State of the Vuenion

VueConf US, Mar.2019
Tampa, Florida

Usage statistics from February 2019

- ~700k weekly active devtool users
- ~800k weekly downloads on NPM
- ~461M hits/month on jsDelivr CDN

Usage statistics from this morning

- ~780k weekly active devtool users
- ~1M weekly downloads on NPM
- ~500M hits/month on jsDelivr CDN

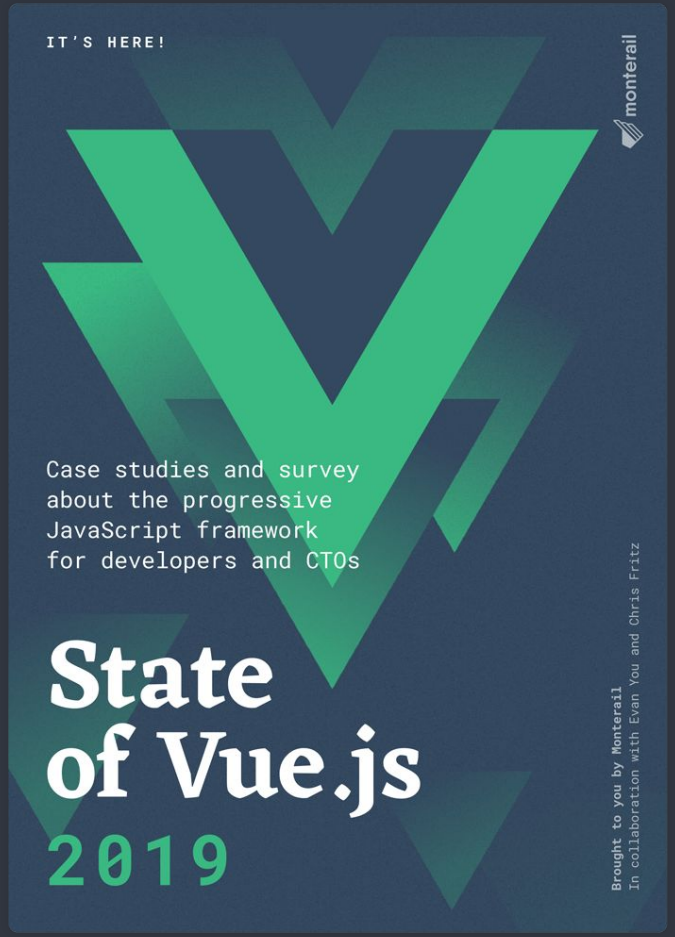
10% month-over-month growth

Now the 2nd most-starred project on GitHub! □

Updated State of Vue.js Report

https://www.monterail.com/state-of-vuejs-report

Case studies from Behance, GitLab, Clemenger BBDO, Livestorm, IBM, Fathom & Laravel



New Releases

Vue 2.6 “Macross”

- New slot syntax / performance improvements
- Improved async error handling
- Improved compiler error messages
- Built-in data prefetch support during server-side rendering

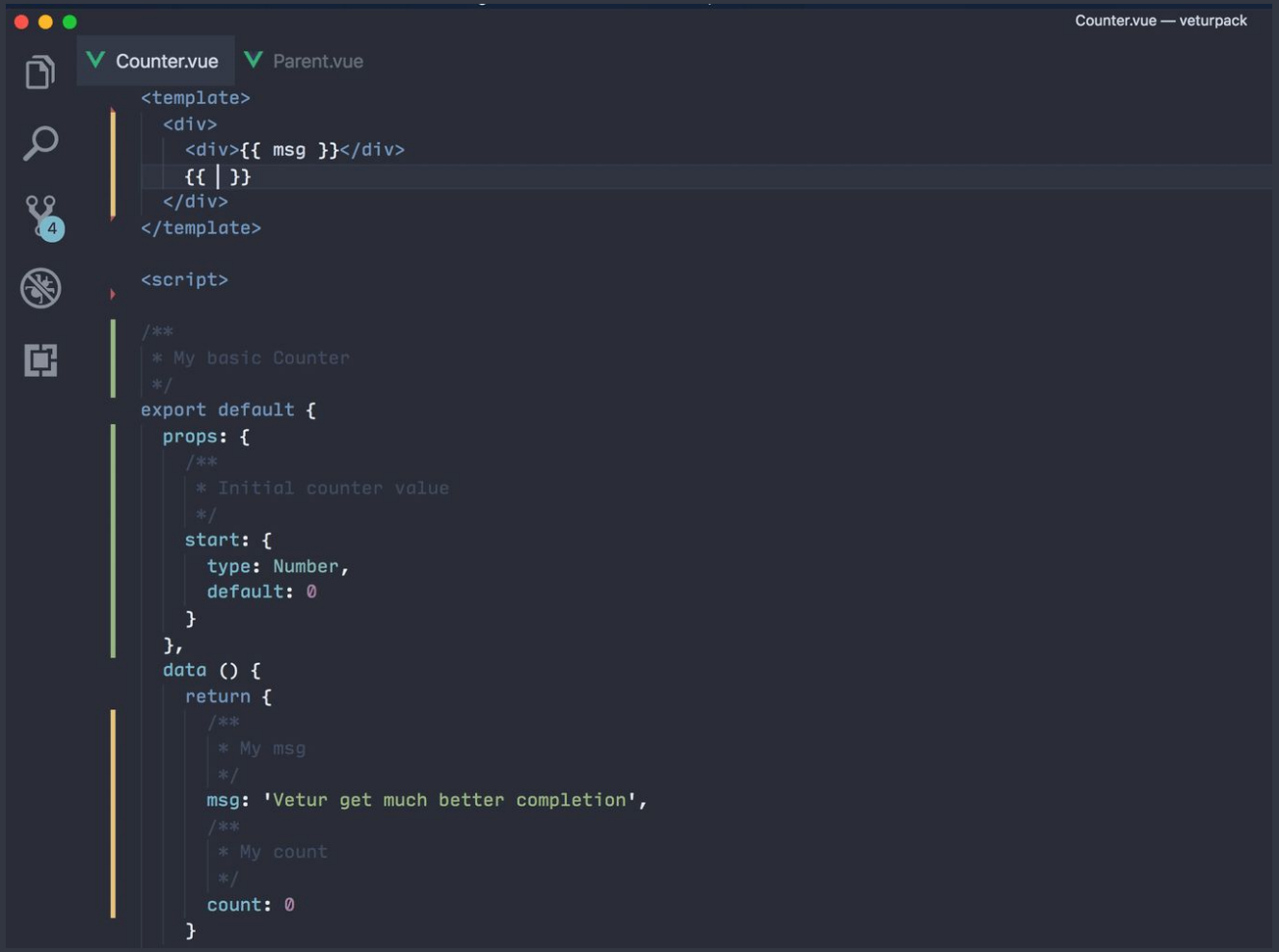
Vue Devtools 5.0

Release notes

- Routing tab
- Performance tab
- Settings tab
- Editable Vuex state
- Initial NativeScript support
- ...and more!
- Shoutout to @Akryum

Vetur 0.16

- Template intellisense support!
  - Expression autocomplete inside interpolations & directives
  - Child component tags & props
- Shoutout to @octref


Counter.vue — veturpack
Counter.vue
Parent.vue
4
<template>

{{ msg }}
{{ }}

</template>
<script>
/**
* My basic Counter
*/
export default {
props: {
/**
* Initial counter value
*/
start: {
type: Number,
default: 0
}
},
data () {
return {
/**
* My msg
*/
msg: 'Vetur get much better completion',
/**
* My count
*/
count: 0
}

Vue CLI 4.0 Roadmap

vuejs/vue-cli#3649

- Jest v24
- Workbox v4
- core-js v3
- Nightwatch v1

3.0 RFCs

New Slot Syntax

RFC-0001

- More succinct usage for default scoped slots

```
<foo v-slot="{ msg }">
  {{ msg }}
</foo>
```

New Slot Syntax

RFC-0001

- More consistent & explicit when using named slots

```
<foo>
  <template v-slot:one>
    foo
  </template>
  <template v-slot:scoped="{ msg }">
     {{ msg }}
  </template>
</foo>
```

New Slot Syntax

- Easier to associate slot scope declarations with the component providing the scope

Slots Unification:

All slots are implicitly scoped

vuejs/rfcs#20

Normal vs. Scoped Slots

```
createElement(
  Foo,
  // evaluated in parent render function
  // this.msg dependency registered by parent
  [createElement('div', this.msg)]
)
```

Normal vs. Scoped Slots

```
createElement(
  Foo,
  // lazy evaluated in child render function
  // this.msg dependency registered by child
  () => [createElement('div', this.msg)]
)
```

Class API

vuejs/rfcs#17

Class API

```
export default class App extends Vue {
 count = 0

 created() {
   console.log(this.count)
 }

 get plusOne() {
   return this.count + 1
 }

 increment() {
   this.count++
 }
}
```

Class API

- Primary goal: to provide a built-in & more efficient replacement for vue-class-component
- Works with both native ES2015 and TypeScript
- Object API still supported

Advanced Reactivity API

vuejs/rfcs#22

Advanced Reactivity API

```
import { state, computed, watch } from '@vue/observer'

const obj = state({ count: 1 })

const plusOne = computed(() => state.count + 1)

watch(plusOne, value => {
 console.log(`count + 1 is: ${value}`)
})
```

Advanced Reactivity API

- Create and observe reactive state outside of components
- Connect state into components by returning them in `data()`

Dynamic Lifecycle Hook Injection

vuejs/rfcs#23

Dynamic Lifecycle Injection

```
import { onMounted, onUnmounted } from 'vue'

export default {
 beforeCreate() {
   onMounted(() => {
     console.log('mounted')
   })

   onUnmounted(() => {
     console.log('unmounted')
   })
 }
}
```

Advanced Reactivity API

+

Dynamic Lifecycle Injection

=

New Composition Pattern

Case Study: Mouse Position

Mixins

```
const mousePositionMixin = {
  data() {
    return {
      x: 0,
      y: 0
    }
  },
  mounted() {
    window.addEventListener('mousemove', this.update)
  },
  destroyed() {
    window.removeEventListener('mousemove', this.update)
  },
  methods: {
    update(e) {
      this.x = e.pageX
      this.y = e.pageY
    }
  }
}
```

Mixins

- When overused:
  - ✕ Namespace clash (all options)
  - ✕ Unclear property source

Higher-order Components

```
const Demo = withMousePosition({
  props: ['x', 'y'],
  template: `<div>Mouse position: x {{ x }} / y {{ y }}</div>`
})
```

Higher-order Components

- When overused:
  - ✕ Namespace clash (props only)
  - ✕ Unclear props source
  - ✕ Extra component instances

Renderless Components

```
<mouse v-slot="{ x, y }">
  Mouse position: x {{ x }} / y {{ y }}
</mouse>
```

Renderless Components

- ✓ No namespace clashing
- ✓ Clear sources of variables
- ✕ Extra component instances

Hooks?

```
new Vue({
  template: `
    <div>
      Mouse position: x {{ mouse.x }} / y {{ mouse.y }}
    </div>
  `,
  data() {
    return {
      mouse: useMousePosition(this)
    }
  }
})
```

Hooks?

```
function useMousePosition(vm) {
  const mousePosition = Vue.observable({
    x: 0,
    y: 0
  })

  const update = e => {
    mousePosition.x = e.pageX
    mousePosition.y = e.pageY
  }

  vm.$on('hook:mounted', () => {
    window.addEventListener('mousemove', update)
  })

  vm.$on('hook:destroyed', () => {
    window.removeEventListener('mousemove', update)
  })

  return mousePosition
}
```

- ✓ No namespace clashing
- ✓ Clear sources of variables
- ✓ No extra component instances

With new API

```
function useMousePosition() {
  const x = value(0)
  const y = value(0)

  const update = e => {
    x.value = e.pageX
    y.value = e.pageY
  }

  onMounted(() => {
    window.addEventListener('mousemove', update)
  })

  onUnmounted(() => {
    window.removeEventListener('mousemove', update)
  })

  return { x, y }
}
```

With new API

```
new Vue({
  template: `
    <div>
      Mouse position: x {{ x }} / y {{ y }}
    </div>
  `,
  data() {
    const { x, y } = useMousePosition()
    return {
      x,
      y,
      // ... other data
    }
  }
})
```

- React-hooks like composability
- Recommended over mixins
- Only called once
  - Closer to standard JS intuition
  - No call-order constraint
  - No stale closures

More 3.0 RFCs to be published soon...

- Global API re-design
- Render function API change
- Functional and Async components API change
- Optional Props Declaration
- Attribute fallthrough behavior change

Thank you!