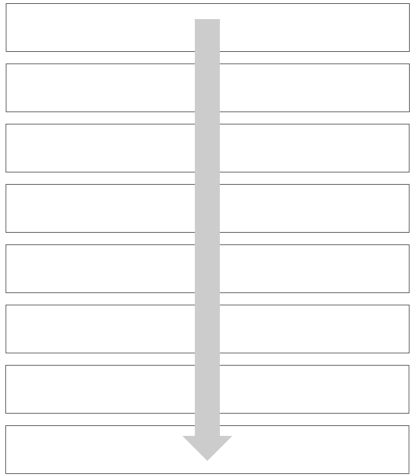
1. Single Column Layout

- 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



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

Bad Example

Credit Card Number

Expiry

CCV

Address

Full Name

Good Example

Full Name

Address

Credit Card Number

Expiry

CCV

3. Group Related Fields

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

Bad Example

Full Name

Address

Credit Card Number

Expiry

CCV

Good Example

Personal

Full Name

Address

Line 1

Payment

Credit Card Number

Expiry

CCV

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

Agenda

- Best Practices
- Inline Forms & Validation
- Complex Forms & Validation

Demo

- Create a form for inline editing of categories
- Add validation support to the new categories form

Agenda


Best Practices
Inline Forms & Validation
Complex Forms & Validation

Demo

- Create a form for editing products
- Implement complex validation rules with Vuelidate

Your Turn!

Follow the instructions to complete the code exercises and challenges

Workshop Guide
bit.ly/masteringvuejs

Workshop Code
bit.ly/vuejs-sydney-2019


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

1

Improves confidence and removes fear of changing code

2

Code Quality

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

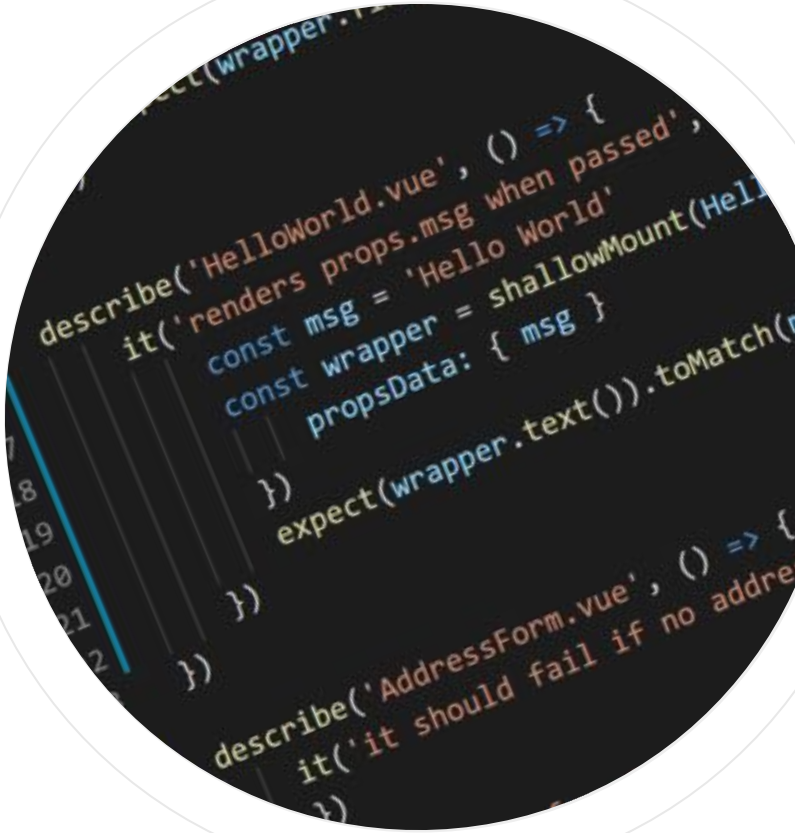
3

Act as documentation

Agenda

- Why automated testing?
- Test types and frameworks
- Testing our components
- Testing our solution

Unit Tests


describe('HelloWorld.vue', () => {
it('renders props.msg when passed',
const msg = 'Hello World'
const wrapper = shallowMount(Hel
propsData: { msg }
})
expect(wrapper.text()).toMatch(
})
})
describe('AddressForm.vue', () => {
it('it should fail if no addre

Integration Tests

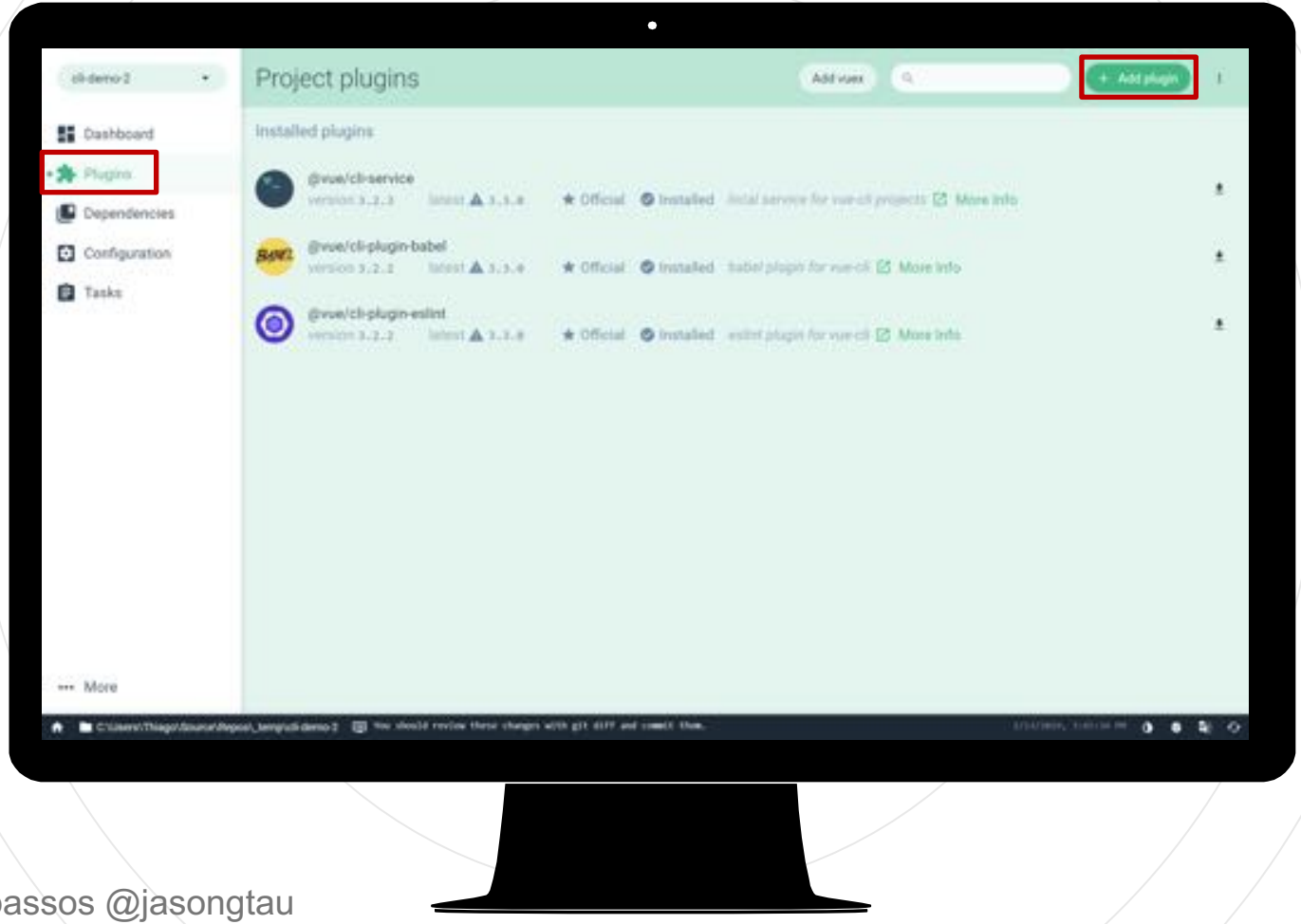
Tools

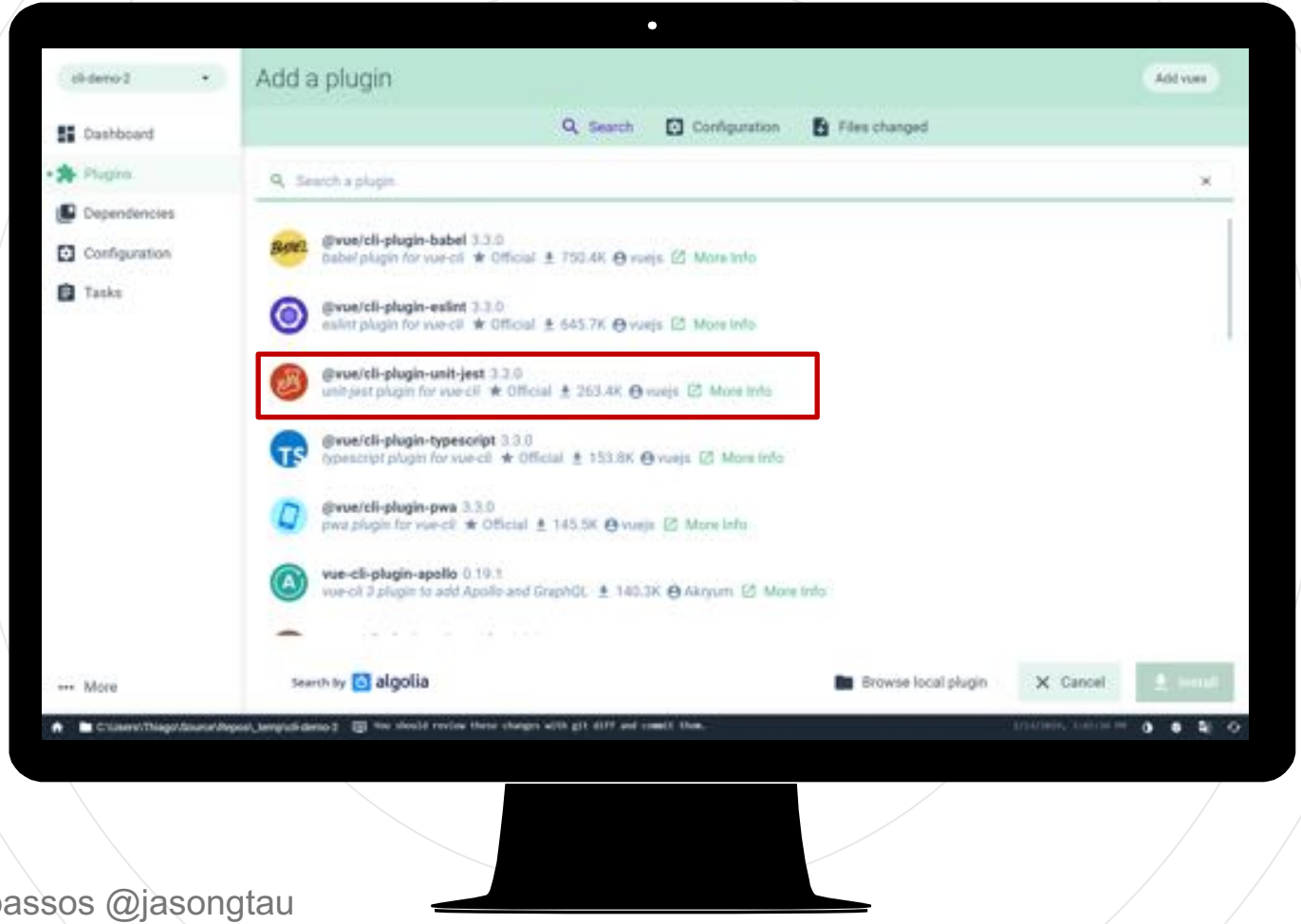
Agenda

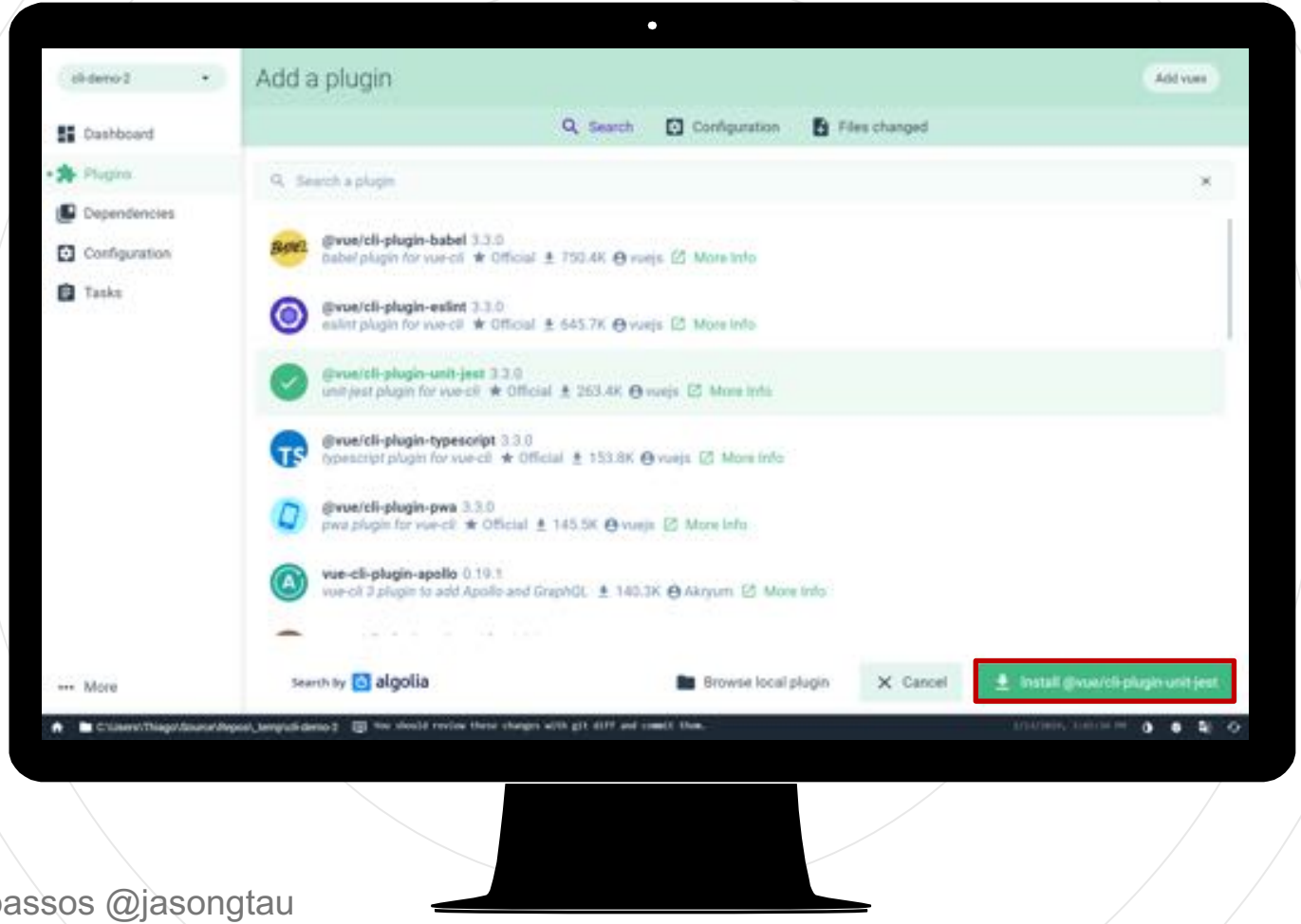
- Why automated testing?
- Test types and frameworks
- Testing our components
- Testing our solution

Unit Tests

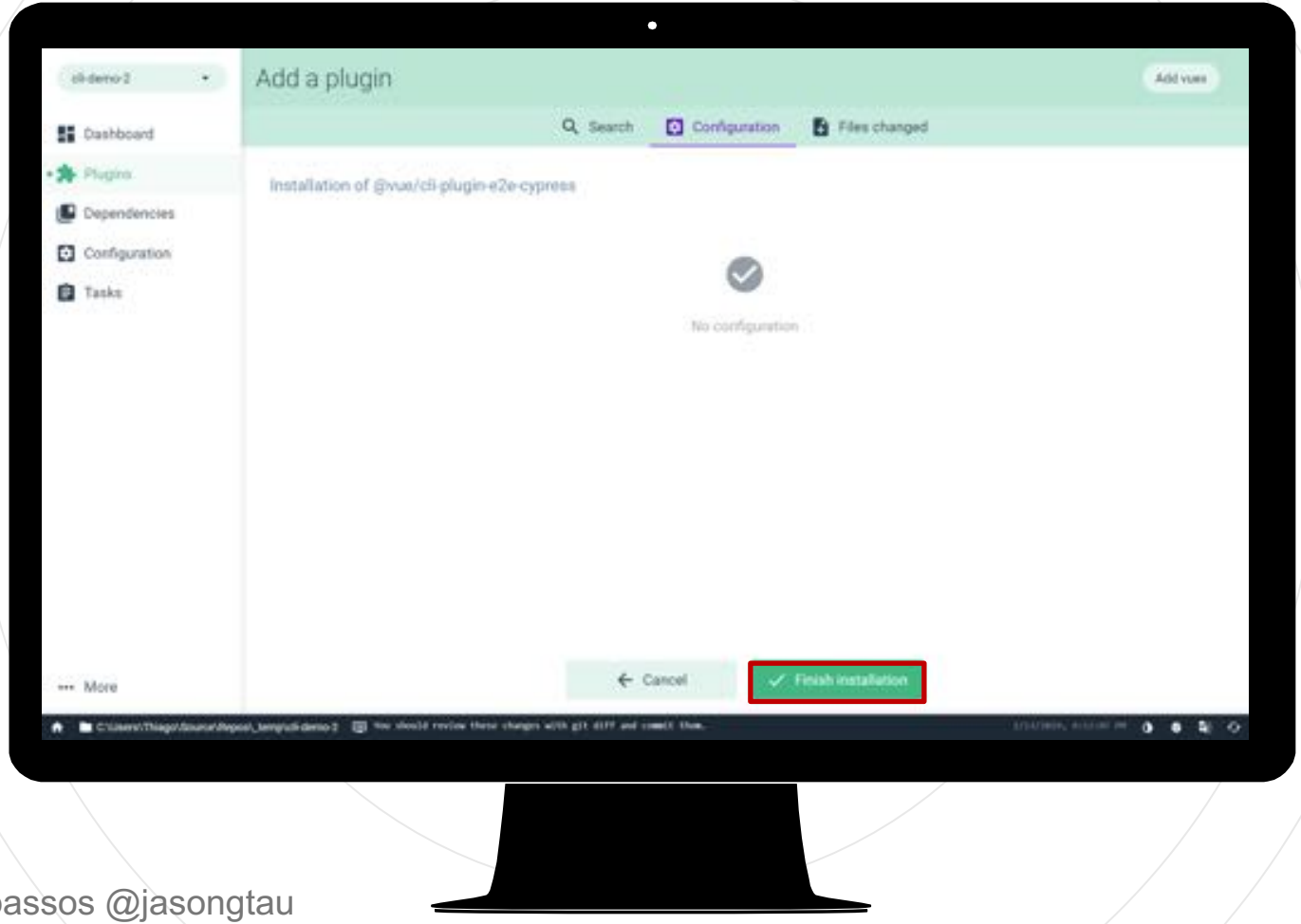
@thiagospassos @jasongtau

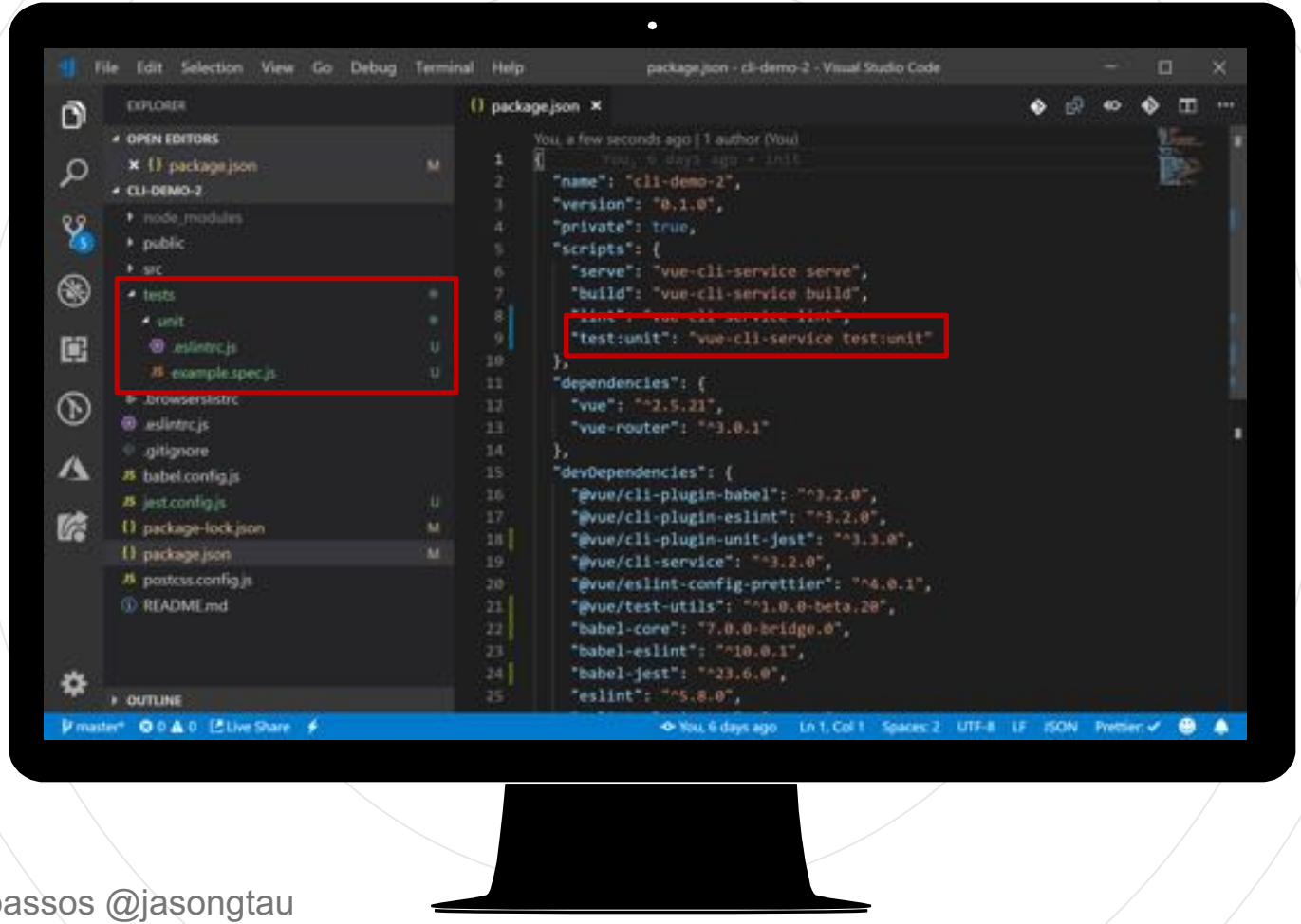

cli-demo-2
Project plugins
Add vuex
+ Add plugin
Dashboard
Plugins
Dependencies
Configuration
Tasks
More
Installed plugins
@vue/cli-service
@vue/cli-plugin-babel
@vue/cli-plugin-eslint
Official
Installed
More info


cli-demo-2
Add a plugin
Add vuex
Dashboard
Plugins
Dependencies
Configuration
Tasks
Search
Configuration
Files changed
Search a plugin
@vue/cli-plugin-babel 3.3.0
babel plugin for vue-cli ★ Official 750.4K vuejs More Info
@vue/cli-plugin-eslint 3.3.0
eslint plugin for vue-cli ★ Official 645.7K vuejs More Info
@vue/cli-plugin-unit-jest 3.3.0
unit-jest plugin for vue-cli ★ Official 263.4K vuejs More Info
@vue/cli-plugin-typescript 3.3.0
typescript plugin for vue-cli ★ Official 153.8K vuejs More Info
@vue/cli-plugin-pwa 3.3.0
pwa plugin for vue-cli ★ Official 145.5K vuejs More Info
vue-cli-plugin-apollo 0.19.1
vue-cli 3 plugin to add Apollo and GraphQL 140.3K Akryum More Info
More
Search by algolia
Browse local plugin
Cancel
Install


cli-demo-2
Add a plugin
Add vuex
Dashboard
Plugins
Dependencies
Configuration
Tasks
Search
Configuration
Files changed
Search a plugin
@vue/cli-plugin-babel 3.3.0
babel plugin for vue-cli ★ Official 750.4K vuejs More Info
@vue/cli-plugin-eslint 3.3.0
eslint plugin for vue-cli ★ Official 645.7K vuejs More Info
@vue/cli-plugin-unit-jest 3.3.0
unit jest plugin for vue-cli ★ Official 263.4K vuejs More Info
@vue/cli-plugin-typescript 3.3.0
typescript plugin for vue-cli ★ Official 153.8K vuejs More Info
@vue/cli-plugin-pwa 3.3.0
pwa plugin for vue-cli ★ Official 145.5K vuejs More Info
vue-cli-plugin-apollo 0.19.1
vue-cli 3 plugin to add Apollo and GraphQL 140.3K Akryum More Info
More
Search by algolia
Browse local plugin
Cancel
Install @vue/cli-plugin-unit-jest


Installing @vue/cli-plugin-unit-jest...

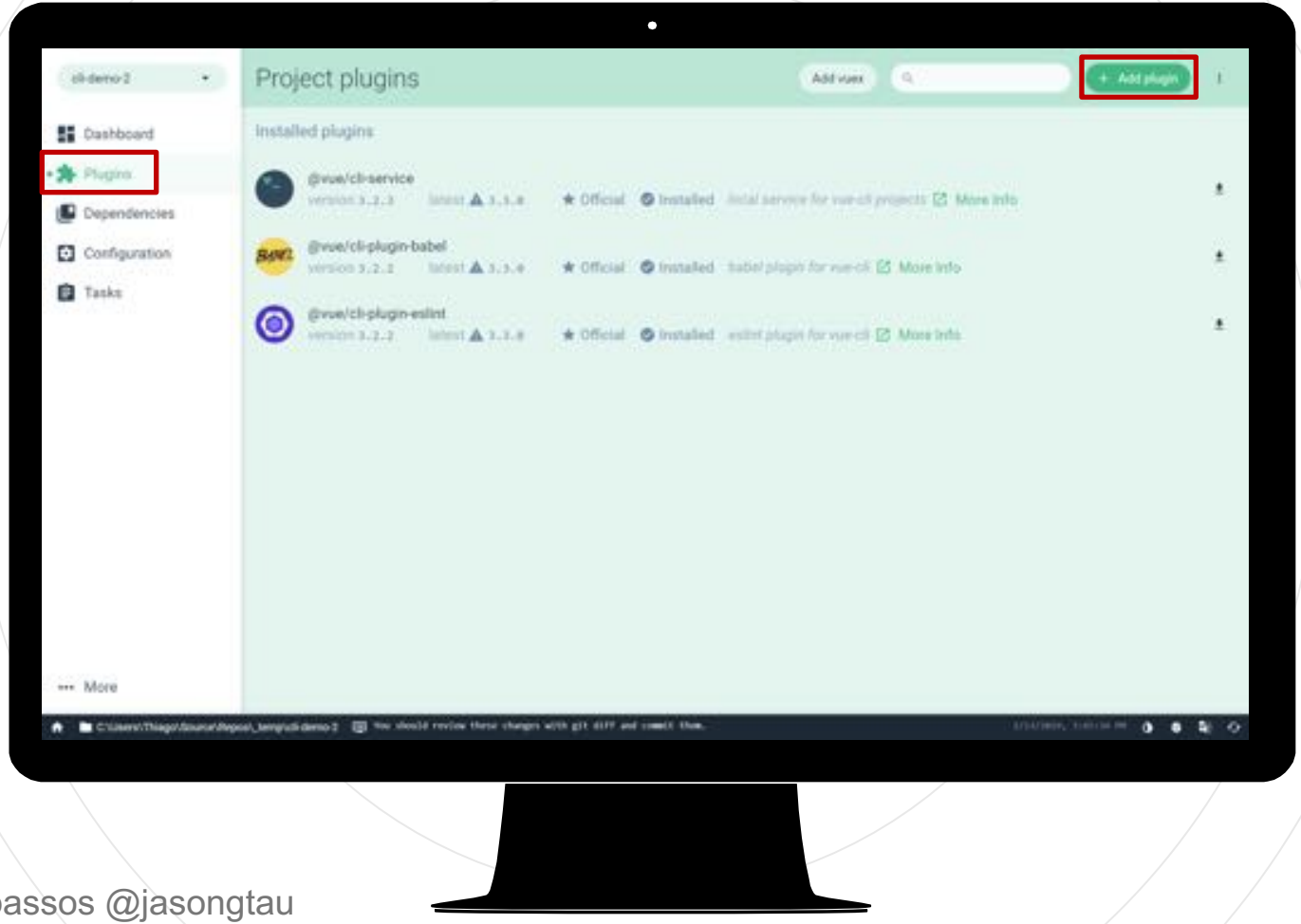

cli-demo-2
Add a plugin
Add vuex
Dashboard
Plugins
Dependencies
Configuration
Tasks
Search
Configuration
Files changed
Installation of @vue/cli-plugin-e2e-cypress
No configuration
More
Cancel
Finish installation

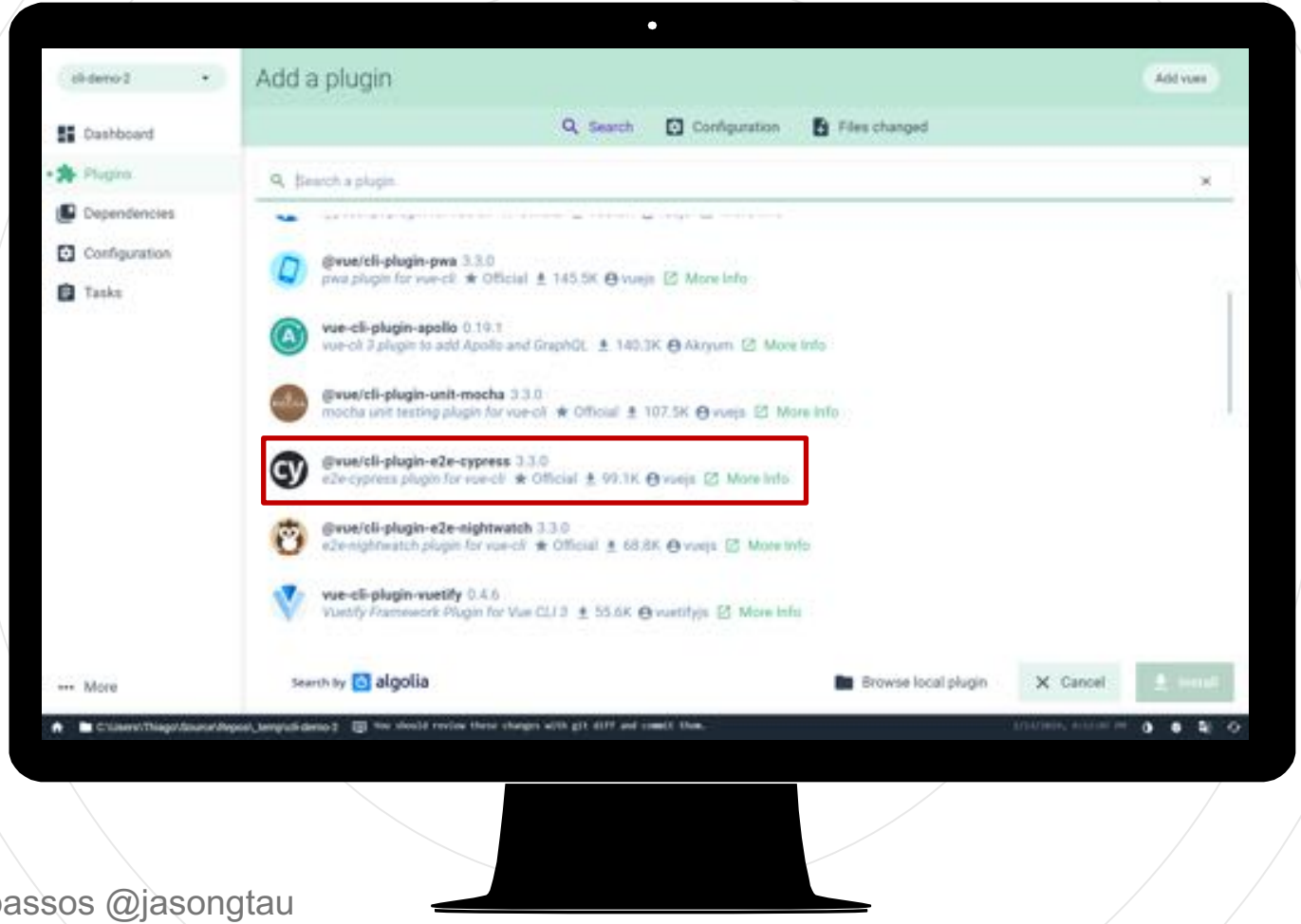

File Edit Selection View Go Debug Terminal Help
package.json - cli-demo-2 - Visual Studio Code
EXPLORER
OPEN EDITORS
package.json
CLI-DEMO-2
node_modules
public
src
tests
unit
.eslintrc.js
example.spec.js
.browserslistrc
.eslintrc.js
.gitignore
babel.config.js
jest.config.js
package-lock.json
package.json
postcss.config.js
README.md
OUTLINE
"name": "cli-demo-2",
"version": "0.1.0",
"private": true,
"scripts": {
"serve": "vue-cli-service serve",
"build": "vue-cli-service build",
"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",

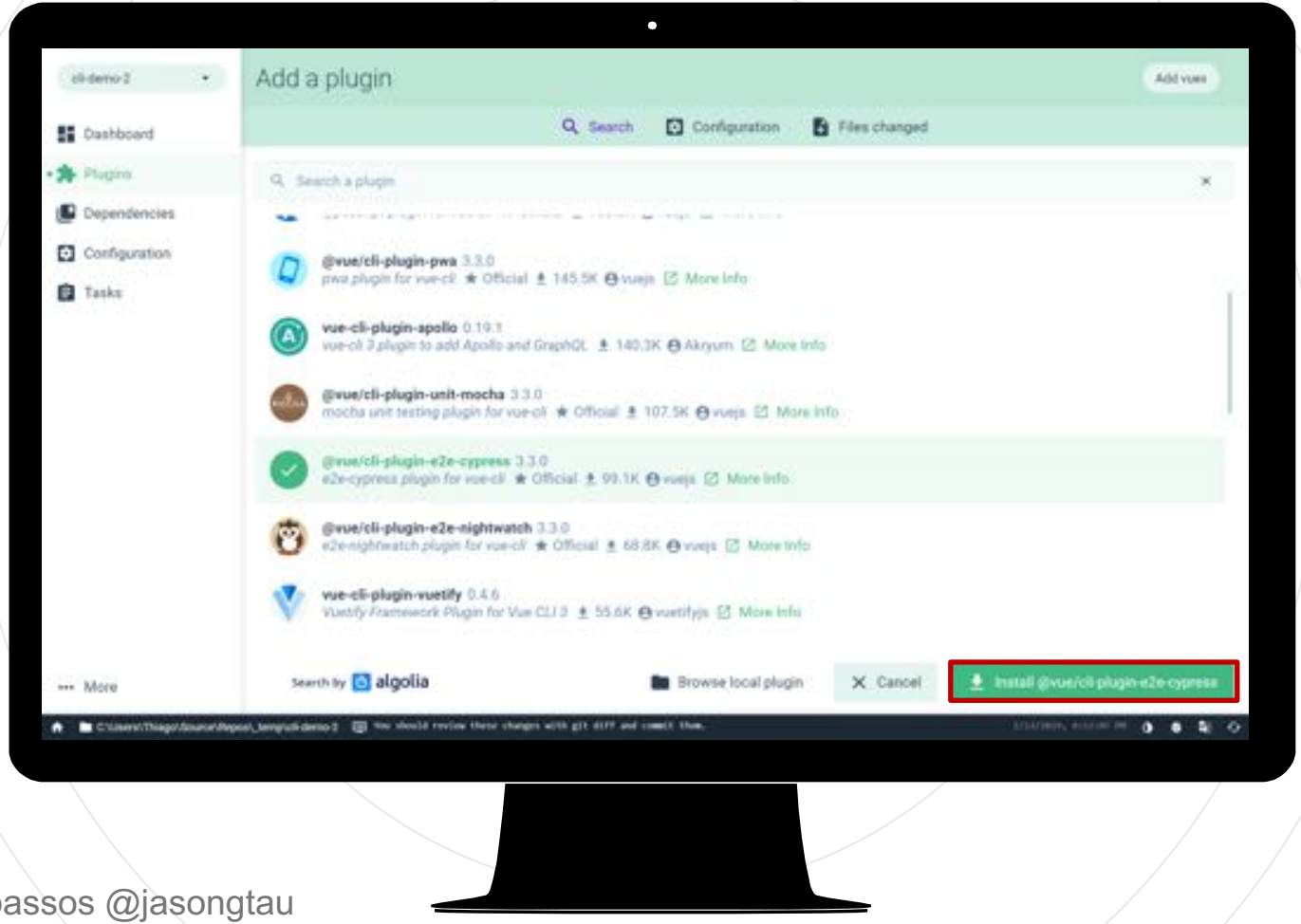
Agenda

- Why automated testing?
- Test types and frameworks
- Testing our components
- Testing our solution

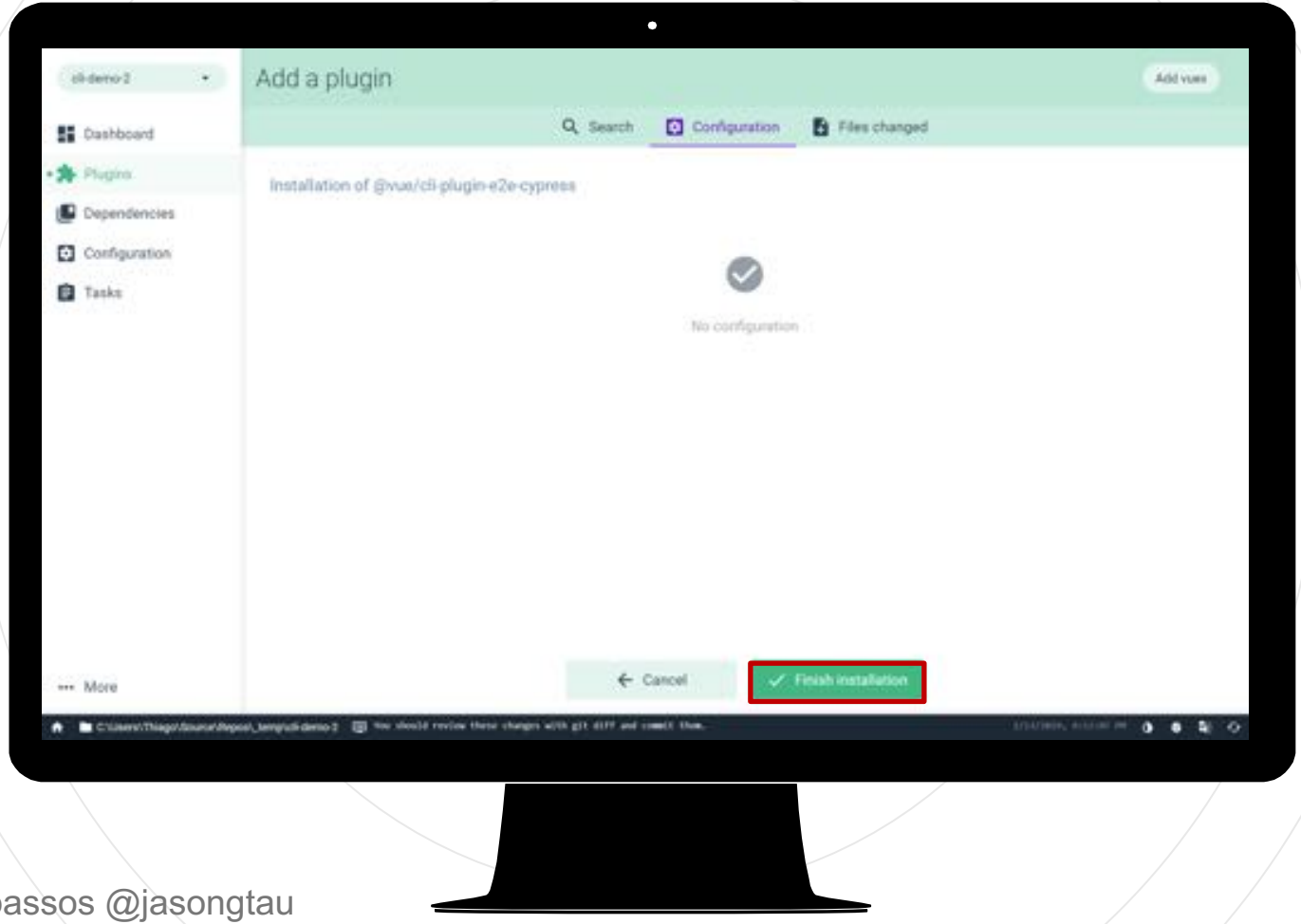
Integration Tests

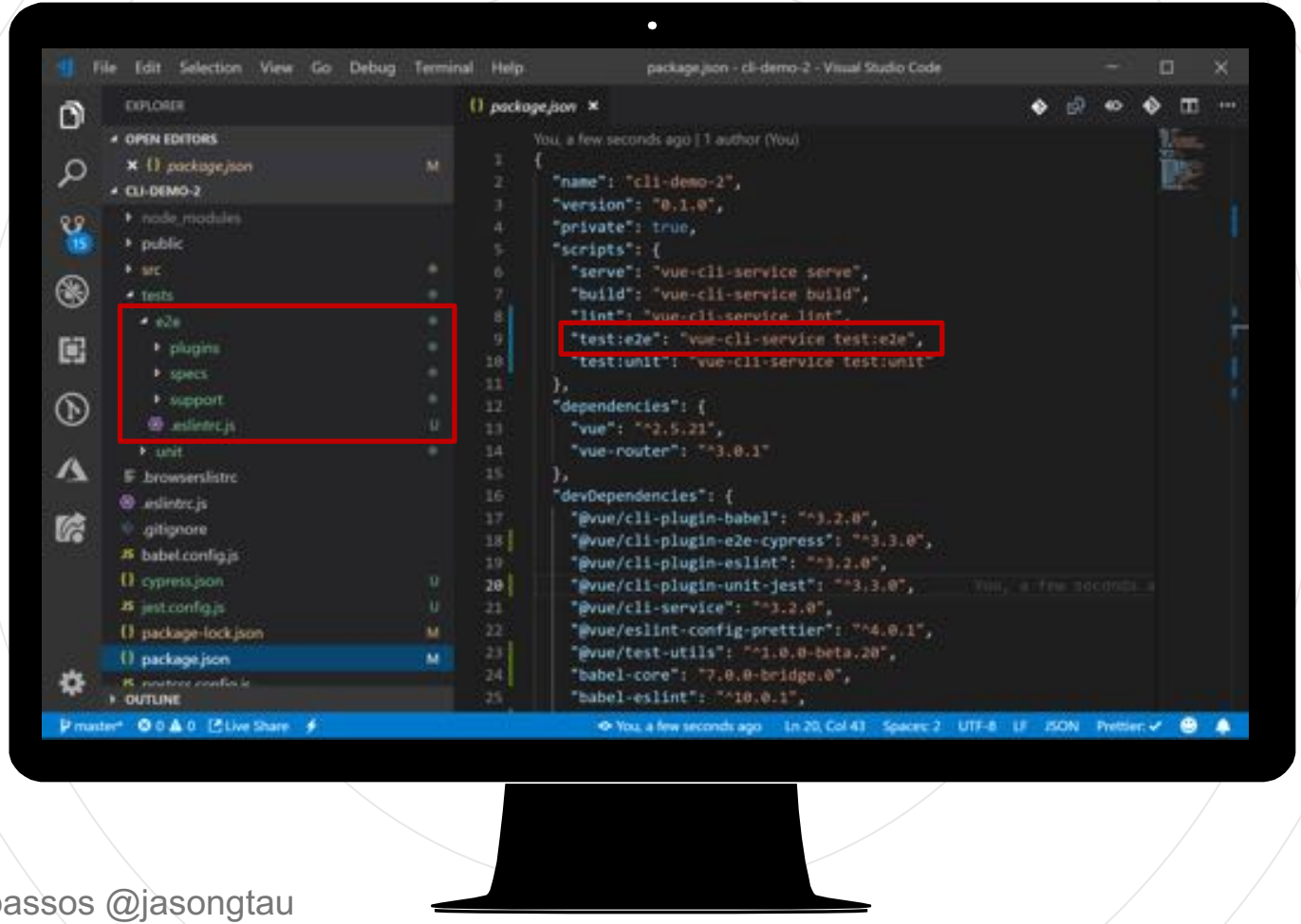

cli-demo-2
Project plugins
Add vuex
+ Add plugin
Dashboard
Plugins
Dependencies
Configuration
Tasks
More
Installed plugins
@vue/cli-service
@vue/cli-plugin-babel
@vue/cli-plugin-eslint
Official
Installed
More info


cli-demo-2
Add a plugin
Add vuex
Dashboard
Plugins
Dependencies
Configuration
Tasks
Search
Configuration
Files changed
Search a plugin
@vue/cli-plugin-pwa 3.3.0
pwa plugin for vue-cli ★ Official 145.5K vuejs More Info
vue-cli-plugin-apollo 0.19.1
vue-cli 3 plugin to add Apollo and GraphQL 140.3K Akryum More Info
@vue/cli-plugin-unit-mocha 3.3.0
mocha unit testing plugin for vue-cli ★ Official 107.5K vuejs More Info
@vue/cli-plugin-e2e-cypress 3.3.0
e2e-cypress plugin for vue-cli ★ Official 99.1K vuejs More Info
@vue/cli-plugin-e2e-nightwatch 3.3.0
e2e-nightwatch plugin for vue-cli ★ Official 68.8K vuejs More Info
vue-cli-plugin-vuetify 0.4.6
Vuetify Framework Plugin for Vue CLI 3 55.6K vuetifyjs More Info
More
Search by algolia
Browse local plugin
Cancel
Install


cli-demo-2
Add a plugin
Add vuex
Dashboard
Plugins
Dependencies
Configuration
Tasks
More
Search
Configuration
Files changed
Search a plugin
@vue/cli-plugin-pwa 3.3.0
pwa plugin for vue-cli ★ Official 145.5K vuejs More Info
vue-cli-plugin-apollo 0.19.1
vue-cli 3 plugin to add Apollo and GraphQL 140.3K Akryum More Info
@vue/cli-plugin-unit-mocha 3.3.0
mocha unit testing plugin for vue-cli ★ Official 107.5K vuejs More Info
@vue/cli-plugin-e2e-cypress 3.3.0
e2e-cypress plugin for vue-cli ★ Official 99.1K vuejs More Info
@vue/cli-plugin-e2e-nightwatch 3.3.0
e2e-nightwatch plugin for vue-cli ★ Official 68.8K vuejs More Info
vue-cli-plugin-vuetify 0.4.6
Vuetify Framework Plugin for Vue CLI 3 55.6K vuetifyjs More Info
Search by algolia
Browse local plugin
Cancel
Install @vue/cli-plugin-e2e-cypress


Installing @vue/cli-plugin-e2e-cypress...


cli-demo-2
Add a plugin
Add vuex
Dashboard
Plugins
Dependencies
Configuration
Tasks
More
Search
Configuration
Files changed
Installation of @vue/cli-plugin-e2e-cypress
No configuration
Cancel
Finish installation


File Edit Selection View Go Debug Terminal Help
package.json - cli-demo-2 - Visual Studio Code
EXPLORER
OPEN EDITORS
package.json
CLI-DEMO-2
node_modules
public
src
tests
e2e
plugins
specs
support
.eslintrc.js
unit
.browserslistrc
.eslintrc.js
.gitignore
babel.config.js
cypress.json
jest.config.js
package-lock.json
package.json
OUTLINE
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",
master* 0 0 Live Share
You, a few seconds ago
Ln 20, Col 43
Spaces: 2
UTF-8
LF
JSON
Prettier: ✓

Your Turn!

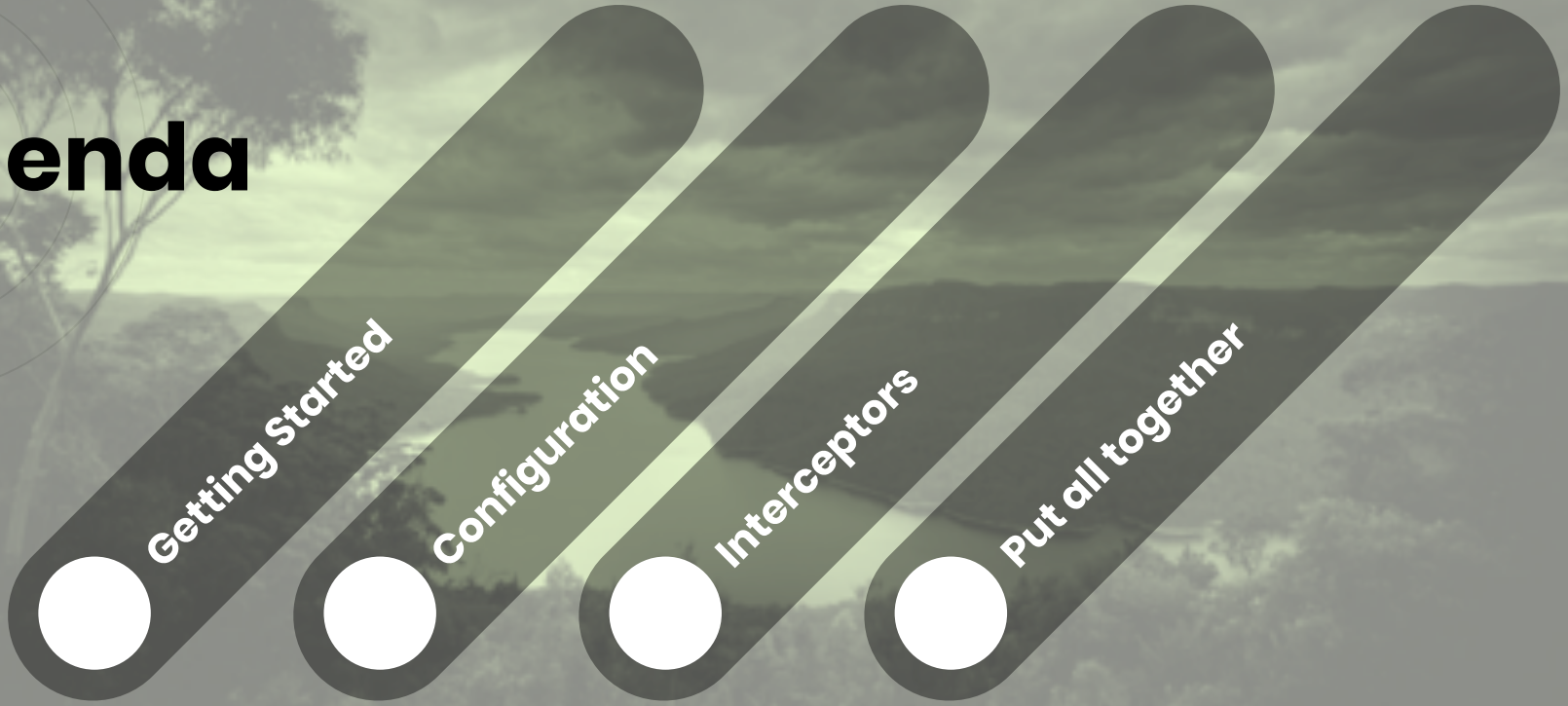
Follow the instructions to complete the code exercises and challenges

Workshop Guide
bit.ly/masteringvuejs

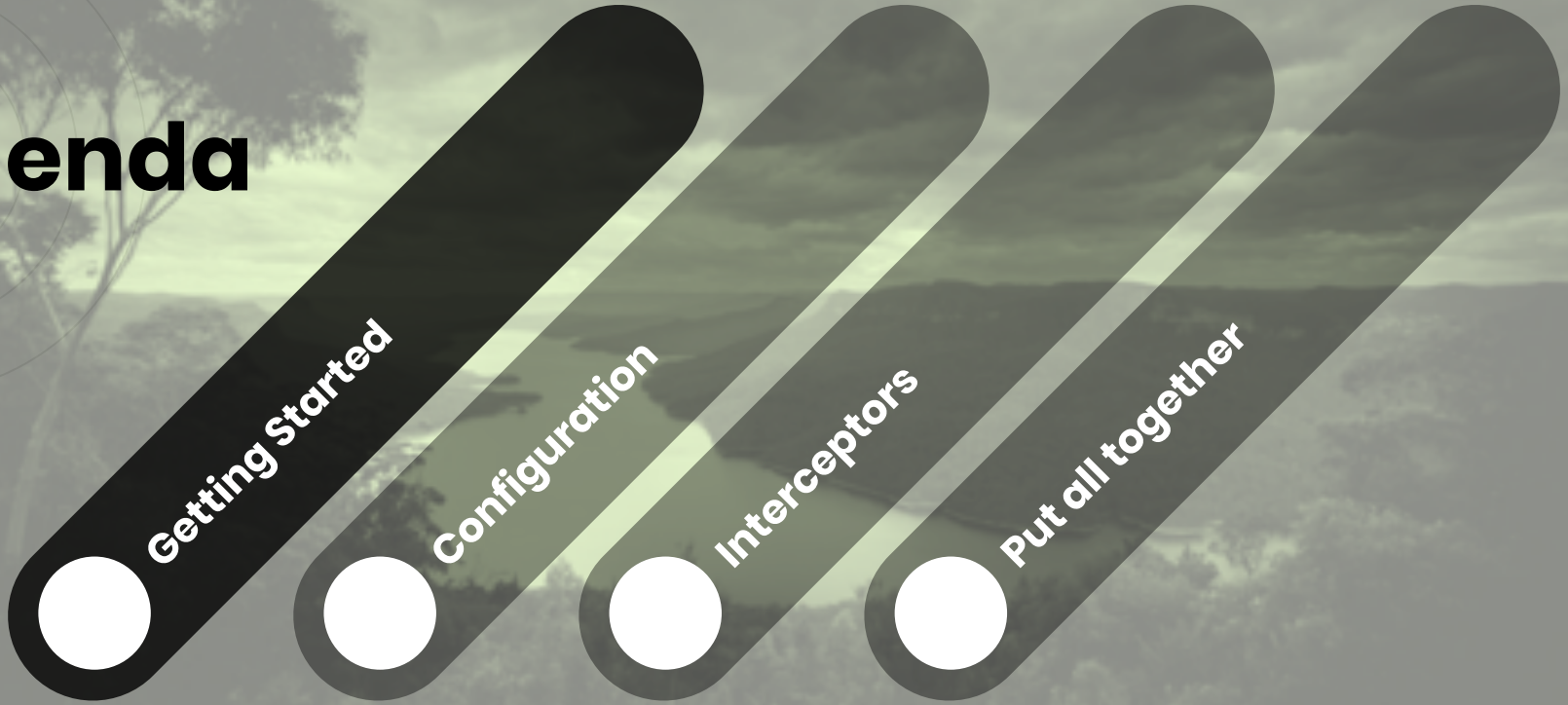
Workshop Code
bit.ly/vuejs-sydney-2019


Vue.js
Consuming APIs

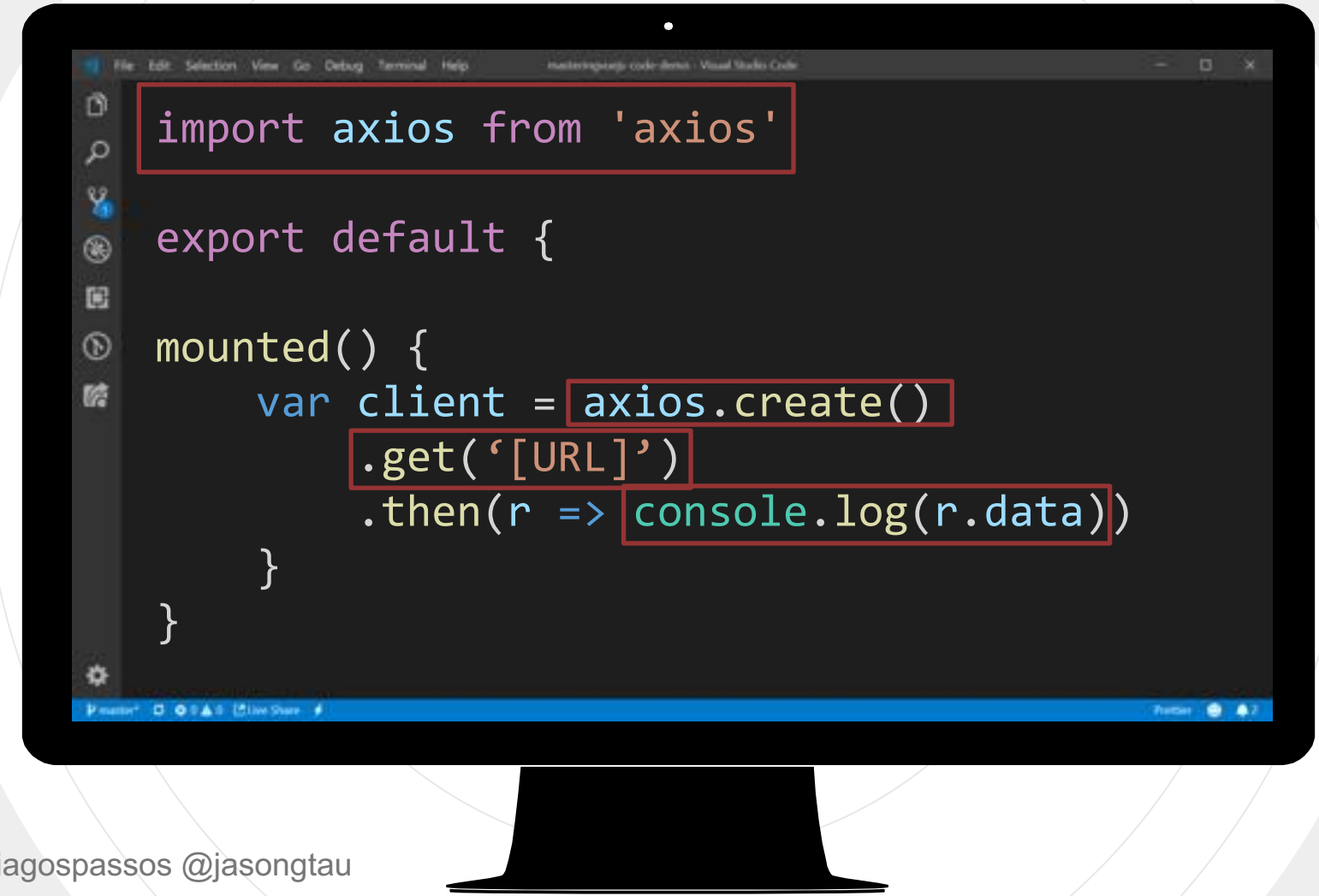
Agenda

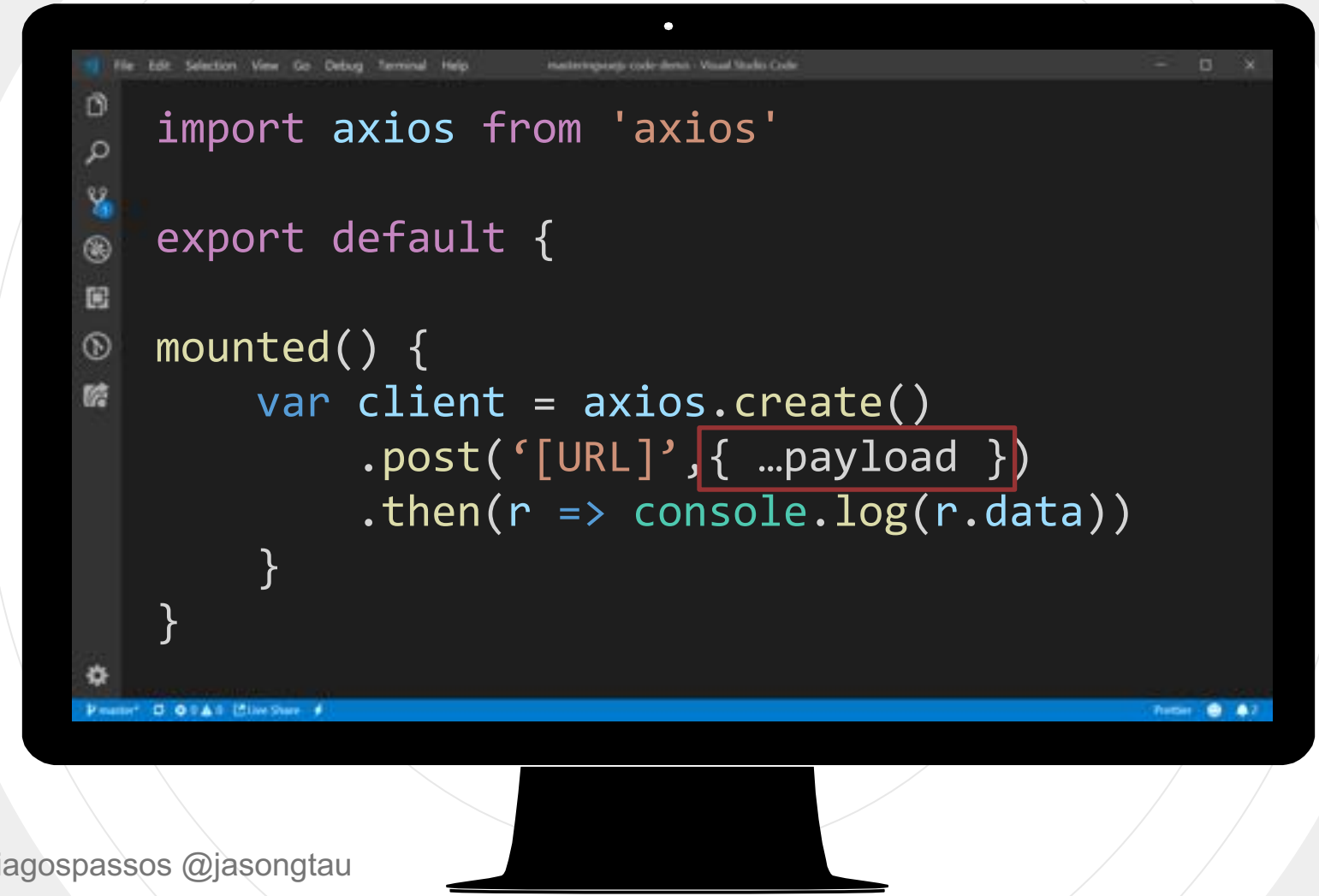

Getting Started
Configuration
Interceptors
Put all together

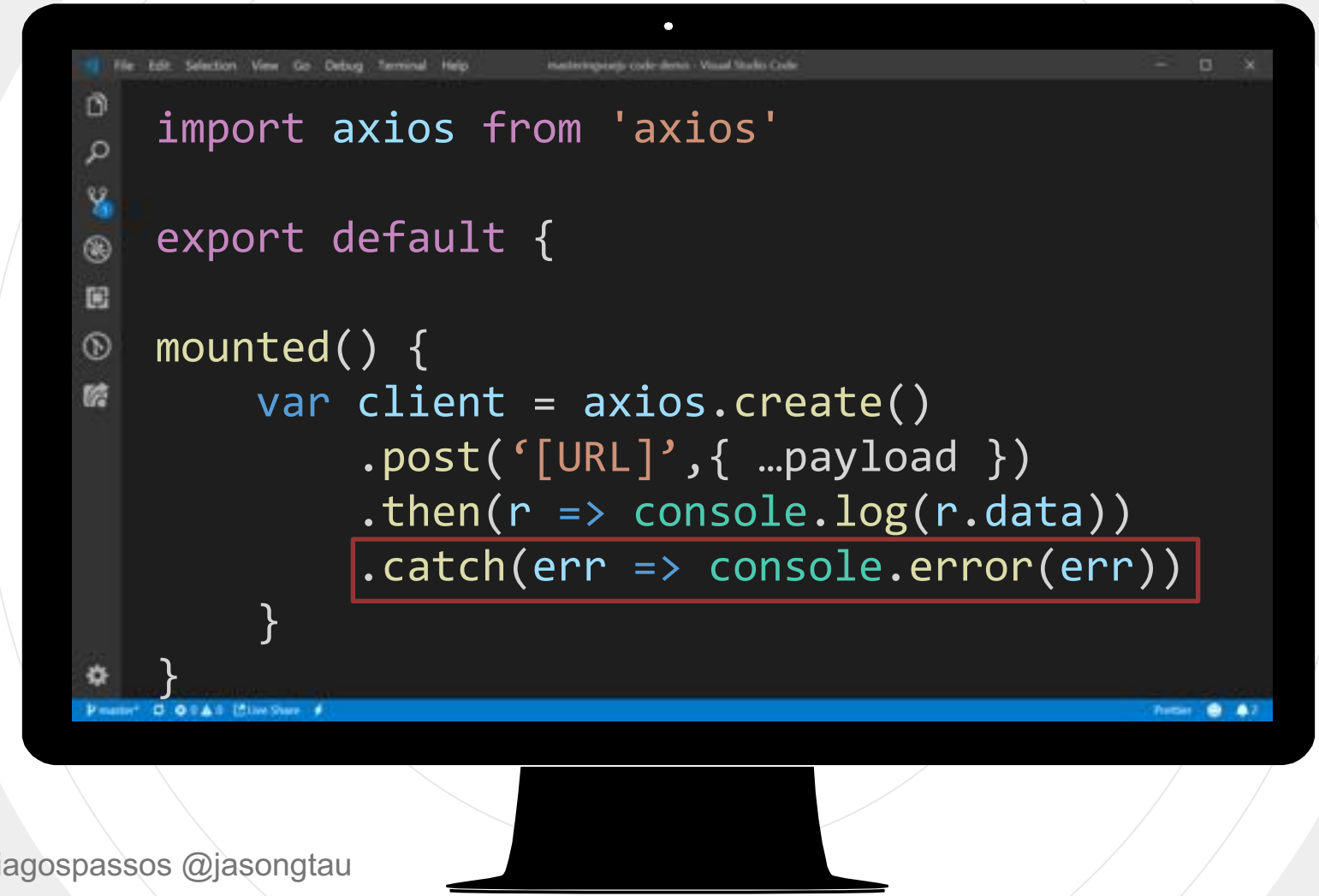
Agenda


Getting Started
Configuration
Interceptors
Put all together

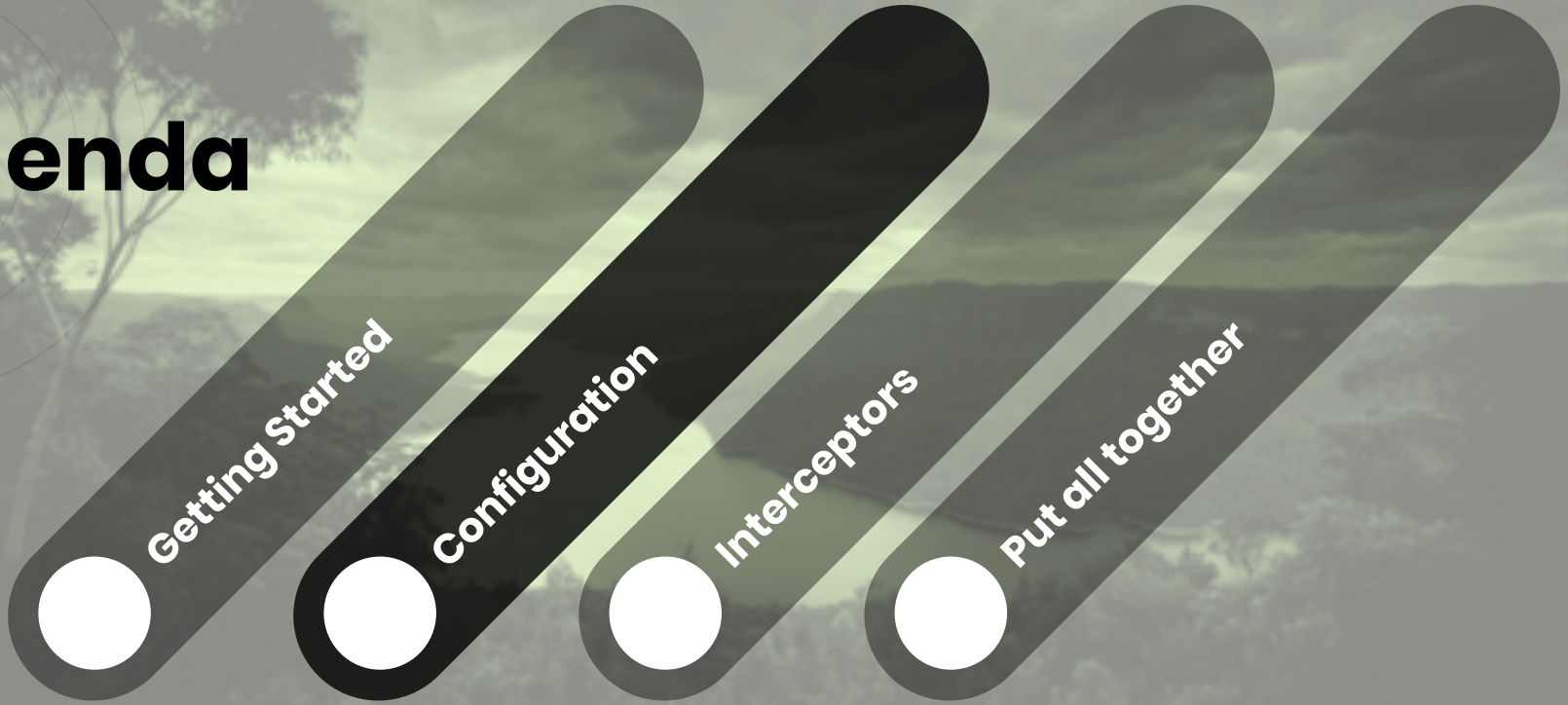

Cmder
λ npm install axios
cmd.exe


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

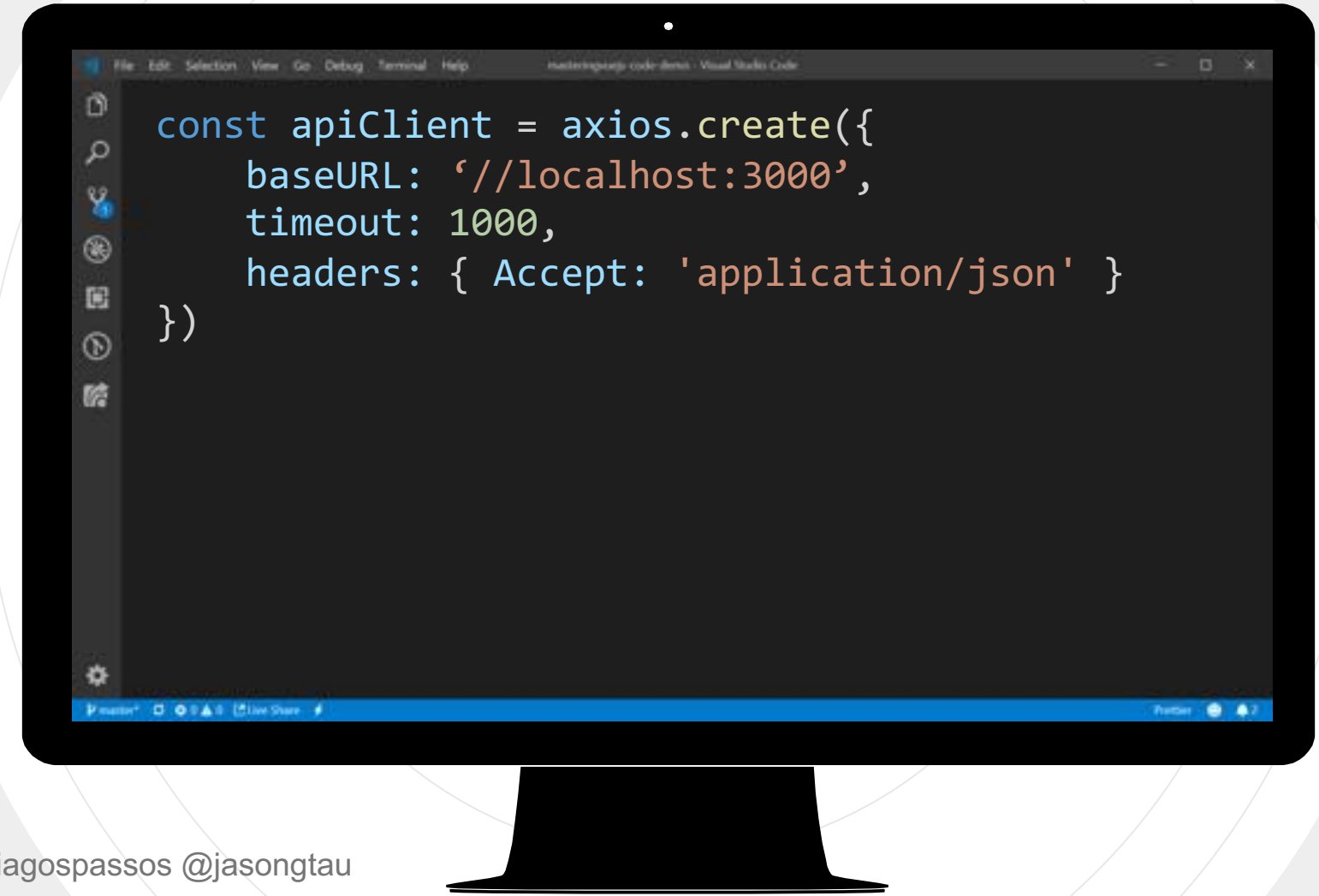

import axios from 'axios'
export default {
mounted() {
var client = axios.create()
.post('[URL]',{ …payload })
.then(r => console.log(r.data))
}
}

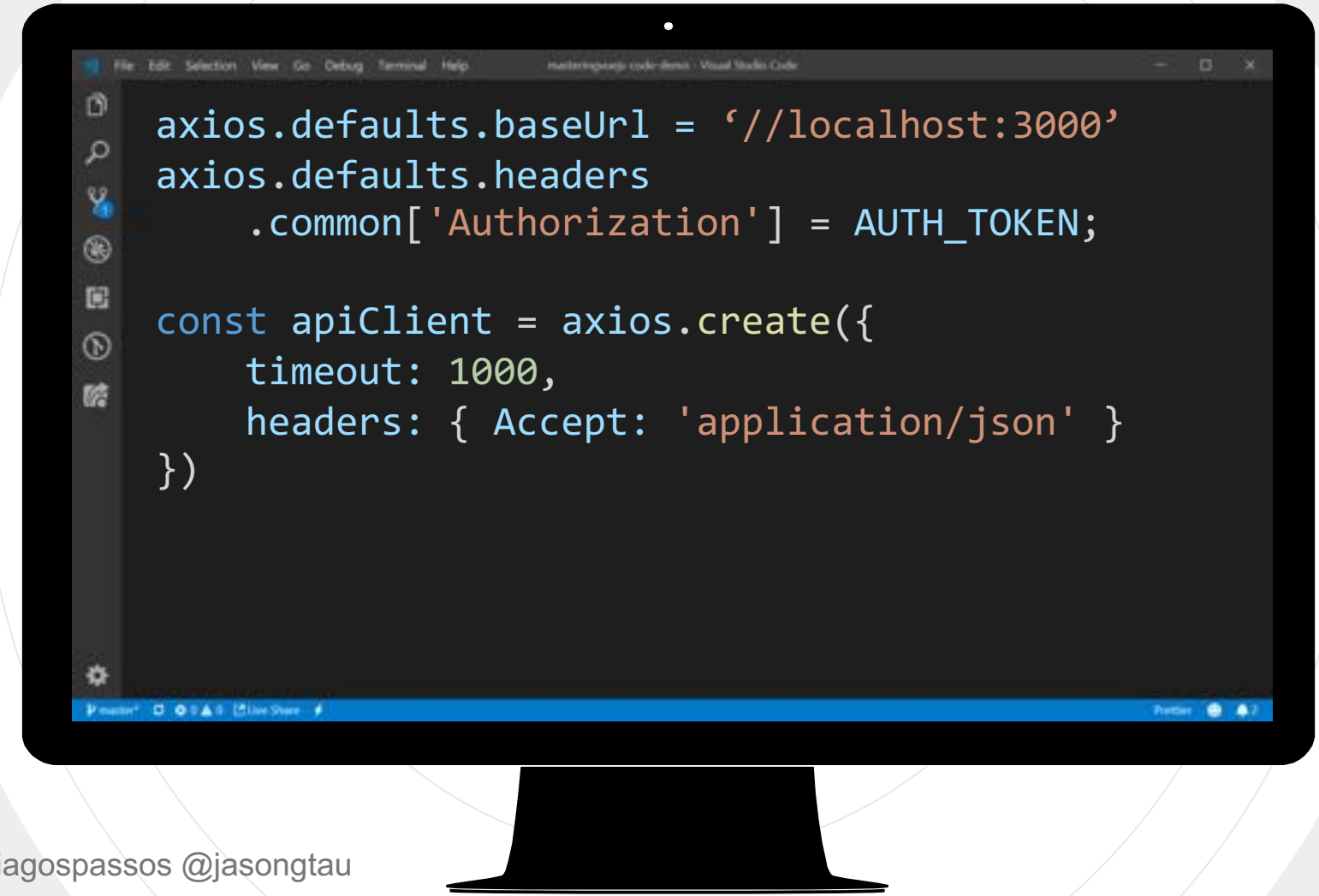

import axios from 'axios'
export default {
mounted() {
var client = axios.create()
.post('[URL]',{ …payload })
.then(r => console.log(r.data))
.catch(err => console.error(err))
}
}

Agenda

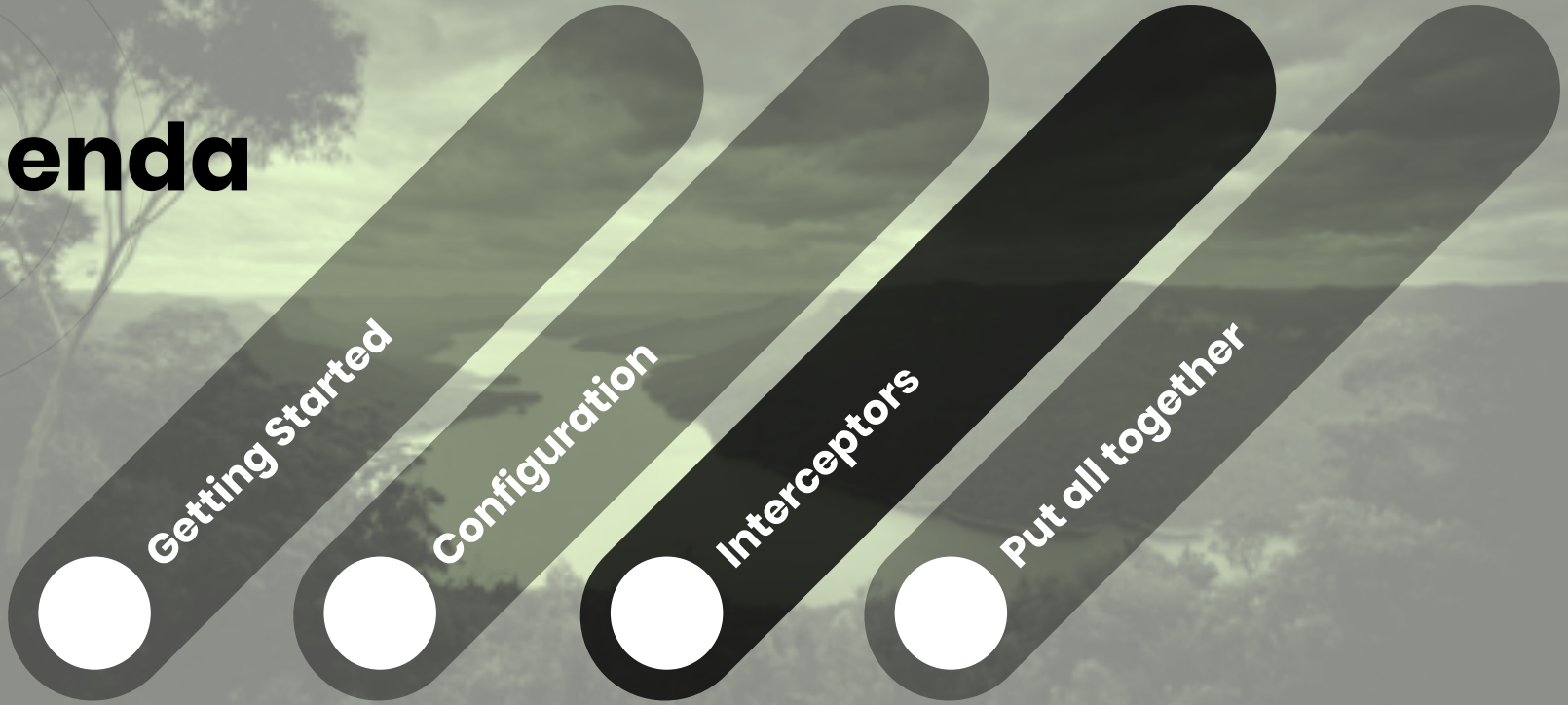

Getting Started
Configuration
Interceptors
Put all together

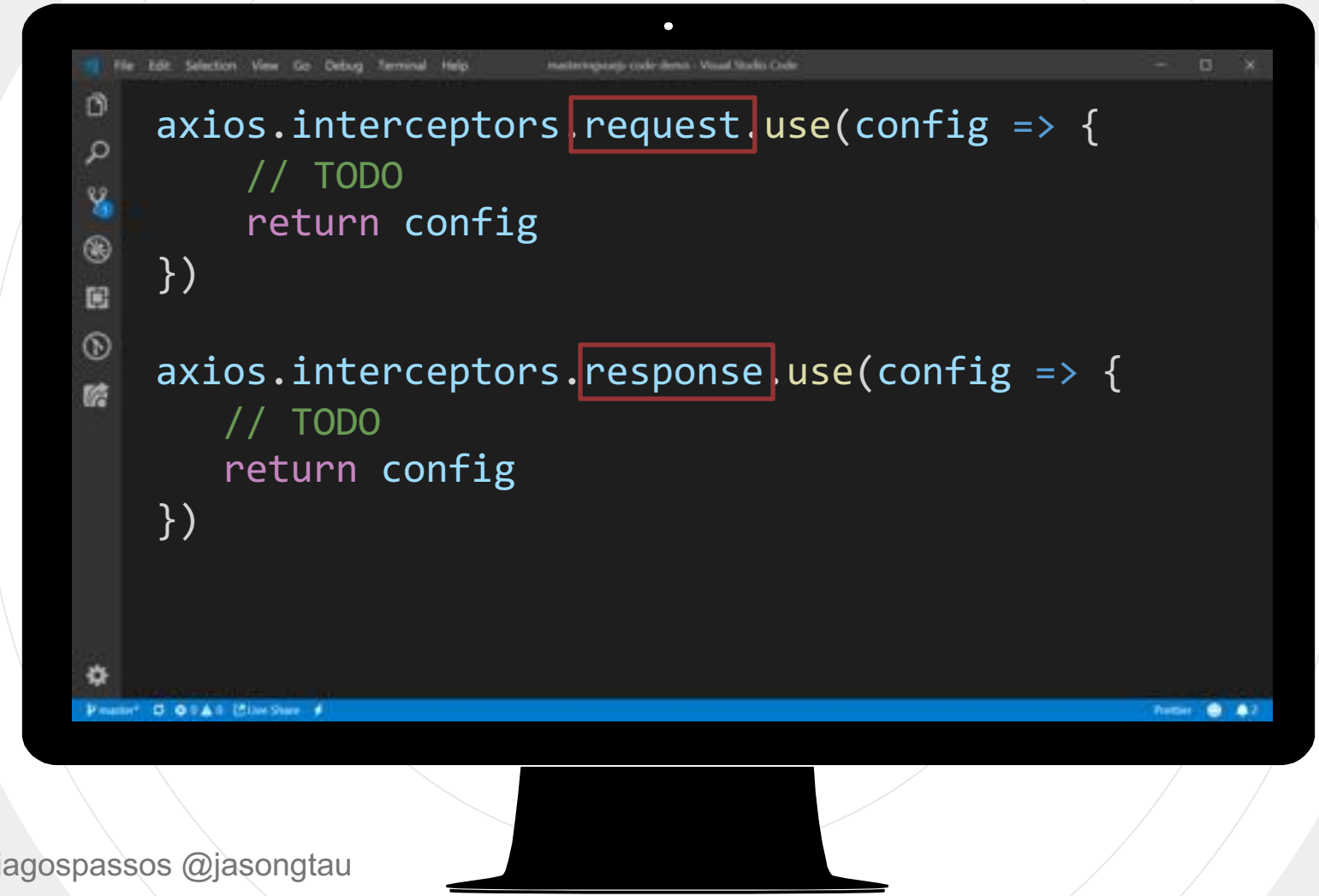

const apiClient = axios.create({
baseURL: '//localhost:3000',
})
export const SupplierService = {
getAll() {
return apiClient.get('/suppliers')
},
get(id) {
return apiClient.get('/suppliers/' + id)
},
}

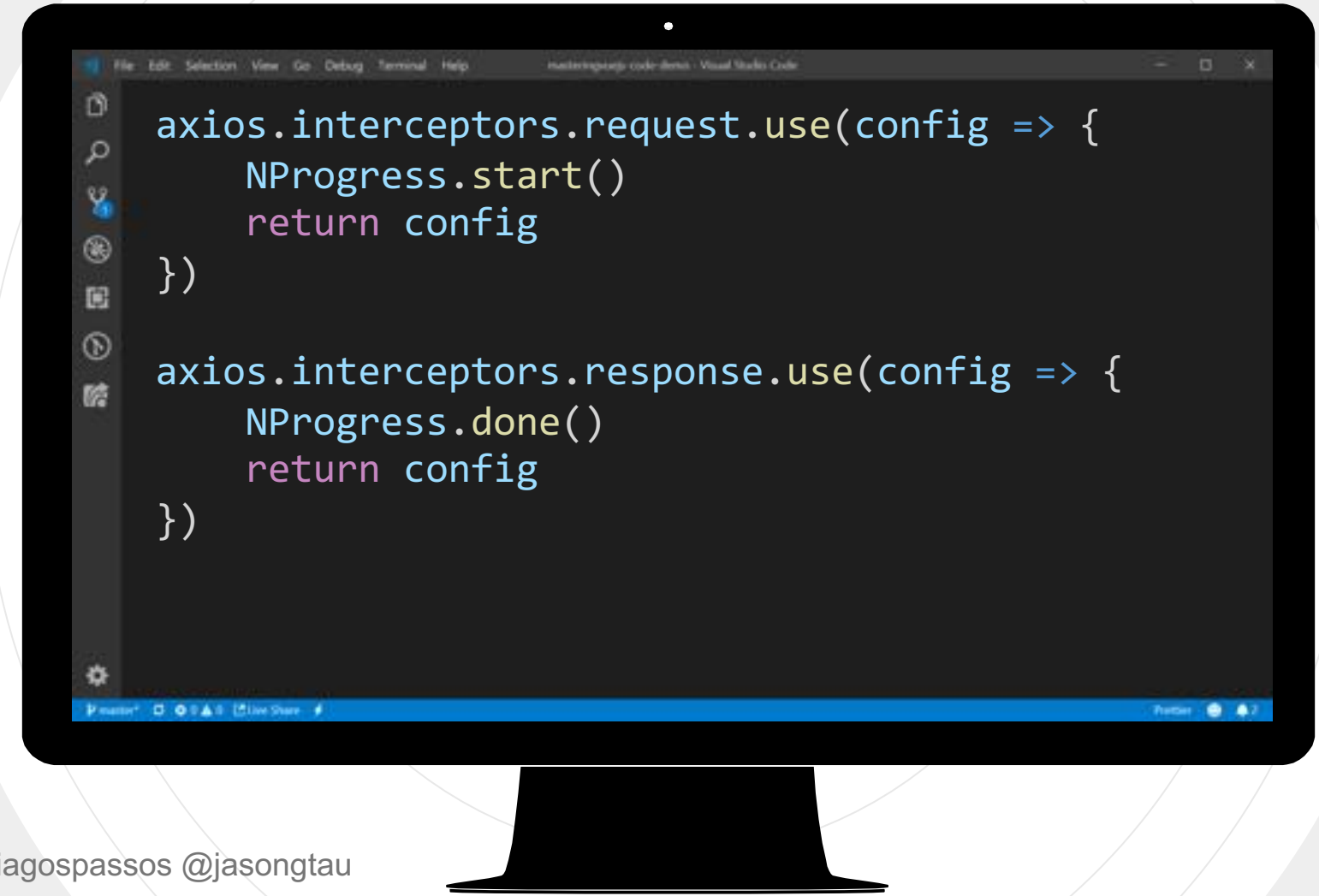

const apiClient = axios.create({
baseURL: ‘//localhost:3000’,
timeout: 1000,
headers: { Accept: 'application/json' }
})

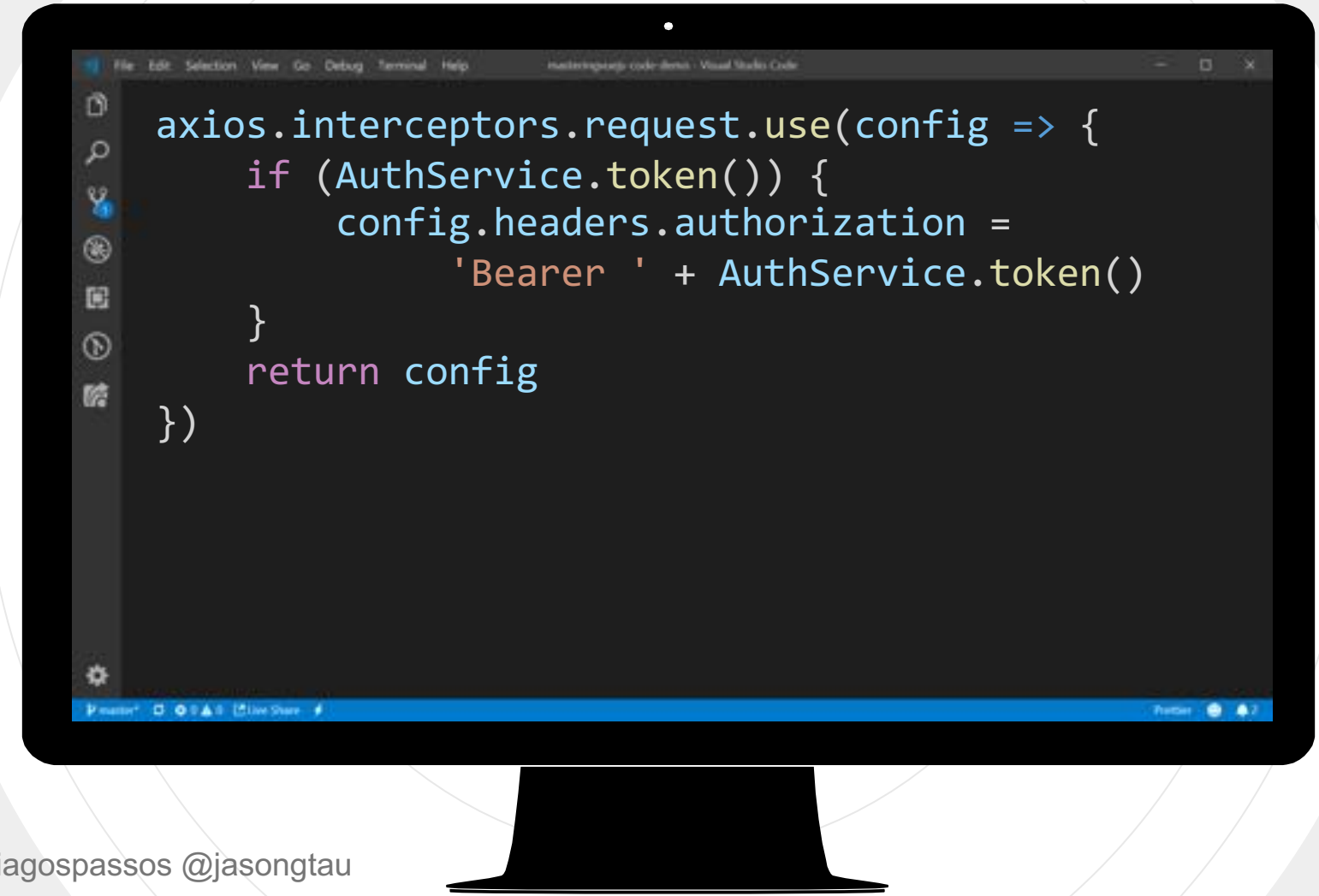

axios.defaults.baseUrl = '//localhost:3000'
axios.defaults.headers
.common['Authorization'] = AUTH_TOKEN;
const apiClient = axios.create({
timeout: 1000,
headers: { Accept: 'application/json' }
})

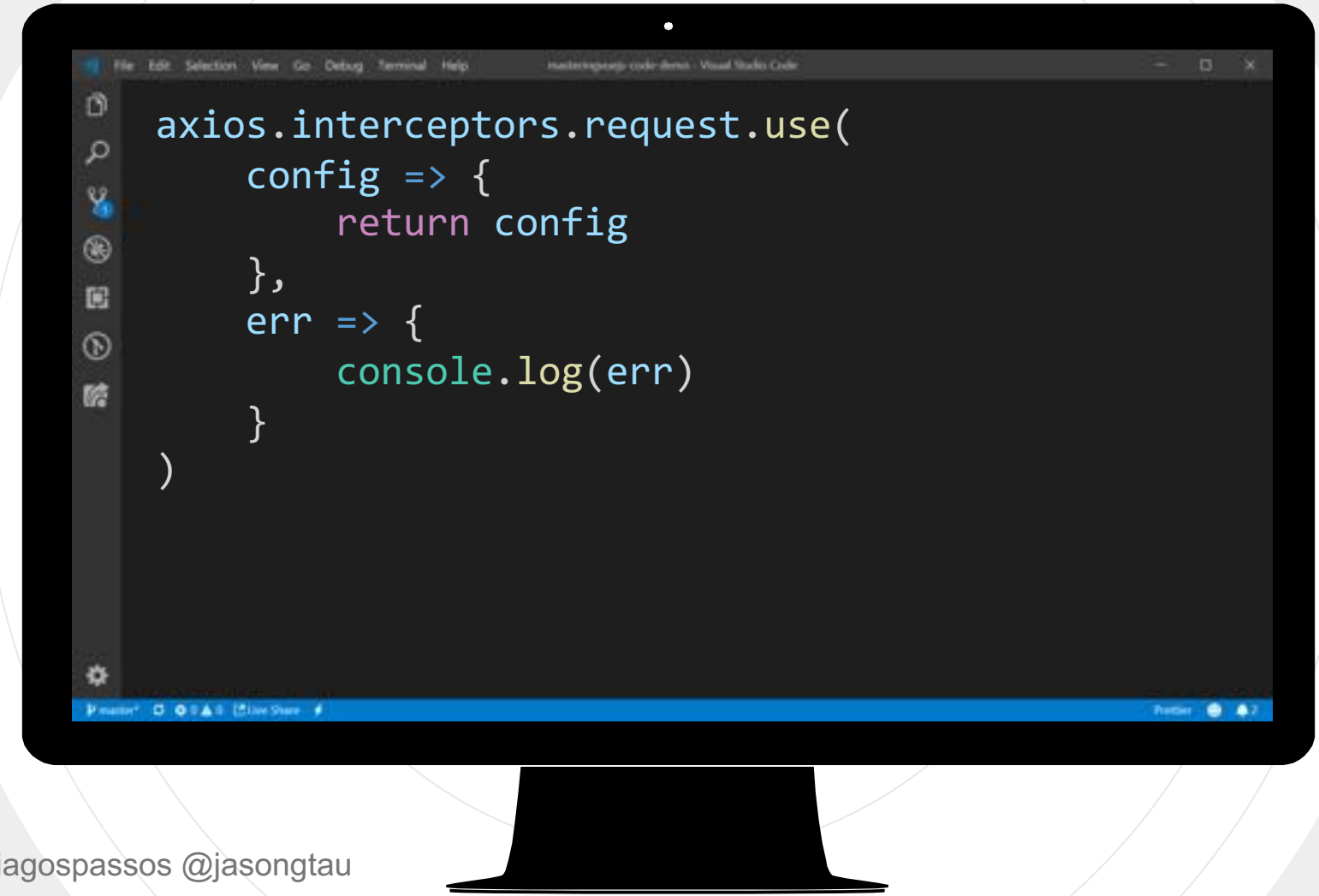
Agenda

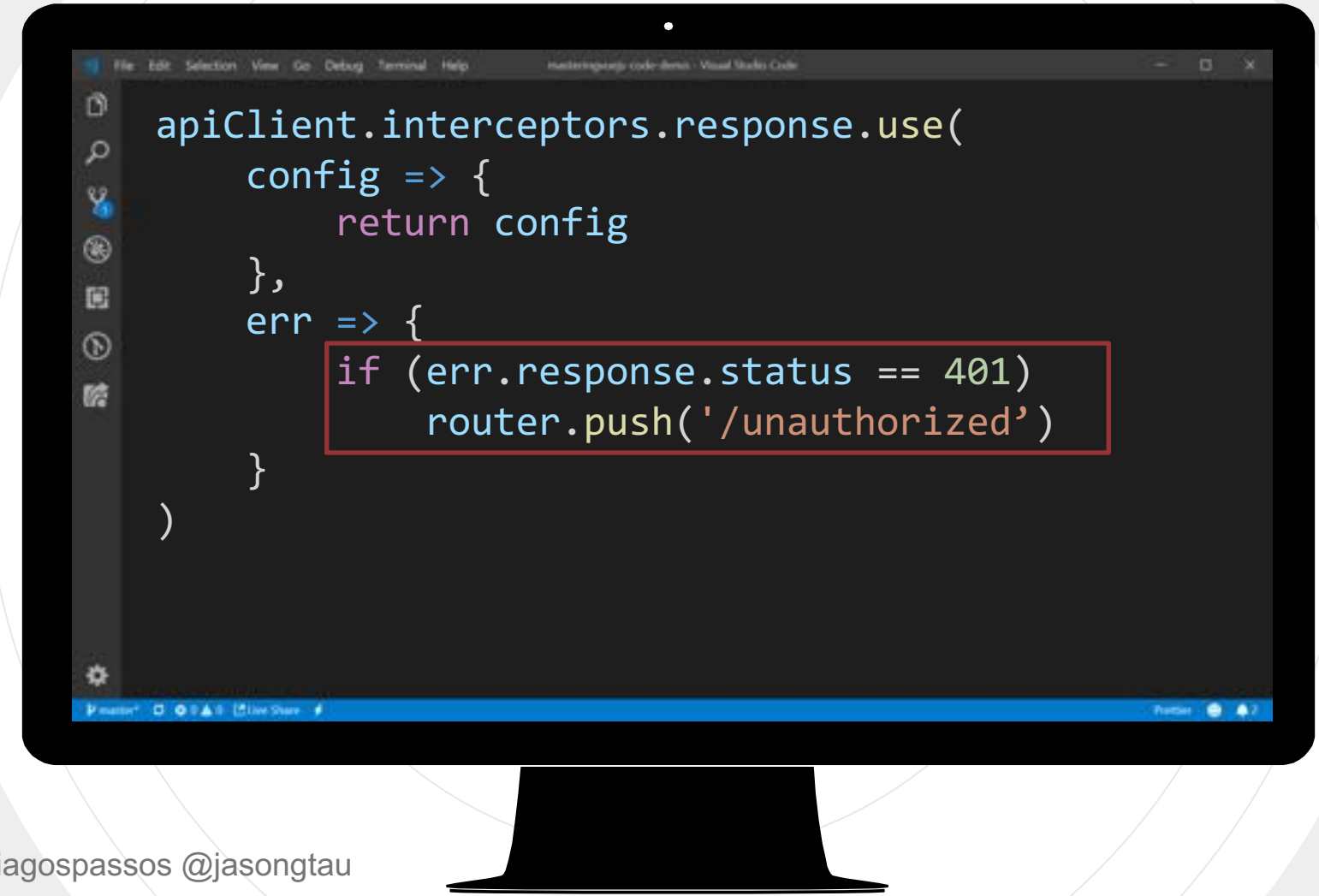

Getting Started
Configuration
Interceptors
Put all together


axios.interceptors.request.use(config => {
// TODO
return config
})
axios.interceptors.response.use(config => {
// TODO
return config
})


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


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


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


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

Workshop Code
bit.ly/vuejs-sydney-2019


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

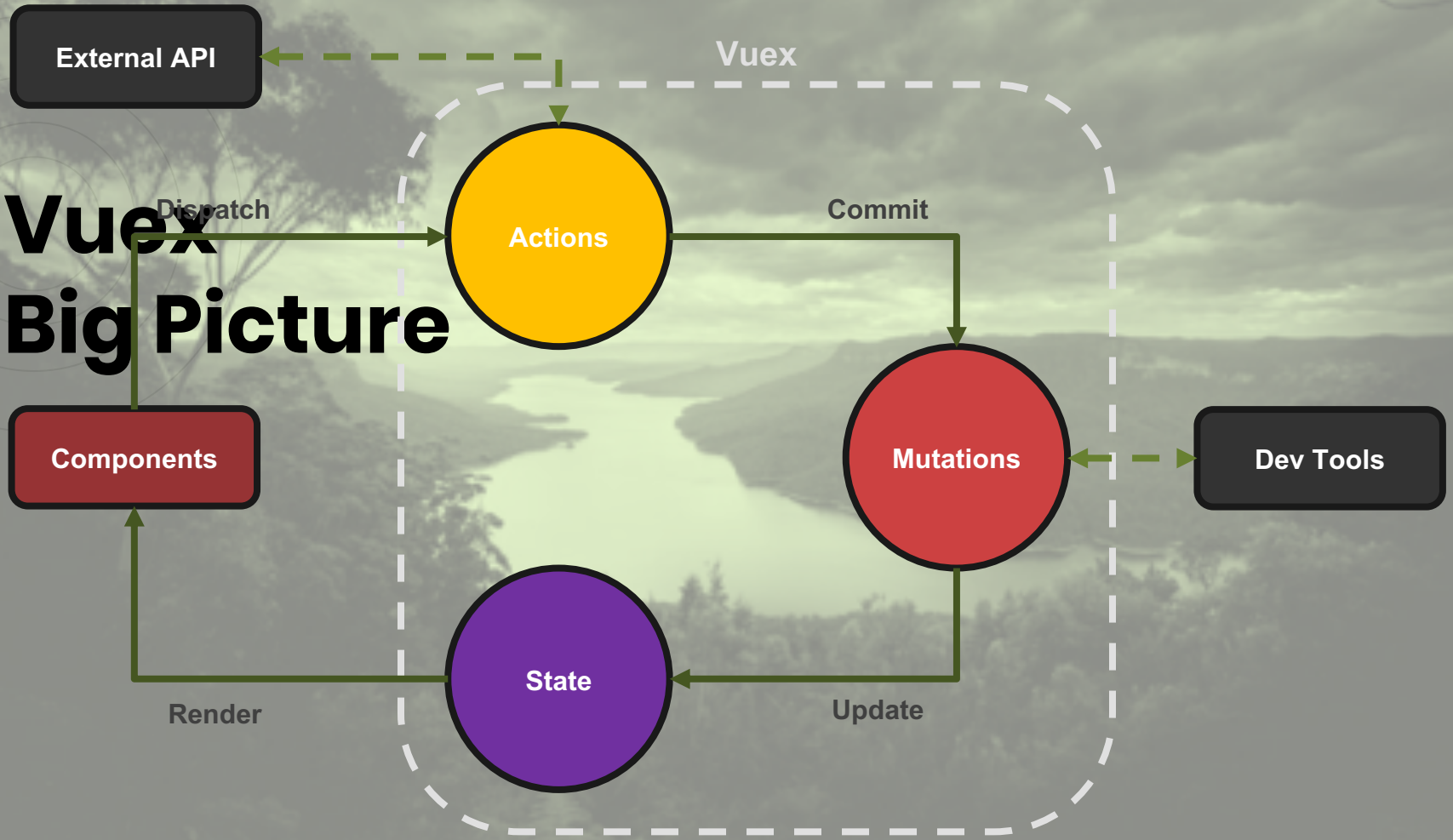
Agenda

- 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



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

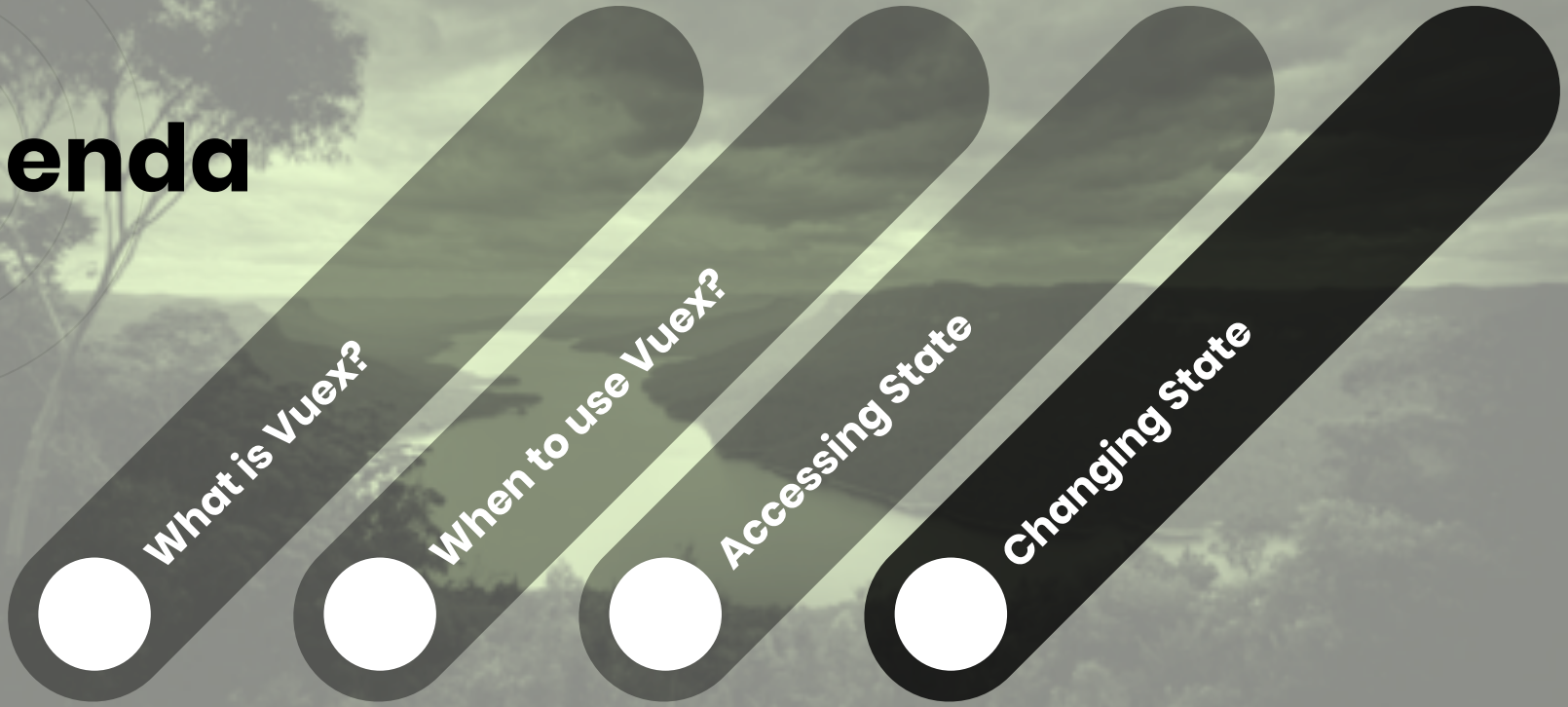
```
state: {
  userName: ‘Jason’,
  roles: [ ‘Administrator’, ‘Contributor’ ]
}
```

Core Concepts: Getters

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

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

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

Core Concepts: Mutations

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

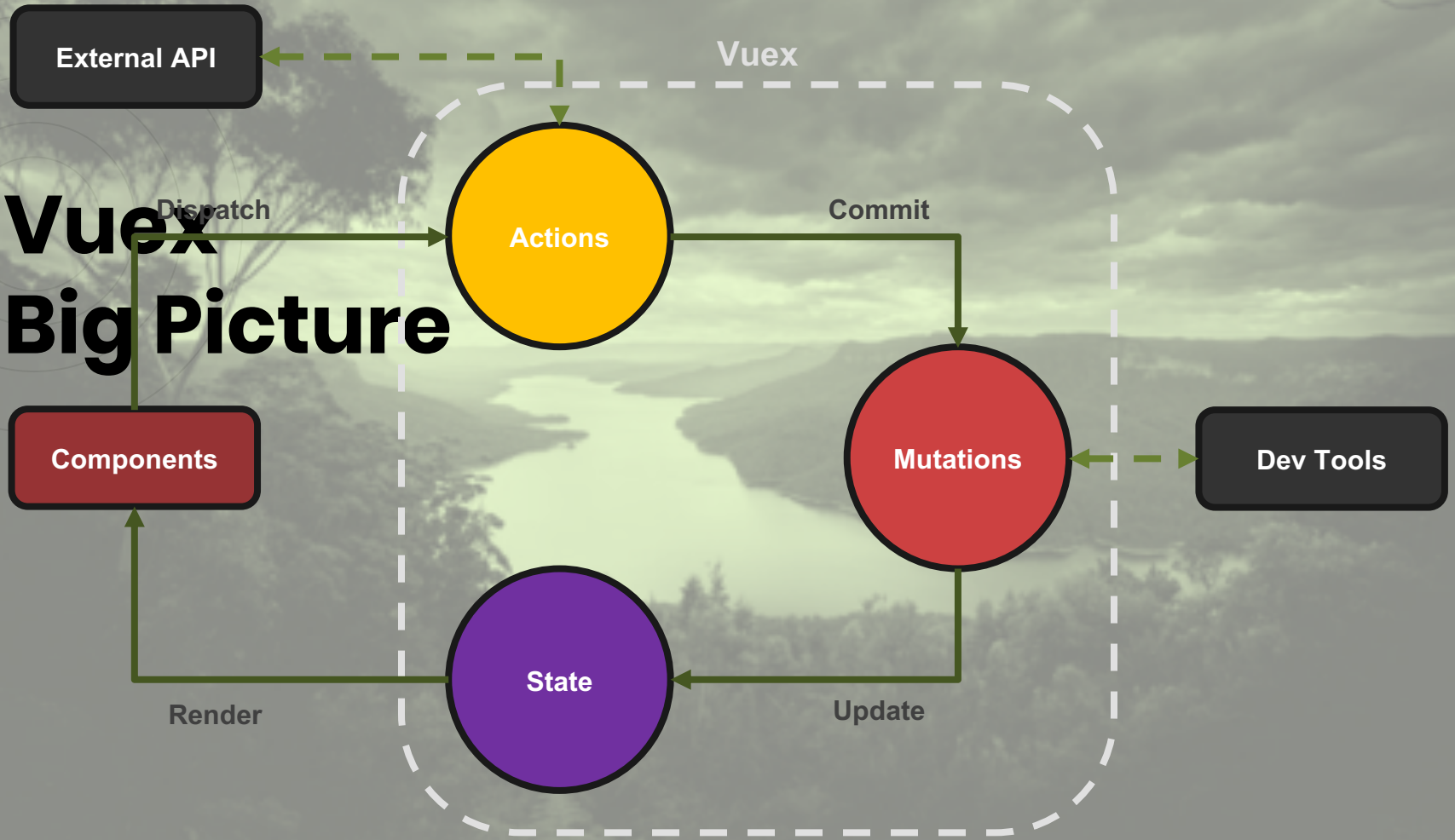
```
mutations: {
  addUser: (state, payload) => {
    return state.users.push(payload)
  }
}
```

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



Key Points

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

Key Points

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

Your Turn!

Follow the instructions to complete the code exercises and challenges

Workshop Guide
bit.ly/masteringvuejs

Workshop Code
bit.ly/vuejs-sydney-2019

Thank you!

The resources for this workshop:

Workshop Guide
bit.ly/masteringvuejs

Workshop Code
bit.ly/vuejs-sydney-2019