



Intro to Vue.js



What is Vue.js?

- Framework for building user interfaces
- Focused on the view layer
- Easily integrate with other libraries
- Current version: 2.4.1

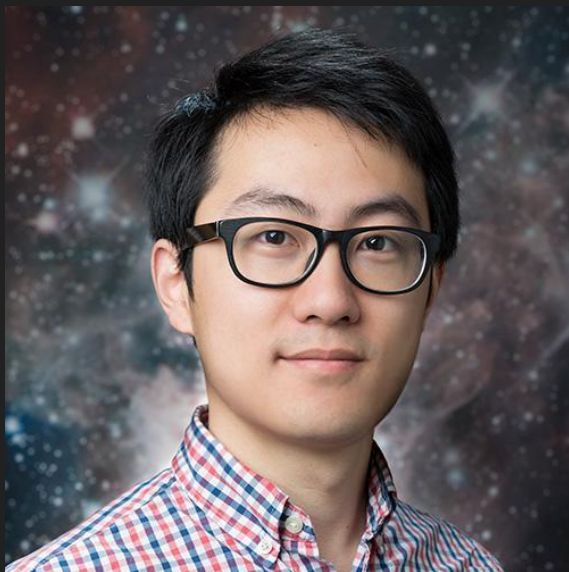


Key features

- Simple(low barrier of entry)
- Lightweight (~24kb min+gzip)
- Fast
- Reactive
- Component based



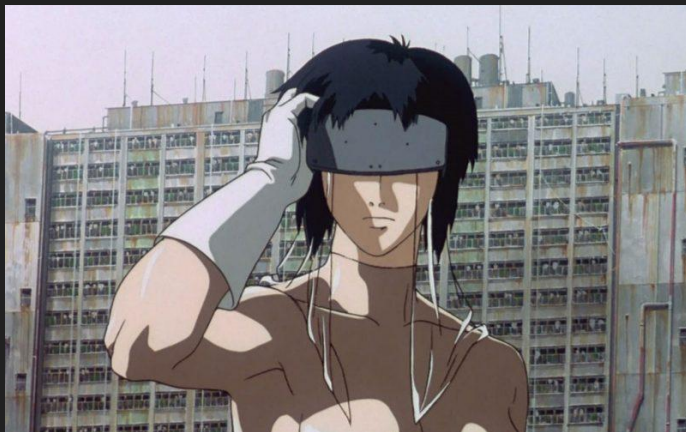
History



Creator:	Evan You - ex employee of Google, and Meteor Dev Group
First commit:	Jul 2013
First release:	Feb 2014
Discovered by Laravel	Apr 2015
Release of second(current) version:	Sep 2016



Ghost in the Shell

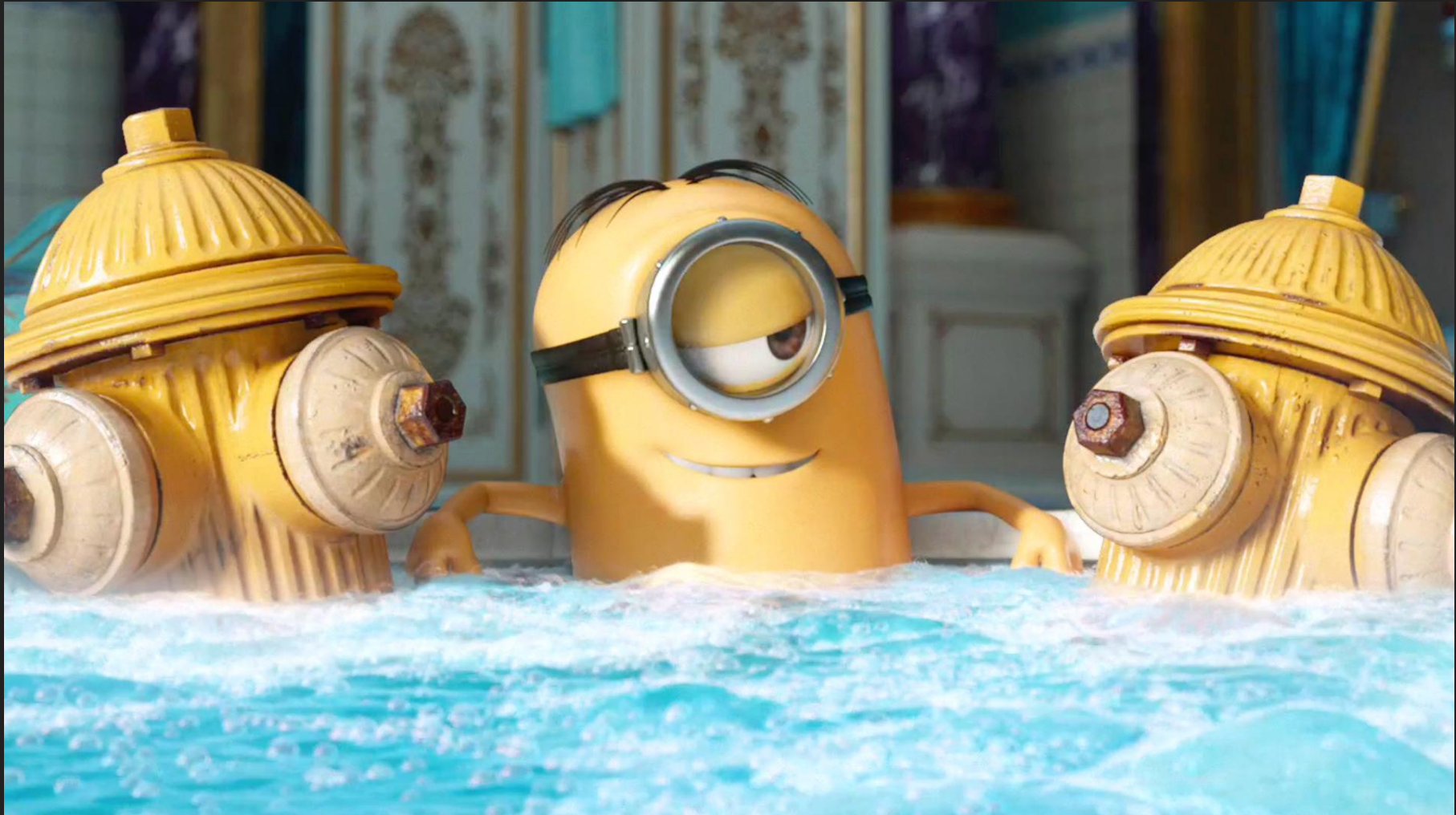


Evangelion





How it relates to Vue.js?





Company which use



GitLab



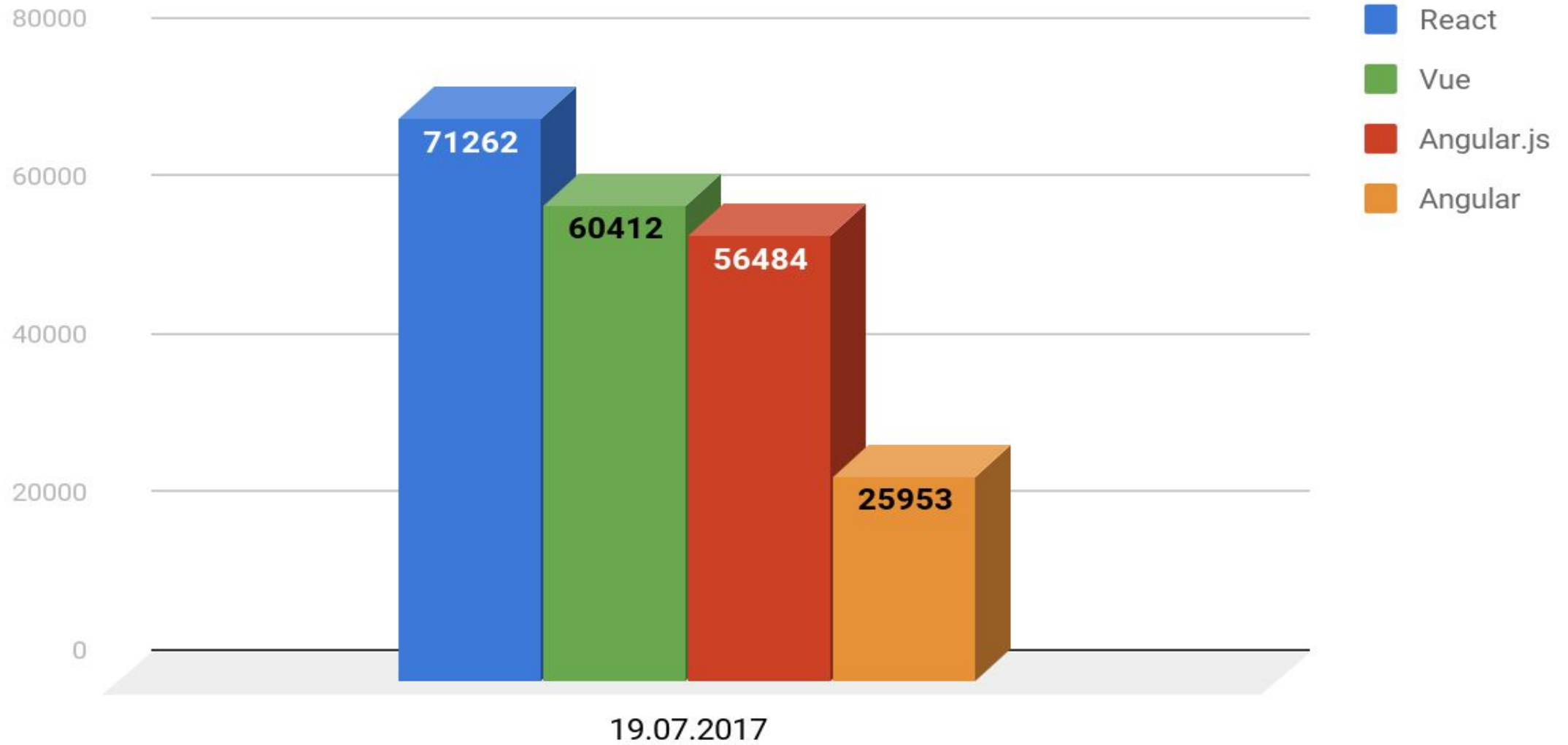
euro
news.

Tencent 腾讯

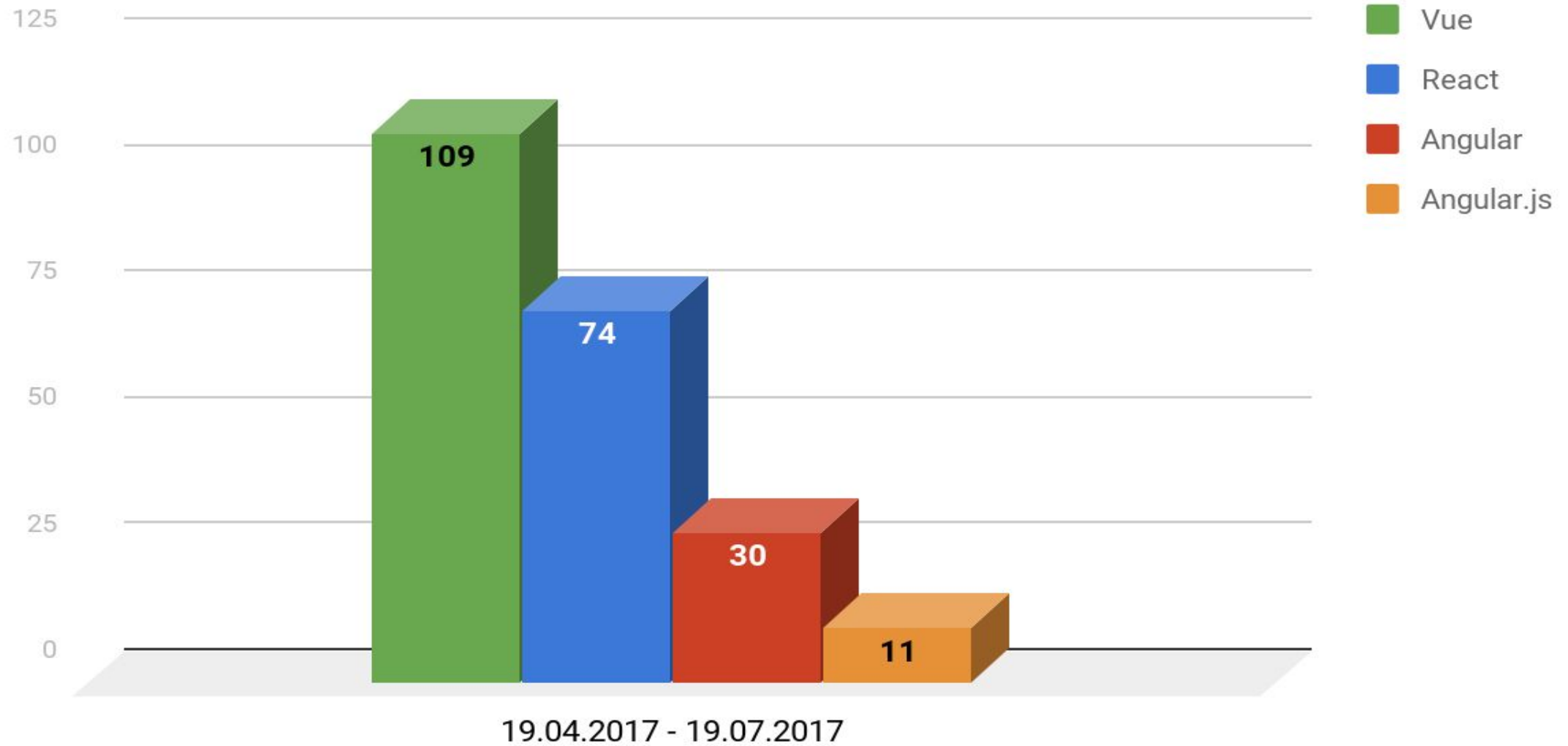


facebook.

Github stars(total)

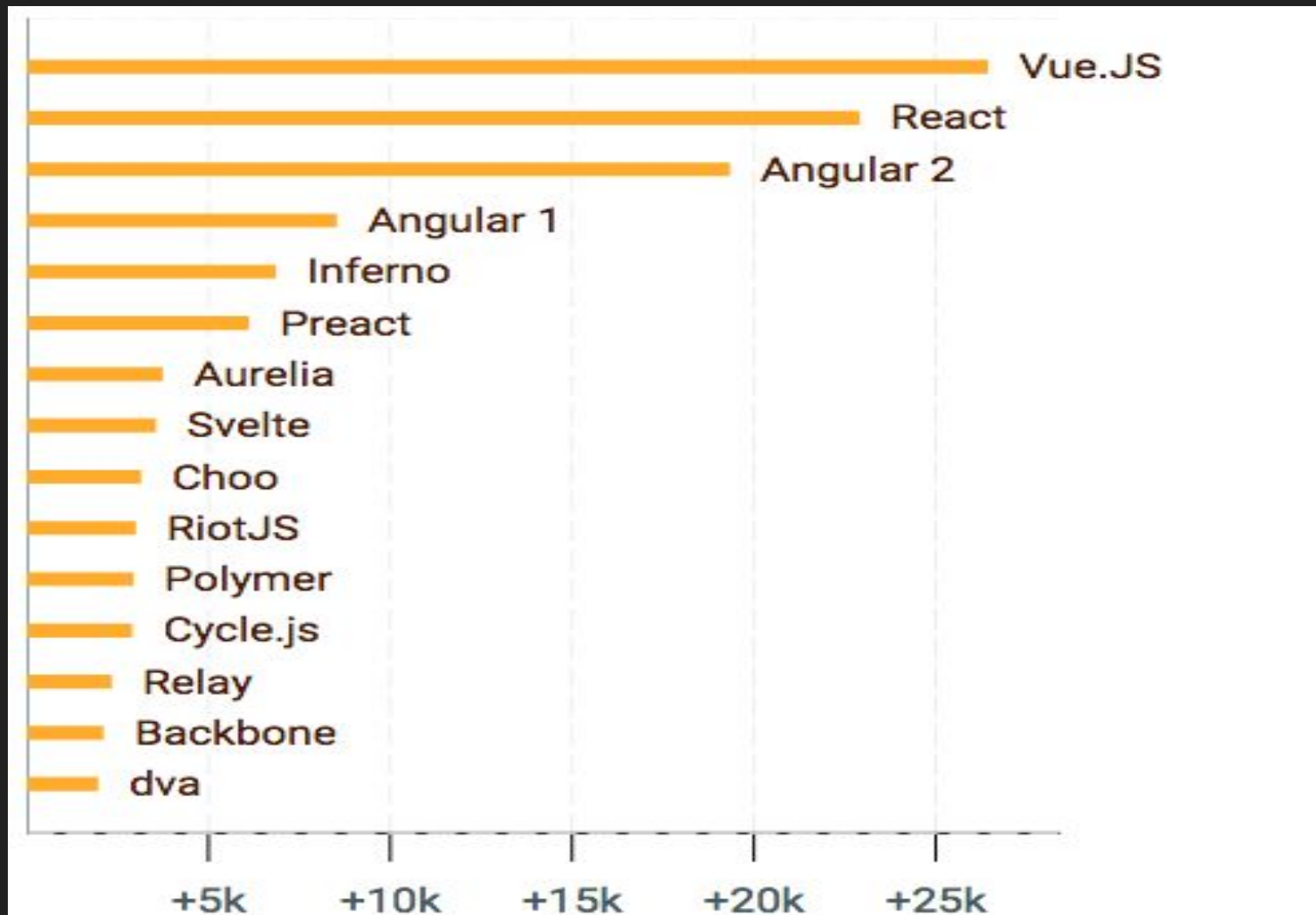


Github stars(average per day, during last 3 months)





@ 2016 JavaScript Frameworks Rising Stars



Rendering time(10000 items), ms



[@ Others Benchmarks](#)



Vue can also!

React

Components

JSX

Fast

Flexible architecture

Testable components

Angular

Components

First-class TypeScript support

Full offering - officially supported router and state management, among others



Installing

- CDN
- NPM
- Bower
- CLI



Ecosystem

Ajax github.com/vuejs/vue-resource || github.com/mzabriskie/axios

Routing github.com/vuejs/vue-router

Flux-like State github.com/vuejs/vuex

CLI Generator github.com/vuejs/vue-cli

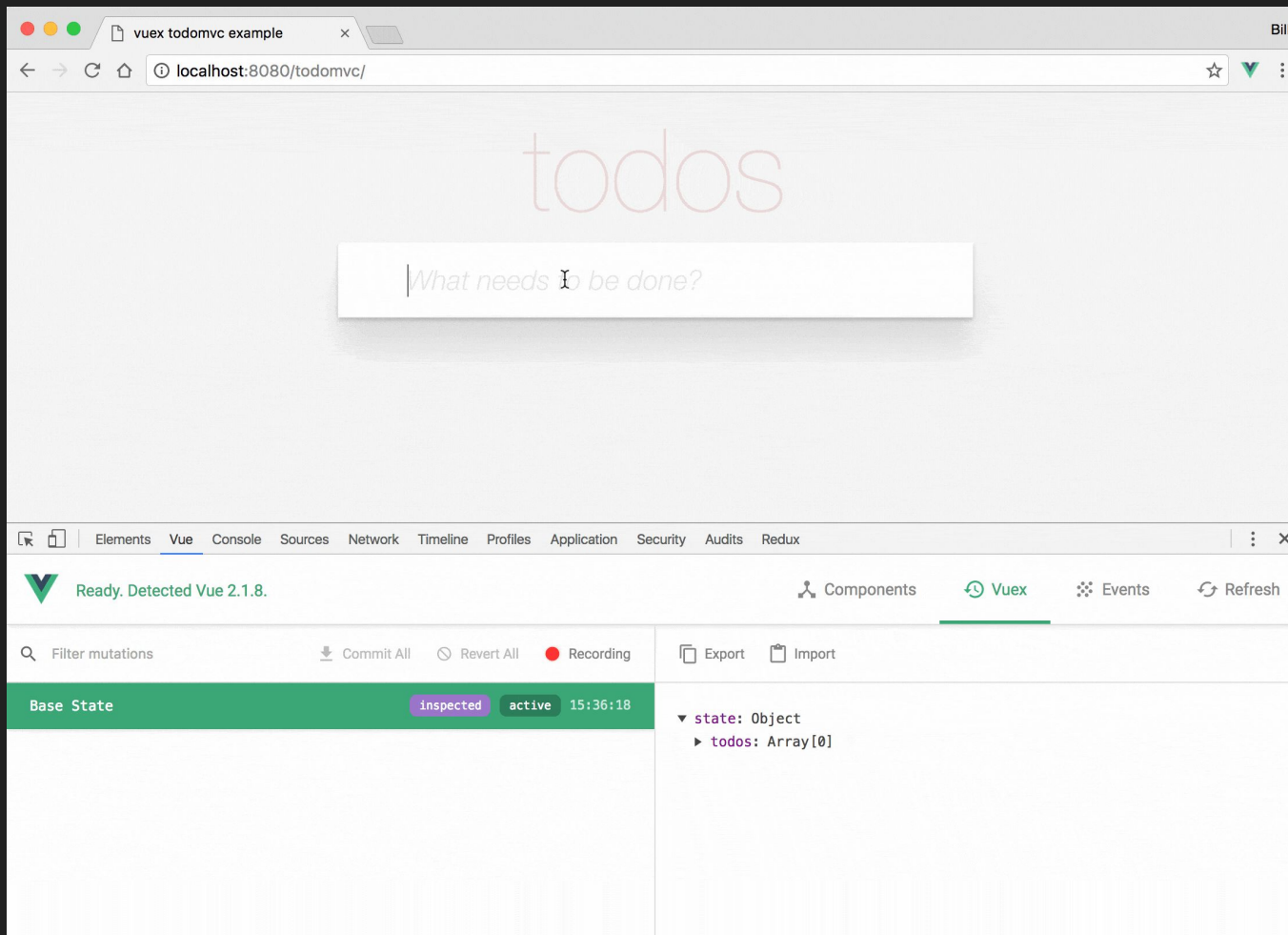
Linting github.com/vuejs/vue-validator

...

...



Devtools(chrome extension/firefox addon)





Basics

HTML

```
1 <script src="https://cdnjs.cloudflare.com/ajax/libs/vue/2.4.1/vue.js"></script>
2 <div id="app">
3   {{greeting}}
4 </div>
```

CSS

JS (Babel)

```
1 new Vue({
2   el: '#app',
3   data: {
4     greeting: 'Hello world!'
5   }
6 })
```

Hello world!

<https://codepen.io/Graniron/pen/JJzwEB?editors=1010>



Directives

- v-text
- v-html
- v-show
- v-if
- v-else
- v-else-if
- v-for
- v-on
- v-bind
- v-model
- v-pre
- v-cloak
- v-once



v-if

```
HTML
1 <script src="https://cdnjs.cloudflare.com/ajax/libs/vue/2.4.1/vue.js">
  </script>
2
3 <div id="app">
4   <span v-if="isVisible">
5     Now you can see mee
6   </span>
7   <br>
8   <button v-on:click="isVisible = !isVisible">Toggle visibility</button>
9 </div>

CSS

JS
1 new Vue({
2   el: '#app',
3   data: {
4     isVisible: true
5   }
6 })
```

Toggle visibility

<https://codepen.io/Graniron/pen/jwjygR?editors=1010>



v-for

HTML

```
1 <script src="https://cdnjs.cloudflare.com/ajax/libs/vue/2.4.1/vue.js"></script>
2
3 <div id="app">
4   <ul>
5     <li v-for="item in items">{{item}}</li>
6   </ul>
7 </div>
```

JS

```
1 new Vue({
2   el: '#app',
3   data: {
4     items: ['apple', 'berry', 'cucumber']
5   }
6 })
```

- apple
- berry
- cucumber

<https://codepen.io/Graniron/pen/pwXebo?editors=1010>



v-model

```
HTML
1 <script
  src="https://cdnjs.cloudflare.com/ajax/libs/vue/2.4.1/vue.
  js"></script>
2
3 <div id="app">
4
5   <h1>{{name}}</h1>
6   <input type="text" v-model="name">
7
8 </div>

CSS

JS
1 new Vue({
2   el: '#app',
3   data: {
4     name: 'Roman'
5   }
6 })
```

Roma

<https://codepen.io/Graniron/pen/BZgWvX?editors=1010>



v-bind/v-on

HTML

```
1 <script src="https://cdnjs.cloudflare.com/ajax/libs/vue/2.4.1/vue.js"></script>
2
3 <div id="app">
4
5   <button v-on:click="name=''>Reset name</button>
6   <button @click="name=''>Reset name</button><!-- using shorthand -->
7
8   <input type="text" v-bind:value="name">
9   <input type="text" :value="name"><!-- using shorthand -->
10
11 </div>
```

JS

```
1 new Vue({
2   el: '#app',
3   data: {
4     name: 'Roman'
5   }
6 })
```

Reset name

Reset name

Roman

Roman

<https://codepen.io/Graniron/pen/vZqygj?editors=1010>



Event modifiers

- .stop
- .prevent
- .capture
- .self
- .once

```
1 <!-- the click event's propagation will be stopped -->
2 <a v-on:click.stop="doThis"></a>
3
4 <!-- the submit event will no longer reload the page -->
5 <form v-on:submit.prevent="onSubmit"></form>
6
7 <!-- modifiers can be chained -->
8 <a v-on:click.stop.prevent="doThat"></a>
9
10 <!-- just the modifier -->
11 <form v-on:submit.prevent></form>
12
13 <!-- use capture mode when adding the event listener -->
14 <!-- i.e. an event targeting an inner element is handled here before being
    handled by that element -->
15 <div v-on:click.capture="doThis">...</div>
16
17 <!-- only trigger handler if event.target is the element itself -->
18 <!-- i.e. not from a child element -->
19 <div v-on:click.self="doThat">...</div>
```




Keys modifiers

- .enter
- .tab
- .delete (captures both “Delete” and “Backspace” keys)
- .esc
- .space
- .up
- .down
- .left
- .right

- .ctrl
- .alt
- .shift
- .meta

Mouse modifiers

- .left
- .right
- .middle

HTML

```
1 <!-- Alt + C -->
2 <input @keyup.alt.67="clear">
3
4 <!-- only call vm.submit() when the keyCode is 13 -->
5 <input @keyup.13="submit">
6
7 <!-- the same as previous -->
8 <input @keyup.enter="submit">
```



Vue instance properties

HTML

```
1 <div id="app">
2   <span>{{fullName}}</span>
3   <input type="text" v-model="firstName">
4   <button @click="onSearch">Search</button>
5 </div>
```

CSS

JS

```
1 new Vue({
2   el: '#app',
3   data: { firstName: 'Roman', lastName: 'Yavoriv' },
4   methods: {
5     onSearch(q) {
6       // perform search
7     }
8   },
9   computed: {
10    fullName() {
11      return this.firstName + ' ' + this.lastName;
12    }
13  },
14  watch: {
15    firstName(newValue) {
16      this.onSearch(newValue);
17    }
18  }
19 })
```

<https://vuejs.org/v2/api/#Options-Data>



```

graph TD
    Start([new Vue()]) --> BeforeCreate([beforeCreate()])
    BeforeCreate --> Init[Initialize Data & Events]
    Init -.-> Created([Instance Created])
    Created --> CreatedEvent([created()])
    CreatedEvent --> Compile[Compile template or el's template]
    Compile -.-> BeforeMount([beforeMount()])
    BeforeMount --> Replace[Replace el with compiled template]
    Replace -.-> Mounted([Mounted to DOM])
    Mounted -.-> DataChanged([Data Changed])
    DataChanged -.-> BeforeUpdate([beforeUpdate()])
    BeforeUpdate -.-> ReRender[Re-render DOM]
    ReRender -.-> Updated([updated()])
    Updated -.-> BeforeDestroy([beforeDestroy()])
    BeforeDestroy --> Destroyed([Destroyed])
    Destroyed --> DestroyedEvent([destroyed()])
    DestroyedEvent --> Start
  
```



Example of using hooks

⚙ JS

```
1 new Vue({
2   el: '#app',
3   data: { firstName: 'Roman', lastName: 'Yavoriv' },
4   created() {
5     // get user by http, ect.
6   },
7   beforeDestroy() {
8     // clear some data, ect.
9   }
10 })
```

<https://vuejs.org/v2/api/#Options-Lifecycle-Hooks>



Components

⚙ HTML

```
1 <div id="app">
2   <app-header></app-header>
3 </div>
```

▼ CSS

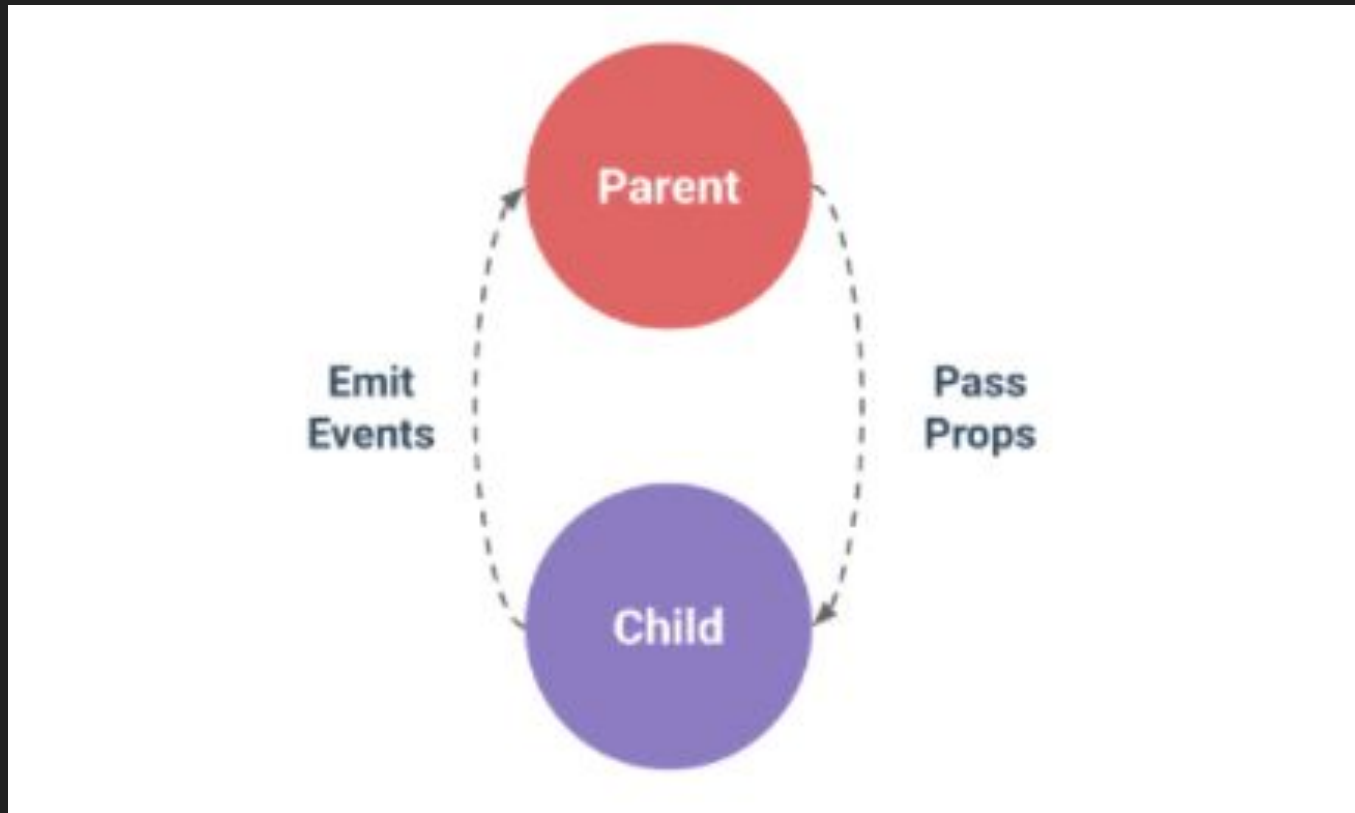
⚙ JS

```
1 Vue.component('appHeader', {
2   template: `<div>Navbar</div>`
3 })
4
5 new Vue({
6   el: '#app',
7 })
```

Navbar



Components communication





Component communication example

HTML

```
1 <div id="app">
2   <app-header :username="username"
3     @change="username='Robert Jr.'"></app-header>
4 </div>
5
6
```

CSS

JS

```
1 Vue.component('appHeader', {
2   props: ['username'],
3   template: `
4     <div>
5       {{username}}
6       <button @click="$emit('change')">Change user</button>
7     </div>`
8 })
9
10 new Vue({
11   el: '#app',
12   data: {
13     username: 'John Doe'
14   }
15 })
```

Robert Jr.



Fast start with CLI

- `$ npm install -g vue-cli`
- `$ vue init <template-name> <project-name>`

Choose from different Templates

simple

index.html + Vue CDN
import

webpack-simple

Basic Webpack
Workflow

webpack

Complex Webpack
Workflow (incl.
Testing)

browserify /
browserify-simple

Browserify Workflows



Single file component

```
1  <template>
2    <div>
3        <h1>{{title}}</h1>
4    </div>
5  </template>
6
7  <script>
8      export default {
9          data() {
10             return {
11                 title: 'Example'
12             }
13         }
14   }
15 </script>
16
17 <style>
18     h1 {
19         color: tomato
20     }
21 </style>
```



What About Separation of Concerns?

```
1  <template>
2    <div id="app">
3      <app-header></app-header>
4      <app-footer></app-footer>
5    </div>
6  </template>
7
8  <script src="./components/Home.vue"></script>
9  <style src="./mq.css"></style>
```



Declare components locally

```
<template>
  <div id="app">
    <app-header></app-header>
    <app-footer></app-footer>
  </div>
</template>

<script>
import Header from './components/Header'
import Footer from './components/Footer'
export default {
  el: '#app',
  components: { appHeader: Header, appFooter: Footer }
}
</script>
```



Filters

```
18 //global
19 Vue.filter('filterName', function(value) {
20     return // thing to transform
21 });
22
23 //locally, like methods or computed
24 filters: {
25     filterName(value) {
26         return // thing to transform
27     }
28 }
```



There are a lot more in Vue

- custom directives,
- mixins,
- slots,
- events bus,
- animation,
- dynamic components
- vue-router, router guards
- vue-resource, interceptors
- vuex
- server side rendering(nuxt.js)
- ...



Cons

- Relatively small community
- Is not backed by giant company
- Relatively small amount of resources



Resources

<https://vuejs.org>

<https://github.com/vuejs/awesome-vue>

<https://github.com/vuejs/vue-cli>

<https://css-tricks.com/guides/vue/>

<https://www.udemy.com/vuejs-2-the-complete-guide/>

<https://github.com/vuejs/vue-router>

<https://github.com/vuejs/vuex>

<https://nuxtjs.org/>

<https://vuejs-tips.github.io/cheatsheet/>

<https://chrome.google.com/webstore/detail/vuejs-devtools/nhdogjmejiglipccpnnnanhbledajbpd>

// Simple book CRUD made with Vue.js

<https://github.com/Graniron/tm-vue-library>





Contacts

 graniron

 roman.yavoriv@techmagic.co

 graniron