



Scalable Front-end Development with **Vue.js**

Galih Pratama

Head Front-end Developer at **CR**  **WDE**



Who are you?



Galih Pratama

Front-end Developer with 30+ web/apps developed.

Tech-based Edu Content Creator with 100+ videos created, since 2011.

Current Work / Projects

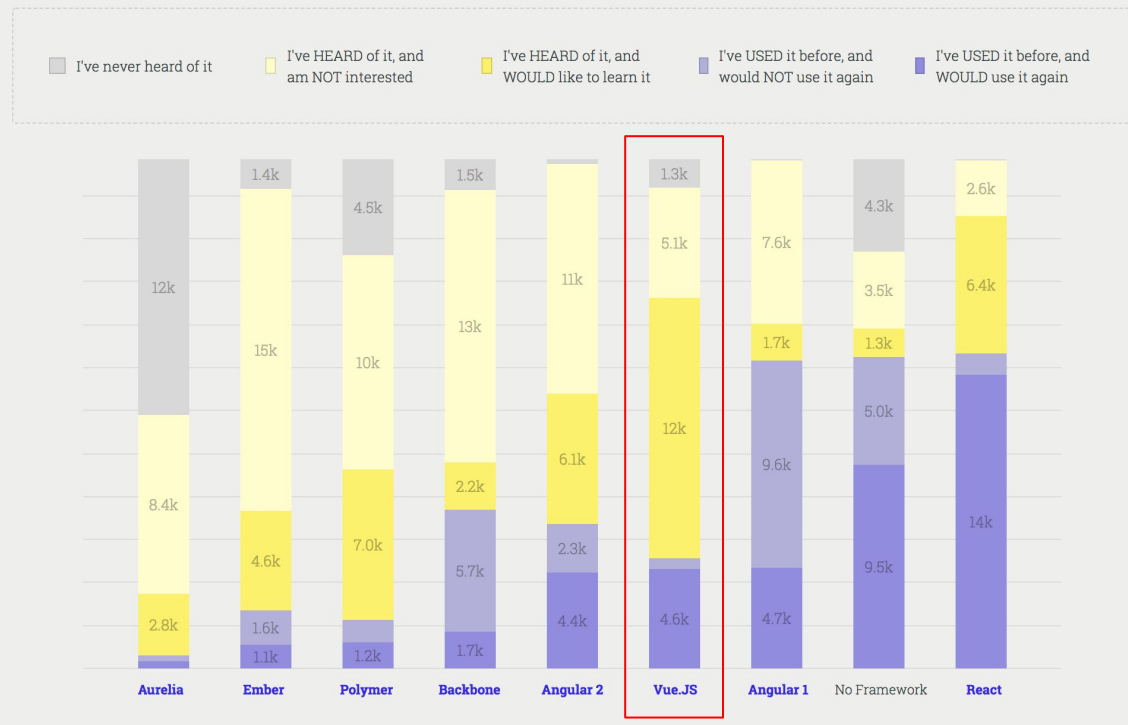




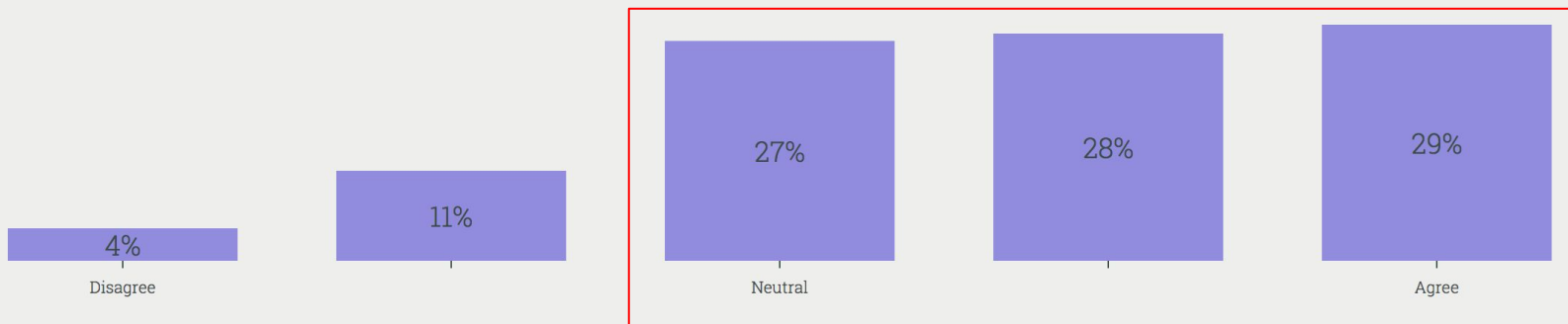
Me in a nutshell

Current State of Javascript

Per-library survey results.

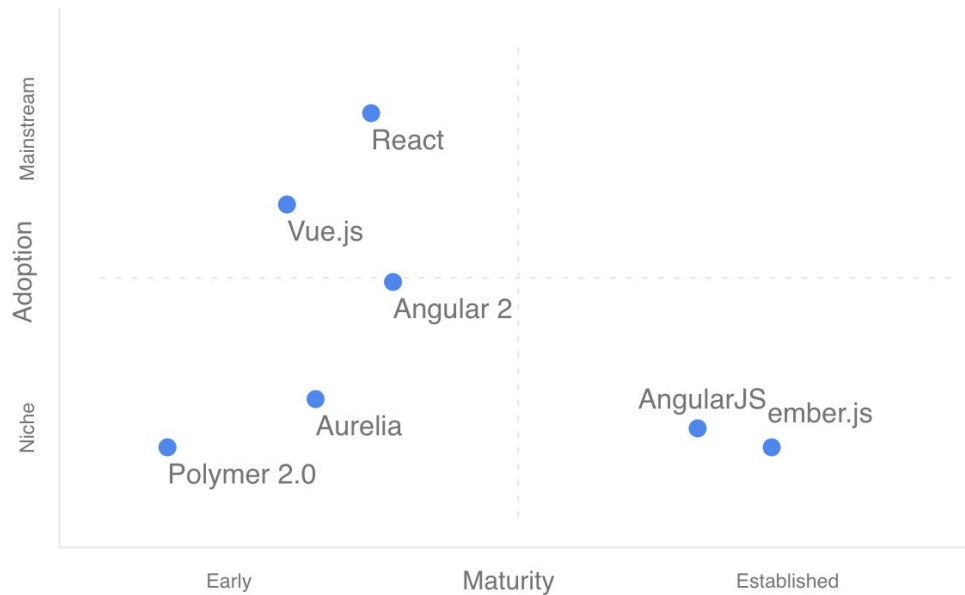


“Building JavaScript apps is overly complex right now”



Javascript Frameworks

last updated February 21, 2018 ...



stateofdev.com/c/frontend

What is “Scalable
Development” means?

Improve development speed

Decrease future technical debt

Modular

Good programming pattern

Not over-engineering things



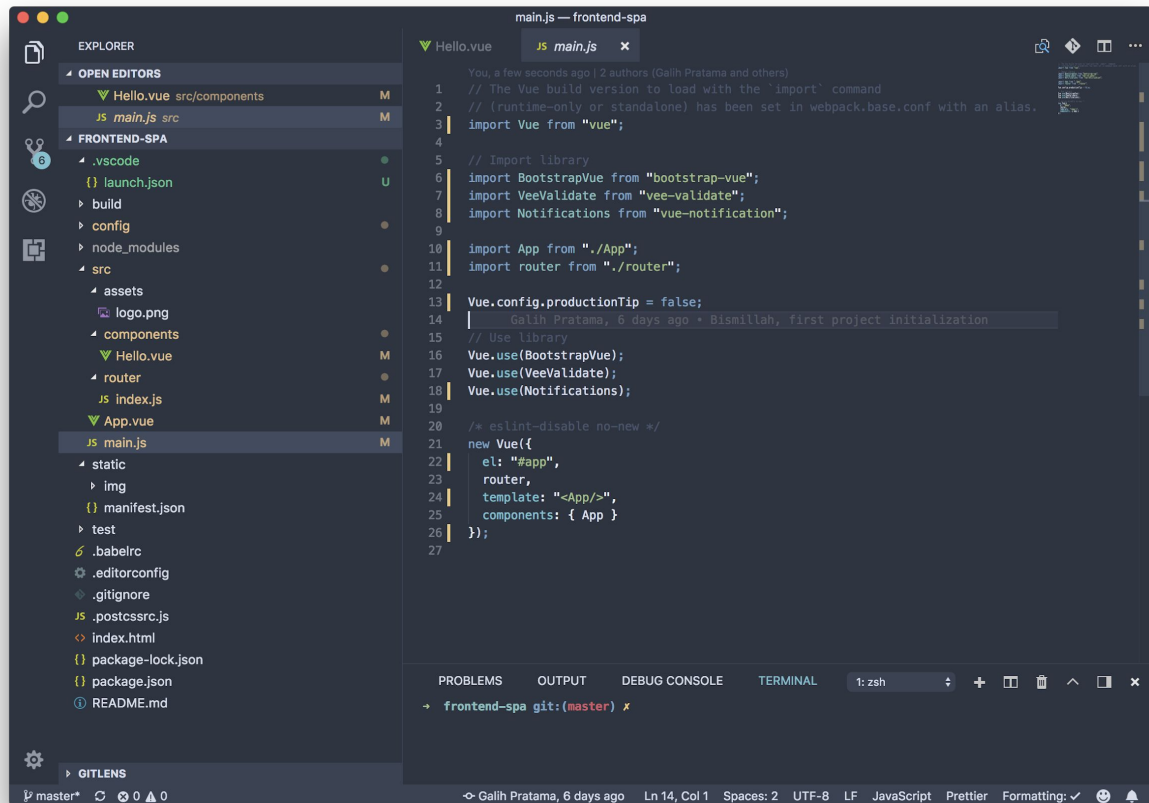
Why Vue.JS?

You can use it simply like this...

```
<!DOCTYPE html>
<html>
<head>
  <title>My first Vue app</title>
  <script src="https://unpkg.com/vue"></script>
</head>
<body>
  <div id="app">
    {{ message }}
  </div>

  <script>
    var app = new Vue({
      el: '#app',
      data: {
        message: 'Hello Vue!'
      }
    })
  </script>
</body>
</html>
```

...or like this...



Learning curve is easy

Simplicity

Hot reload

Write HTML like an usual

Flexible

Single page apps

Vue Devtools

Can used on existing app

Small file size

Less boilerplate code

Fun

Two-way data binding

Vuex

Detailed documentation

Reactive

vue-cli

Single Page Apps (SPA)

Why SPA?

App rendered in browser

Less server overhead

(no need to render anything in server, only serve html/css/js)

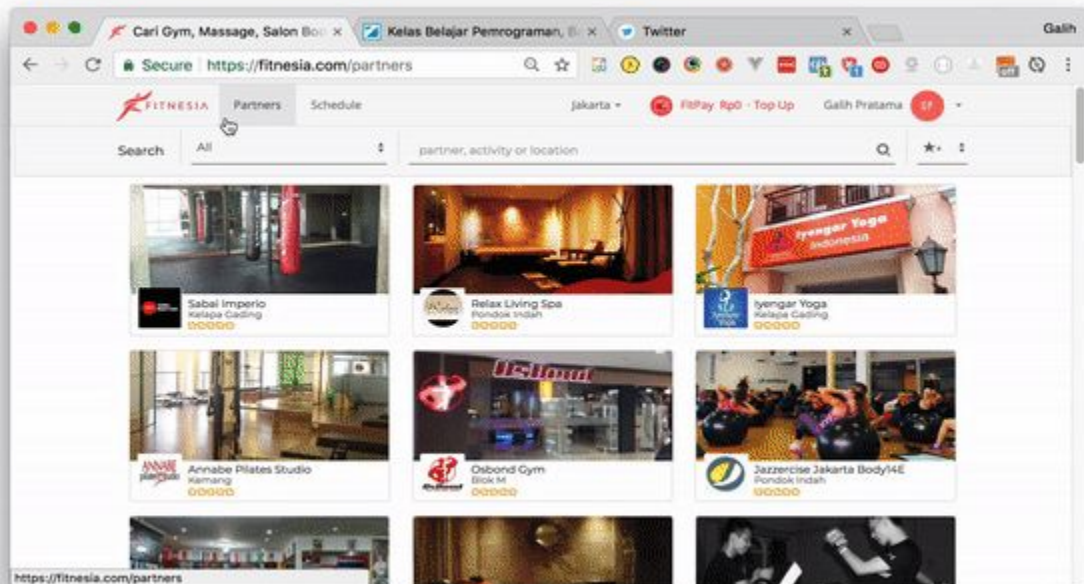
Separation of Concerns

(frontend do frontend, backend do backend)

Routing? Included.

vue-router

Wuzz wuzz



Vue Devtools

Makes developing on Vue.js even easier than it already is.

Developer Tools - http://localhost:8080/

Elements Console Sources Performance Memory Application Security Audits Network **Vue**

Ready. Detected Vue 2.4.4. Components Vuex Events Refresh

Filter components Select Format <BCollapse> Filter inspected data Inspect DOM

<Root>

<App>

<BNavToggle>

<RouterLink>

> <BCollapse> = \$vm0

<CurriculumIndex>

<RouterLink>

> <LessonIndex>

<Notifications>

router-view: /curriculum

router-view: /curriculum/lesson

type String

required false

props

accordion: null

id: "nav_collapse"

isNav: true

tag: "div"

visible: false

data

\$route

show: false

transitioning: false

computed

classObject: Object

collapse: true

navbar-collapse: true

show: false

errors: Object

items: Array[0]

formFields: Store

state

getters

Developer Tools - http://localhost:8080/

Elements Console Sources Performance Memory Application Security Audits Network **Vue**

Ready. Detected Vue 2.4.4. Components Vuex Events Refresh

Filter mutations Commit All Revert All Recording

Base State 11:39:57

route/ROUTE_CHANGED 11:40:05

route/ROUTE_CHANGED inspected active 11:40:06

Filter inspected state Export Import

state

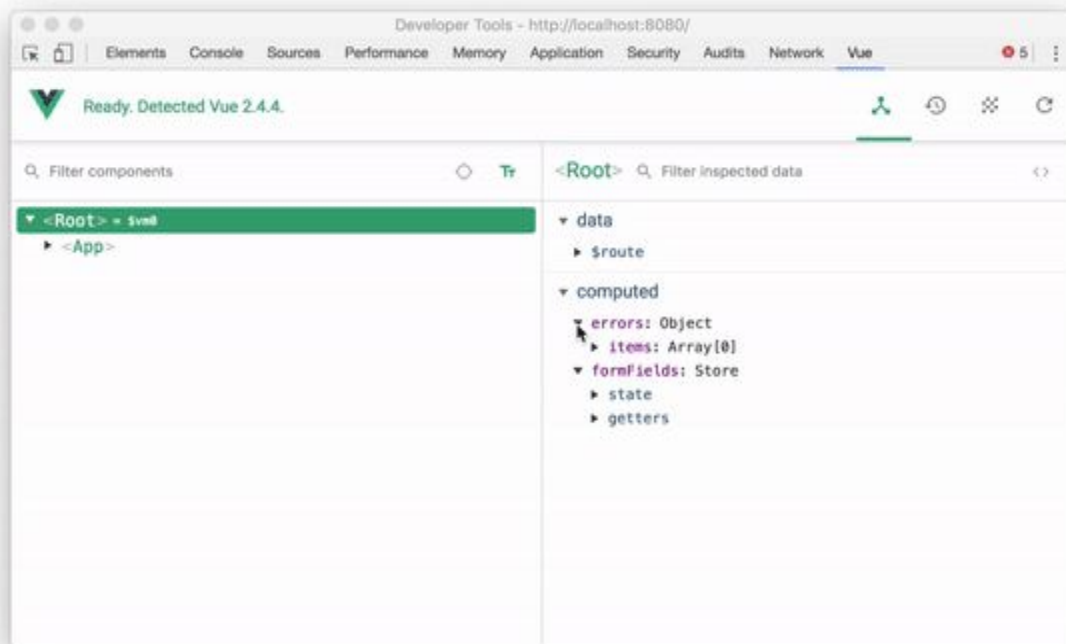
- Authentication: Object
- route: Object

getters

Authentication/isLoggedIn: true

mutation

- payload: Object
 - from: Object
 - to: Object
 - type: "route/ROUTE_CHANGED"



vue-cli

A simple CLI for scaffolding Vue.js projects.

```
vue init <templatename> <projectname>
```





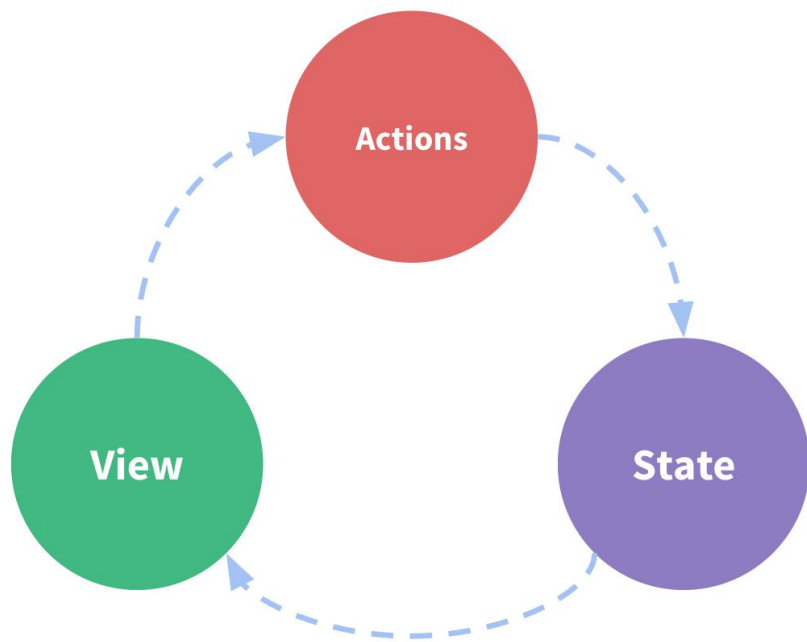
You can use it in any way!

- **webpack**
A full-featured Webpack + vue-loader setup with hot reload, linting, testing & css extraction.
- **webpack-simple**
A simple Webpack + vue-loader setup for quick prototyping.
- **browserify**
A full-featured Browserify + vueify setup with hot-reload, linting & unit testing.
- **browserify-simple**
A simple Browserify + vueify setup for quick prototyping.
- **pwa**
PWA template for vue-cli based on the webpack template
- **simple**
The simplest possible Vue setup in a single HTML file

Vuex

Vue.js on Steroids





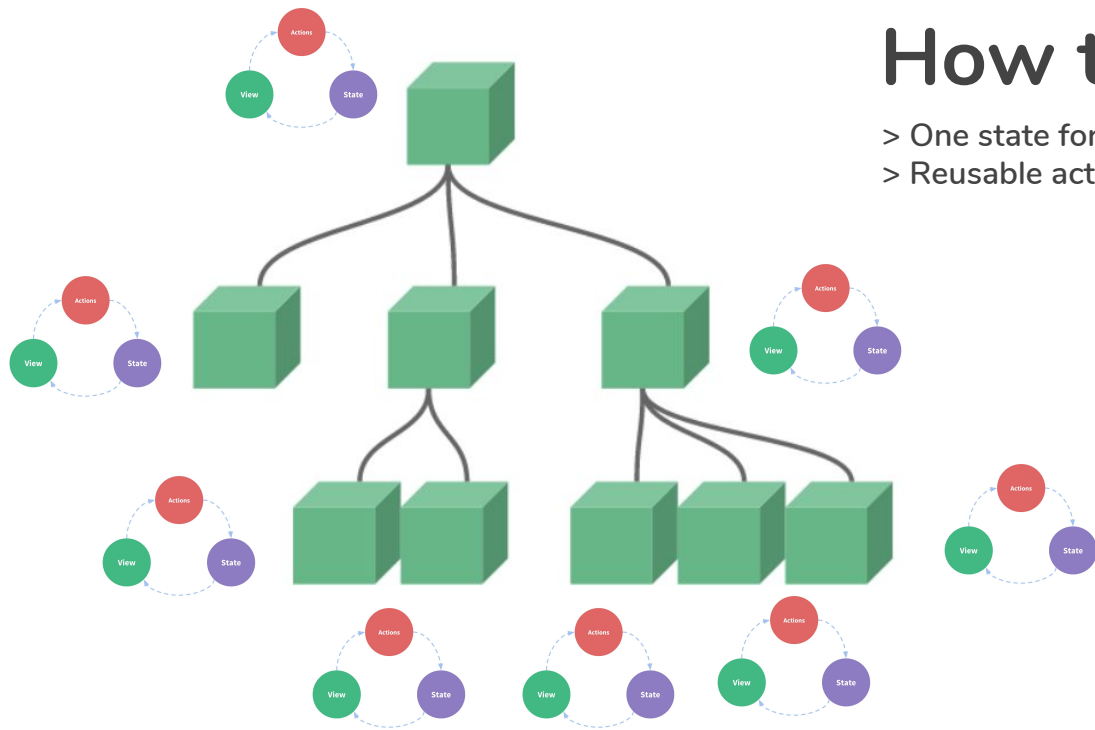
```
new Vue({  
  // state  
  data () {  
    return {  
      count: 0  
    }  
  },  
  // view  
  template: `  
    <div>{{ count }}</div>  
  `,  
  // actions  
  methods: {  
    increment () {  
      this.count++  
    }  
  }  
})
```

How Vue.js “one-way data flow” lifecycle works

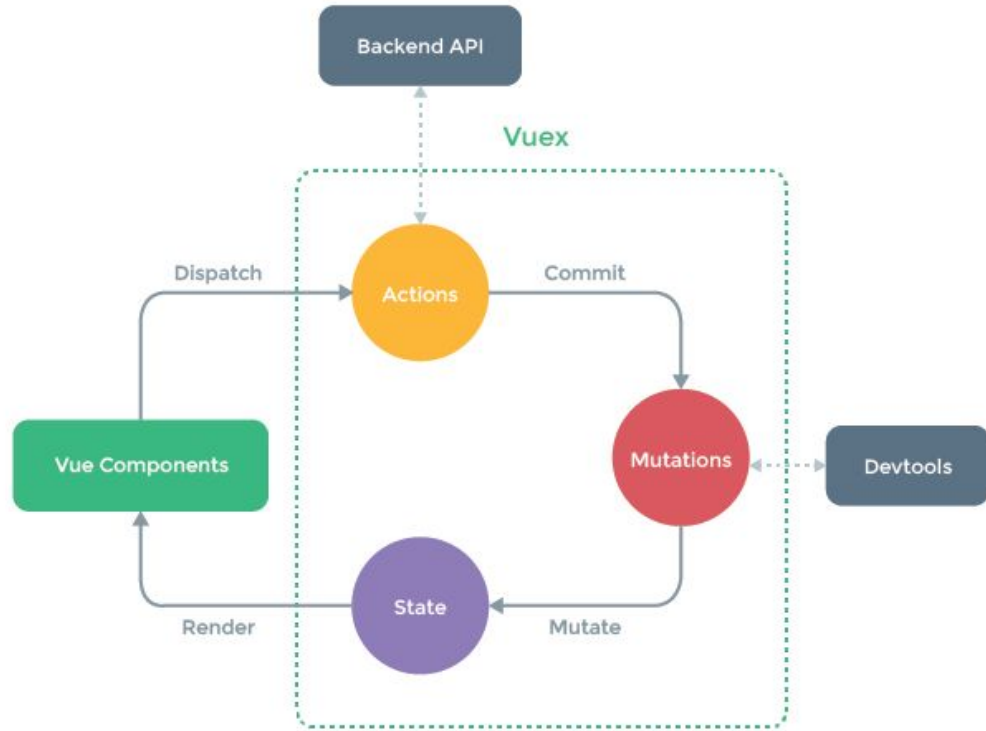
The problem?

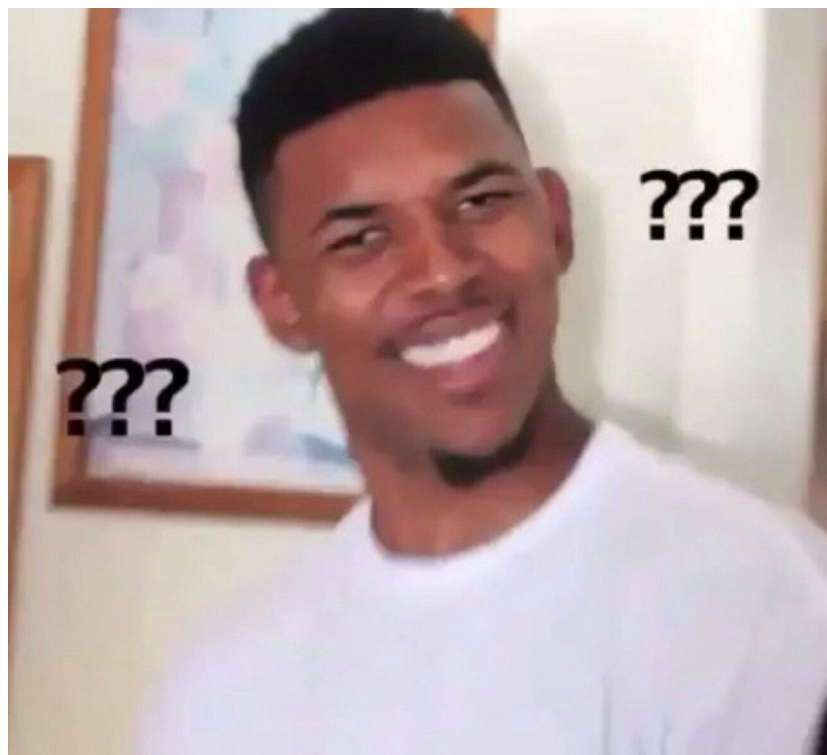
How to handle this?

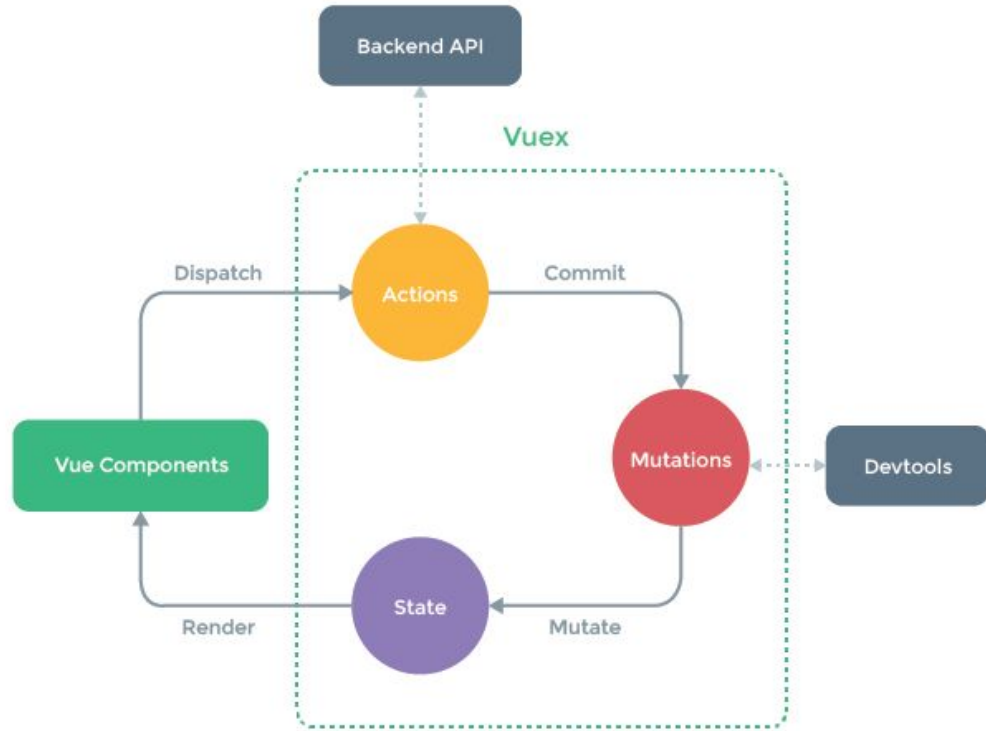
- > One state for multiple component
- > Reusable action from different component



Solution?







Vuex Store

A place to “store” globally in your app

```
import User from '@/models/User'
```

```
export default {  
  namespaced: true,
```

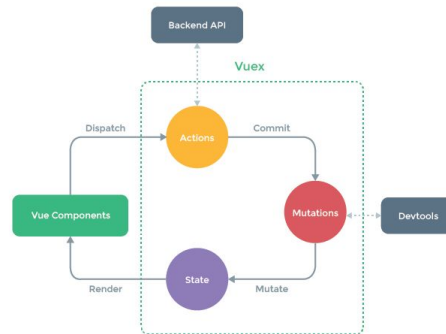
```
  state: {  
    isLoggedIn: !!localStorage.getItem("token"),  
    isAuthenticating: false,  
    username: ''  
  },
```

```
  mutations: {  
    ['LOGIN'](state) {  
      state.isAuthenticating = true;  
    },  
    ['LOGIN_SUCCESS'](state, {  
      username  
    }) {  
      state.isLoggedIn = true;  
      state.isAuthenticating = false;  
      state.username = username;  
    },  
    ['LOGOUT'](state) {  
      state.isLoggedIn = false;  
      state.username = null;  
    },  
  },
```

```
  actions: {  
    async login({ commit }, { email, password }) {
```

```
    },  
    actions: {  
      async login({ commit }, { email, password }) {  
        const login = await User.loginUser({ email, password });  
        if (login.data.success) {  
          commit('LOGIN');  
          localStorage.setItem('username', login.data.result.username);  
          localStorage.setItem('token', login.data.result.token);  
        }  
      }  
    },  
    logout({ commit }) {  
      localStorage.removeItem("token");  
      localStorage.removeItem("username");  
      commit('LOGOUT');  
    },  
  },
```

```
  getters: {  
    isLoggedIn: state => {  
      return state.isLoggedIn  
    }  
  }  
}
```





```
const moduleA = {
  state: { ... },
  mutations: { ... },
  actions: { ... },
  getters: { ... }
}

const moduleB = {
  state: { ... },
  mutations: { ... },
  actions: { ... }
}

const store = new Vuex.Store({
  modules: {
    a: moduleA,
    b: moduleB
  }
})
```

Vuex State

```
state: {  
  isLoggedIn: !!localStorage.getItem("token"),  
  isAuthenticated: false,  
  username: ''  
},
```

Vue Data

```
data(): {  
  return{  
    isLoggedIn: !!localStorage.getItem("token"),  
    isAuthenticated: false,  
    username: ''  
  }  
},
```


Vuex Action

```
actions: {  
  async login({ commit }, { email, password }) {  
    const login = await User.loginUser({ email, password });  
  }  
},
```

Vue Methods

```
methods: {  
  async login({ email, password }) {  
    const login = await User.loginUser({ email, password });  
  }  
},
```

How About SEO?



Google(bot) can index Single Page Apps

Rendering AJAX-crawling pages

Monday, December 04, 2017

The [AJAX crawling scheme](#) makes webpages accessible to Googlebot. Over time, Google engineers have significantly improved rendering of JavaScript for Googlebot. Given these advances, in the second quarter of 2018, we'll be switching to rendering these pages on Google's side, rather than on requiring that sites do this themselves. In short, we'll no longer be using the AJAX crawling scheme.

down. Over time, Google engineers have significantly improved rendering of JavaScript for Googlebot. Given these advances, in the second quarter of 2018, we'll be switching to rendering these pages on Google's side, rather than on requiring that sites do this themselves. In short, we'll no longer be using the AJAX crawling scheme.

As a reminder, the AJAX crawling scheme accepts pages with either a `#!` in the URL or a ["fragment meta tag"](#) on them, and then crawls them with an `"?_escaped_fragment_="` in the URL. That escaped version needs to be a fully-rendered and/or equivalent version of the page, created by the website itself.

With this change, Googlebot will render the `#!` URL directly, making it unnecessary for the website owner to provide a rendered version of the page. We'll continue to support these URLs in our search results.

We expect that most AJAX-crawling websites won't see significant changes with this update. Webmasters can double-check their pages as detailed below, and we'll be sending notifications to any sites with potential issues.



Lucamug

[Follow](#)

Aug 19, 2017 · 9 min read

SPA and SEO: Google (Googlebot) properly renders Single Page Application and execute Ajax calls

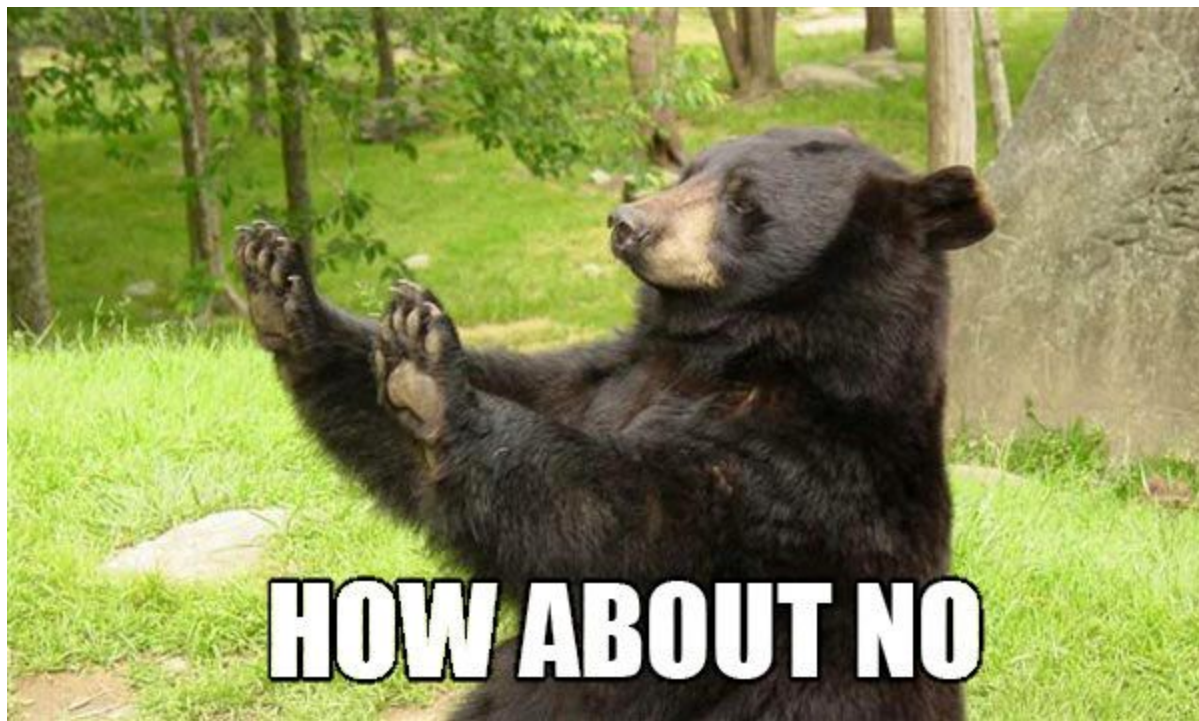


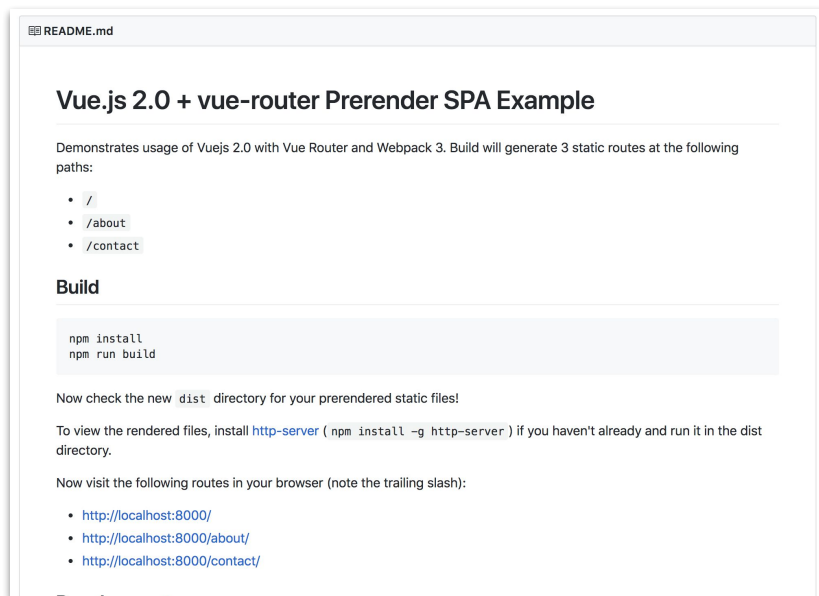
SPA & Googlebot

- Googlebot run the Javascript on the page and the Ajax calls are properly executed
- Googlebot waits between 5 to 20 seconds before taking a snapshot of each page

More :

<https://medium.com/@l.mugnaini/spa-and-seo-is-googlebot-able-to-render-a-single-page-application-1f74e706ab11>





Prerender?

Use prerender-spa-plugin



NUXT

Universal Vue.js Applications

GET STARTED

GITHUB (V1.4.0)

Server-side rendering?

Use Nuxt.js instead

Conclusion



Conclusion

1. Building javascript apps is somewhat **confusing**
2. Need to **scale up** without make it more complicated.
3. Vue.JS make it **easier** without sacrificing scalability.
4. Official project generator using **vue-cli**
5. Ready to use **devtools** with zero configuration
6. Use **Vuex** in large-scale apps
7. **SEO** is safe (like Google said)

Thank....

... one more thing

JS Code-Sharing Workshop #1

Saturday
31/Maret
08.30 — 12.30

@Bandung Creative
Hub - Ruang Kelas
Jl. Laswi no.7, Kacapiring



Featured Trainer :

Galih Pratama

Head Front-End - Crowde.co

Workshop Topic :

**Scalable Web Development
with Vue.JS**

RSVP

bit.ly/bjsworkshop1

Bandung JS

CROWDE

GEEK
HUNTER

STARTUP
BANDUNG

RASI

Want to know how
to build it ?

Thank you!

Questions?



galpratama



pratama.galih



hi@galihpratama.net