Vue.js

Vue.js (commonly referred to as **Vue**; pronounced /vjuː/, like "view"[4]) is an open-source Model–view–viewmodel JavaScript framework for building user interfaces and single-page applications.[11] It was created by Evan You, and is maintained by him and the rest of the active core team members coming from various companies such as Netlify and Netguru[12].

Vue.js

Original author(s)	Evan You
Initial release	February 2014[1]
Stable release	2.6.11 / December 13, 2019[2]
Repository	Vue.js Repository (https://github.com/vuejs/vue)
Written in	JavaScript
Size	33.30KB min+gzip
Type	JavaScript framework
License	MIT License[3]
Website	vuejs.org (https://vuejs.org)

Contents

Overview

Vue.js features an incrementally adoptable architecture that focuses on declarative rendering and component composition. Advanced features required for complex applications such as routing, state management and build tooling are offered via officially maintained supporting libraries and packages[13], with Nuxt.js as one of the most popular solutions.

Vue.js lets you extend HTML with HTML attributes called directives[14]. The directives offer functionality to HTML applications, and come as either built-in or user defined[15] directives.

History

Vue was created by Evan You after working for Google using AngularJS in a number of projects. He later summed up his thought process: "I figured, what if I could just extract the part that I really liked about Angular and build something really lightweight."[16] The first source code commit to the project was dated July 2013, and Vue was first released the following February, in 2014.

Versions

Version	Release date	Title
2.6	2019 Feb 4	Macross
2.5	2017 Oct 13	Level E
2.4	2017 Jul 13	Kill la Kill
2.3	2017 Apr 27	JoJo's Bizarre Adventure
2.2	2017 Feb 26	Initial D
2.1	2016 Nov 22	Hunter X Hunter
2.0	2016 Sep 30	Ghost in the Shell
1.0	2015 Oct 27	Evangelion
0.12	2015 Jun 12	Dragon Ball
0.11	2014 Nov 7	Cowboy Bebop
0.10	2014 Mar 23	Blade Runner
0.9	2014 Feb 25	Animatrix
0.8	2014 Jan 27	N/A
0.7	2013 Dec 24	N/A
0.6	2013 Dec 8	N/A

Features

Components

Vue components extend basic HTML elements to encapsulate reusable code. At a high level, components are custom elements to which the Vue's compiler attaches behavior. In Vue, a component is essentially a Vue instance with pre-defined options.[17] The code snippet below contains an example of a Vue component. The component presents a button and prints the number of times the button is clicked:

```
<div id="tuto">
    <button-clicked v-bind:initial-count="0"></button-clicked>
</div>

<script>
Vue.component('button-clicked', {
  props: [ "initialCount" ],
  data: () => ({
    count: 0,
  }),
  template: `<button v-on:click="onClick">Clicked {{ count }} times</button>`,
  computed: {
    countTimesTwo() {
      return this.count * 2;
    }
  },
  watch: {
```

```
    count(newValue, oldValue) {
      console.log(`The value of count is changed from ${oldValue} to ${newValue}.`);
    }
  },
  methods: {
    onClick() {
        this.count += 1;
    }
  },
  mounted() {
    this.count = this.initialCount;
  }
});

new Vue({
  el: '#tuto',
});
</script>
```

Templates

Vue uses an HTML-based template syntax that allows binding the rendered DOM to the underlying Vue instance's data. All Vue templates are valid HTML that can be parsed by specification-compliant browsers and HTML parsers. Vue compiles the templates into virtual DOM render functions. A virtual Document Object Model (or "DOM") allows Vue to render components in its memory before updating the browser. Combined with the reactivity system, Vue is able to calculate the minimal number of components to re-render and apply the minimal amount of DOM manipulations when the app state changes.

Vue users can use template syntax or choose to directly write render functions using JSX.[18] Render functions allow application to be built from software components.[19]

Reactivity

Vue features a reactivity system that uses plain JavaScript objects and optimized re-rendering. Each component keeps track of its reactive dependencies during its render, so the system knows precisely when to re-render, and which components to re-render.[20]

Transitions

Vue provides a variety of ways to apply transition effects when items are inserted, updated, or removed from the DOM. This includes tools to:

- Automatically apply classes for CSS transitions and animations
- Integrate third-party CSS animation libraries, such as Animate.css
- Use JavaScript to directly manipulate the DOM during transition hooks
- Integrate third-party JavaScript animation libraries, such as Velocity.js

When an element wrapped in a transition component is inserted or removed, this is what happens:

1. Vue will automatically sniff whether the target element has CSS transitions or animations applied. If it does, CSS transition classes will be added/removed at appropriate timings.
2. If the transition component provided JavaScript hooks, these hooks will be called at appropriate timings.
3. If no CSS transitions/animations are detected and no JavaScript hooks are provided, the DOM operations for insertion and/or removal will be executed immediately on next frame.[21][22]

Routing

A traditional disadvantage of single-page applications (SPAs) is the inability to share links to the exact "sub" page within a specific web page. Because SPAs serve their users only one URL-based response from the server (it typically serves index.html or index.vue), bookmarking certain screens or sharing links to specific sections is normally difficult if not impossible. To solve this problem, many client-side routers delimit their dynamic URLs with a "hashbang" (#!), e.g. *page.com/#!/*. However, with HTML5 most modern browsers support routing without hashbangs.

Vue provides an interface to change what is displayed on the page based on the current URL path -- regardless of how it was changed (whether by emailed link, refresh, or in-page links). Additionally, using a front-end router allows for the intentional transition of the browser path when certain browser events (i.e. clicks) occur on buttons or links. Vue itself doesn't come with front-end hashed routing. But the open source "vue-router" package provides an API to update the application's URL, supports the back button (navigating history), and email password resets or email verification links with authentication URL parameters. It supports mapping nested routes to nested components and offers fine-grained transition control. With Vue, developers are already composing applications with small building blocks building larger components. With vue-router added to the mix, components must merely be mapped to the routes they belong to, and parent/root routes must indicate where children should render.[23]

```
<div id="app">
  <router-view></router-view>
</div>
...

<script>
...
const User = {
  template: '<div>User {{ $route.params.id }}</div>'
}

const router = new VueRouter({
  routes: [
    { path: '/user/:id', component: User }
  ]
})
...
</script>
```

The code above:

1. Sets a front-end route at `websitename.com/user/<id>`.
2. Which will render in the User component defined in (const User...)
3. Allows the User component to pass in the particular id of the user which was typed into the URL using the $route object's params key: `$route.params.id`.
4. This template (varying by the params passed into the router) will be rendered into `<router-view></router-view>` inside the DOM's div#app.
5. The finally generated HTML for someone typing in: `websitename.com/user/1` will be:

```
<div id="app">
  <div>
    <div>User 1</div>
  </div>
</div>
```

[24]

Ecosystem

The core library comes with tools and libraries both developed by the core team and contributors.

Official Tooling

- **Devtools** - Browser devtools extension for debugging Vue.js applications
- **Vue CLI** - Standard Tooling for rapid Vue.js development
- **Vue Loader** - a webpack loader that allows the writing of Vue components in a format called Single-File Components (SFCs)

Official Libraries

- **Vue Router** - The official router for Vue.js
- **Vuex** - Flux-inspired Centralized State Management for Vue.js
- **Vue Server Renderer** - Server-Side Rendering for Vue.js

Vue.js Conferences

The first Vue.js Conference in the world was organized in 2017 by a polish software development company, Monterail, and the author of Vue.js — Evan You [25]. The conference took place on June 21-23, 2017 in New Horizons Cinema in Wrocław, Poland. It was a huge success and attracted over 300 attendees from around the world.

The speakers included:

- Evan You — Creator of Vue.js
- Sarah Drasner — Co-founder of Web Animation Workshops
- Sean Larkin — webpack Maintainer
- Blake Newman — Software Engineer at Attest
- Alexandre Chopin — Nuxt.js Co-author
- Sébastien Chopin — Nuxt.js Co-author
- Roman Kuba - Lead Frontend Scientist at Codeship
- Eduardo San Martin Morote — Vue.js Core Team Member
- Masahiro Tanaka — Founder & CEO, Monaca
- Callum Macrae — Front-End Developer at SamKnows
- Guillaume Chau — Vue.js Core Team Member
- Eric Baer — Lead Software Engineer at Formidable
- Jacob Lee — Software Engineer at StdLib
- Chris Fritz — Consultant, Vue.js Core Team Member
- Pine Wu — Author of Vetur
- Filipa Lacerda — Frontend Engineer at Gitlab[1] (https://en.wikipedia.org/wiki/GitLab)
- Paweł Grabarz - Frontend Developer at Monterail

So far, in 2018 and 2019 Vue.js Conferences were organized in:

- Toronto, Canada[26]
- Berlin, Germany[27]
- London, United Kingdom[28]
- Tokyo, Japan[29]
- Shanghai, China[30]
- Alicante, Spain[31]
- Bangalore, India[32]
- Venice, Italy[33]
- Florida, USA[34]
- Amsterdam, Netherlands[35]
- Sao Paulo, Brasil[36]
- Atlanta, Georgia[37]
- Park City, UT[38]

See also

- Comparison of JavaScript frameworks
- React
- AngularJS
- Angular
- JavaScript framework
- JavaScript library
- Model–view–ViewModel

Sources

This article incorporates text from a free content work. Licensed under MIT License License statement (https://github.com/vuejs/vuejs.org/blob/fdc66945e18020cdbdc729fd7ab8005d123216a7/LICENSE): *Vue.js Guide* (https://vuejs.org/v2/guide), Vue.js,

References

1. "First Week of Launching Vue.js" (http://blog.evanyou.me/2014/02/11/first-week-of-launching-an-oss-project/). *Evan You*.
2. "Vue.js Releases" (https://github.com/vuejs/vue/releases). *GitHub*. February 5, 2020.
3. "vue/LICENSE" (https://github.com/vuejs/vue/blob/dev/LICENSE). *GitHub*. Retrieved April 17, 2017.
4. "Guide: What is Vue.js?" (https://vuejs.org/v2/guide/index.html#What-is-Vue-js). *Vue.js*. Retrieved January 3, 2020.
5. Macrae, Callum (2018). *Vue.js: Up and Running: Building Accessible and Performant Web Apps* (https://books.google.com/books?id=bJpNDwAAQBAJ). O'Reilly Media. ISBN 9781491997215.
6. Nelson, Brett (2018). *Getting to Know Vue.js: Learn to Build Single Page Applications in Vue from Scratch* (https://books.google.com/books?id=mTpsDwAAQBAJ). Apress. ISBN 9781484237816.

7. Yerburgh, Edd (2019). *Testing Vue.js Applications* (https://books.google.com/books?id=7-FAtAEACAAJ). Manning Publications. ISBN 9781617295249.
8. Freeman, Adam (2018). *Pro Vue.js 2* (https://books.google.com/books?id=HQFuDwAAQBAJ). Apress. ISBN 9781484238059.
9. Franklin, Jack; Wanyoike, Michael; Bouchefra, Ahmed; Silas, Kingsley; Campbell, Chad A.; Jacques, Nilson; Omole, Olayinka; Mulders, Michiel (2019). *Working with Vue.js* (https://books.google.com/books?id=OKScDwAAQBAJ). SitePoint. ISBN 9781492071440.
10. "Introduction — Vue.js" (https://vuejs.org/v2/guide/#What-is-Vue-js). Retrieved March 11, 2017.
11. [5][6][7][8][9][10]
12. "Meet the Team — Vue.js" (https://vuejs.org/v2/guide/team.html). *vuejs.org*. Retrieved September 23, 2019.
13. "Evan is creating Vue.js | Patreon" (https://www.patreon.com/evanyou). *Patreon*. Retrieved March 11, 2017.
14. "What is Vue.js" (https://www.w3schools.com/whatis/whatis_vue.asp). *www.w3schools.com*. Retrieved January 21, 2020.
15. "User-defined definition and meaning | Collins English Dictionary" (https://www.collinsdictionary.com/dictionary/english/user-defined). *www.collinsdictionary.com*. Retrieved January 21, 2020.
16. "Between the Wires | Evan You" (https://web.archive.org/web/20170603052649/https://betweenthewires.org/2016/11/03/evan-you/). *Between the Wires*. November 3, 2016. Archived from the original (https://betweenthewires.org/2016/11/03/evan-you/) on June 3, 2017. Retrieved August 26, 2017.
17. "Components — Vue.js" (https://vuejs.org/v2/guide/components.html). Retrieved March 11, 2017.
18. "Template Syntax — Vue.js" (https://vuejs.org/v2/guide/syntax.html). Retrieved March 11, 2017.
19. "Vue 2.0 is Here!" (https://medium.com/the-vue-point/vue-2-0-is-here-ef1f26acf4b8#77d9). *The Vue Point*. September 30, 2016. Retrieved March 11, 2017.
20. "Reactivity in Depth — Vue.js" (https://vuejs.org/v2/guide/reactivity.html). Retrieved March 11, 2017.
21. "Transition Effects — Vue.js" (https://vuejs.org/v2/guide/transitions.html). Retrieved March 11, 2017.
22. "Transitioning State — Vue.js" (https://vuejs.org/v2/guide/transitioning-state.html). Retrieved March 11, 2017.
23. "Routing — Vue.js" (https://vuejs.org/v2/guide/routing.html). Retrieved March 11, 2017.
24. You, Evan. "Vue Nested Routing (2)" (https://router.vuejs.org/en/essentials/nested-routes.html). *Vue Home Page (subpage)*. Retrieved May 10, 2017.
25. https://evanyou.me
26. https://vuetoronto.com
27. https://vuecamp.de
28. https://vuejs.london
29. https://vuefes.jp/2019/
30. https://vue.w3ctech.com
31. https://vueday.org
32. https://hasgeek.com/jsfoo/2019-vueday/
33. https://2019.vueday.it
34. https://vueconf.us
35. https://www.vuejs.amsterdam

36. https://vuejssummit.com
37. http://connect.tech
38. https://www.frameworksummit.com

External links

- Official website (https://vuejs.org/)