



Vue 3 > Vue 2 + 1

Mehr als ein Major Release



# Semver

Given a version number MAJOR.MINOR.PATCH, increment the

- MAJOR version when you make incompatible API changes,
- MINOR version when you add functionality in a backwards compatible manner, and
- PATCH version when you make backwards compatible bug fixes.

Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.





# Vue 3 ist ein "kleines" & großes Major Release

*klein*, nicht so groß in den API-Änderungen:

- Änderungen (breaking changes) sind zwar recht zahlreich...
- ...aber: Umfang und Komplexität überschaubar
- Wer eine Vue 2 App schreiben kann, kann mit Vue 3 direkt loslegen

**groß, unter der Haube:**

- neu geschrieben - in Typescript
- neue Features: Composition API<sup>\*</sup>
- Compiler und VirtualDOM Implementierung neu und grundverschieden
- ermöglicht erhebliche Performancezugewinne

**mehr als ein Major Release: ein KickStart**

<sup>\*</sup>: Nicht in diesem Talk inkludiert














---

# Ein großes Major Release

unter der Haube

# Alles auf Typescript

- Typescript ist überall
- Types können in IDEs auch bei JS code unterstützen
- Manuelle Pflege von Type in Vue 2
  - Mühsam
  - Fehleranfällig
  - unvollständig
- Types im Sourcecode machen Ihn für Contributors besser navigierbar

- >  .vscode
- ▼  packages
  - >  compiler-core
  - >  compiler-dom
  - >  compiler-sfc
  - >  compiler-ssr
  - >  reactivity
  - >  runtime-core
  - >  runtime-dom
  - >  runtime-test
  - >  server-renderer
  - >  sfc-playground
  - >  shared

# Neue Source-Struktur

- Monorepo
- Packages mit klaren APIs und Verantwortlichkeiten
- Einfacher für neue Contributors

## Custom Renderer

- core compiler DOM-unabhängig
- Community kann eigene Renderer bauen
- Bsp: Trois JS
- "VueNative" wäre nun möglich

## Demos

**Demo 1**

Demo 2

Demo 3

Demo 4

Demo 5

Demo 6

Demo 7

## Lights

## Materials

Textures



# Performance - Compiler & VirtualDOM

## Compiler

- Blocks - flache Arrays mit *dynamischen* Elementen
- Compiler Flags für *dynamische* Elemente
- Hoisting von `_statischen` Elementen

## Virtual DOM

- VNodes sind nun flache Objekte
- geringerer Memory Footprint
- schnelleres Diffing



# Ein Blick auf den Compiler

App.vue MyComponent.vue × +

```
1 <template>
2   <h1>Welcome!</h1>
3   <div class="static">
4     <MyComponent
5       prop1="static"
6       :subtitle="subtitle"
7     >
8     <span> {{ msg }}</span>
9   </MyComponent>
10 </div>
11 </template>
12
13 <script>
14   import { ref } from 'vue'
15   import MyComponent from './MyComponent.vue'
16
17   export default {
18     components: {
19       MyComponent,
20     },
21     setup() {
22       const msg = 'Hello World!'
```

PREVIEW

JS

CSS

SSR

# Welcome!

Hello World!



# Wo steht das Vue Framework?

## Offizielle Libraries

### Stable

- Vue 3.0 (18.09.2020)
- Vue CLI (Vue 3 Support) (18.09.2020)
- eslint-plugin-vue 7.0 (30.09.2020)
- vue-loader 16.0 (25.11.2020)
- Vue Router 4.0 (07.12.2020)
- Vuex 4.0 (02.02.2021)

### Release Candidate

- @vue/test-utils - 2.0.0-rc.6

### Beta

- vue-devtools 6.0.0-beta.11
- rollup-plugin-vue: 6.0.0-beta.11

- 
- Offizielles Ökosystem fast vollständig stable
  - Q1/Q2 2021 wichtige Stablisierungsphase
  - Vue 3 bekam 11 Patch releases und erstes Minor Release 3.1 (07.06.2021)

# Community Ökosystem

## Frameworks

- Ant Design Vue - stable
- Element-Plus - stable
- Quasar - Beta
- Vuetify - Alpha
- Nuxt - First beta announced
- ChakraUI/Vue under development

## Integrationen

- VSCode (Volar)
- JetBrains Webstorm / PHPStorm etc
- Storybook `v6.3`
- Cypress
- Vite

# Next -> Latest

Bisher:

```
npm i vue@next
```

Ab Q3-2021:

```
npm i vue
```

# Vue 3 Contributions

Vue 2

*seit 11.06.2016*

**1.036**

merged PRs

---

**399**

Contributors

Vue 3

*seit 03.01.2020*

**1.372**

merged PRs

---

**234**

Contributors

---

PS: Vue 2 wurde gerade 5 Jahre alt! 🎉

---

# Ein "kleines" Major Release

Für Developer

# Was hat sich *genau* geändert?

- Es gibt 33 Einträge im Migration Guide
- davon sind 3 aber neue Funktionen (Fragments, Suspense)
- Die meisten "Breaking Changes" sind "one-liner" Fixes.

## Was bedeutet das für Developer?

- Die Migration (fast) jedes Breaking Change ist *einfach*
- Aber die Migration *aller* Changes ist *zeitaufwändig*

# Globale APIs

```
import Vue from 'vue'
import Dialog from './Dialog.vue'
import ClickOutside from './clickOutside.js'
import PortalVue from 'portal-vue'

Vue.component('Dialog', Dialog)
Vue.directive('clickOutside', ClickOutside)
Vue.use(PortalVue)

new Vue(App).$mount('#app')
```

```
import { createApp } from 'vue'
import Dialog from './Dialog.vue'
import ClickOutside from './clickOutside.js'
import PortalVue from 'portal-vue'

const app = createApp(App)
app.component('Dialog', Dialog)
app.directive('clickOutside', ClickOutside)
app.use(PortalVue)

app.mount('#app')
```

- 
- Einmalige Änderungen
  - Verbesserung: Mehrere Apps auf einer Seite sind besser isoliert.



# Lifecycle Hooks

```
<template>
  <h1>{{ message }}</h1>
</template>
<script>
export default {
  data: () => ({
    message
  }),
  beforeDestroy() {
    console.log('Destruction imminent!')
  }
}
</script>
```

```
<template>
  <h1>{{ message }}</h1>
</template>
<script>
export default {
  data: () => ({
    message
  }),
  beforeUnmount() {
    console.log('Destruction imminent!')
  }
}
</script>
```

---

```
`@vue/compat`
```

Migration mit der Compat Build (Vue 3.1)

# Warum eine Migration Build?

- Migration großer Projekte nicht "an einem Wochenende"
- Dependencies sind noch nicht alle Vue 3 kompatibel
- Deprecation Warnings zeigen notwendige Änderungen auf.

# Workflow - Initiale Upgrades

1. Upgrade der notwendigen Dependencies
2. Compat Mode aktivieren
3. App starten
4. Compiler Warnings fixen (z.B. wegen Filters)
5. Auf Vuex ``4.0`` upgraden
6. Auf VueRouter ``3.0`` upgraden

Eure App läuft jetzt auf Vue 3 - mit Vue 2 Component Syntax! 🤯

# Workflows für die Migration

2 Varianten:

1. Migration per Feature
2. Migration per Component

# Migration - per Feature

- Ein Feature aussuchen - z.B. Lifecycle hooks
- in jeder Component migrieren

Von Vue 2:

```
export default {  
  beforeDestroy() {  
    this.doSomething()  
  }  
}
```

zu Vue 3:

```
export default {  
  beforeUnmount() {  
    this.doSomething()  
  }  
}
```

- Anschließend das compat feature global deaktivieren

```
// main.js  
import {  
  createApp,  
  configureCompat  
} from 'vue'  
import App from './App.vue'  
  
configureCompat({  
  OPTIONS_BEFORE_DESTROY: false,  
})  
  
createApp(App).mount('#app')
```

# Migration - per Component

- Alle Migrationen in einer Component vornehmen

```
export default {
  props: ['value']
  components: {
    Child: () => import('./Child.vue')
  },
  methods: {
    update(value) {
      this.$emit('input', value)
    },
  },
  beforeDestroy() {
    this.doSomething()
  }
}
```

```
export default {
  compatConfig: {
    MODE: 3,
  },
  props: ['modelValue']
  components: {
    Child: defineAsyncComponent(
      () => import('./Child.vue')
    )
  },
  methods: {
    update(value) {
      this.$emit('update:modelValue', value)
    },
  },
  beforeUnmount() {
    this.doSomething()
  }
}
```

# Tips

- Ihr werdet eine **MENGE** Warnings sehen - keine Panik!
  - Nur in development
  - Minimaler Perf-Overhead, wettgemacht durch Vue 3 Perf-Verbesserungen
- Migriert zuerst "find&replace" features global in allen Components
- Migriert danach einzelne Components
- Migriert zuletzt eure dependencies auf ihre Vue 3 versionen



# Migration Build docs



3-months Vue School for only \$49 \$75!  
**Limited Time Offer**

GET ACCESS



V

Docs ▾

API Reference

Ecosystem ▾

Support Vue ▾

Translations ▾

GitHub [↗](#)

Introduction

Migration Build

Overview

Intended Use Cases

Known Limitations

Expectations

Upgrade Workflow

Preparations

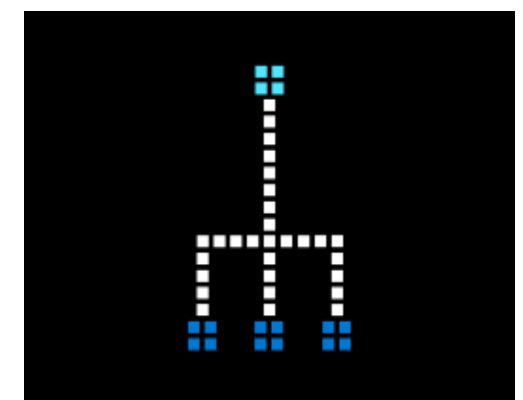
Installation

## Migration Build

### Overview

`@vue/compat` (aka "the migration build") is a build of Vue 3 that provides configurable Vue 2 compatible behavior.

The migration build runs in Vue 2 mode by default - most public APIs behave exactly like Vue 2, with only a few exceptions. Usage of features that have changed or been deprecated in Vue 2 will emit



Azure kostenfrei  
testen

ads via Carbon

---

# Mehr als ein Major Release

Wohin die Reise geht

# DX - IDE Integration

- Derzeit 3 Extensions: Vetur, VueDX, Volar
- Neues Fundament wird Volar sein
- Language Server Implementation auch für andere IDEs
- **Type Hints / Autocomplete in Templates!**

```
<template>
  <FileSelector/>
  <div class="editor-container">
    <CodeMirror @change="onChange" :value="activeCode" :mode="activeMode" />
    <Message :err="store.errors[0]" />
  </div>
</template>
```

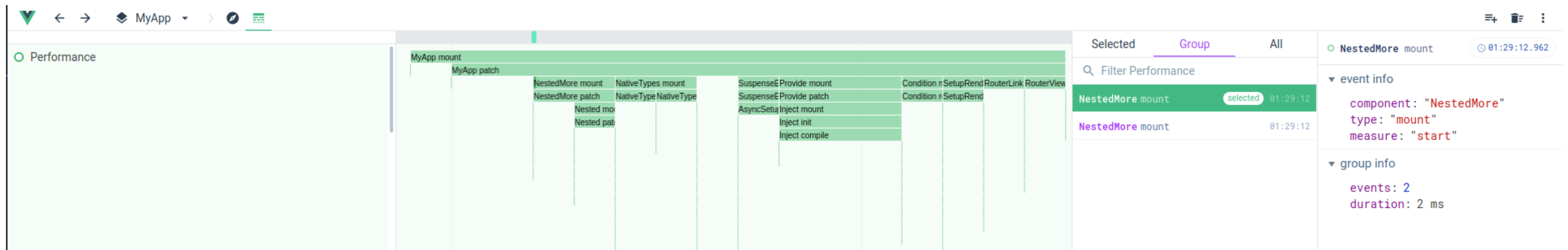
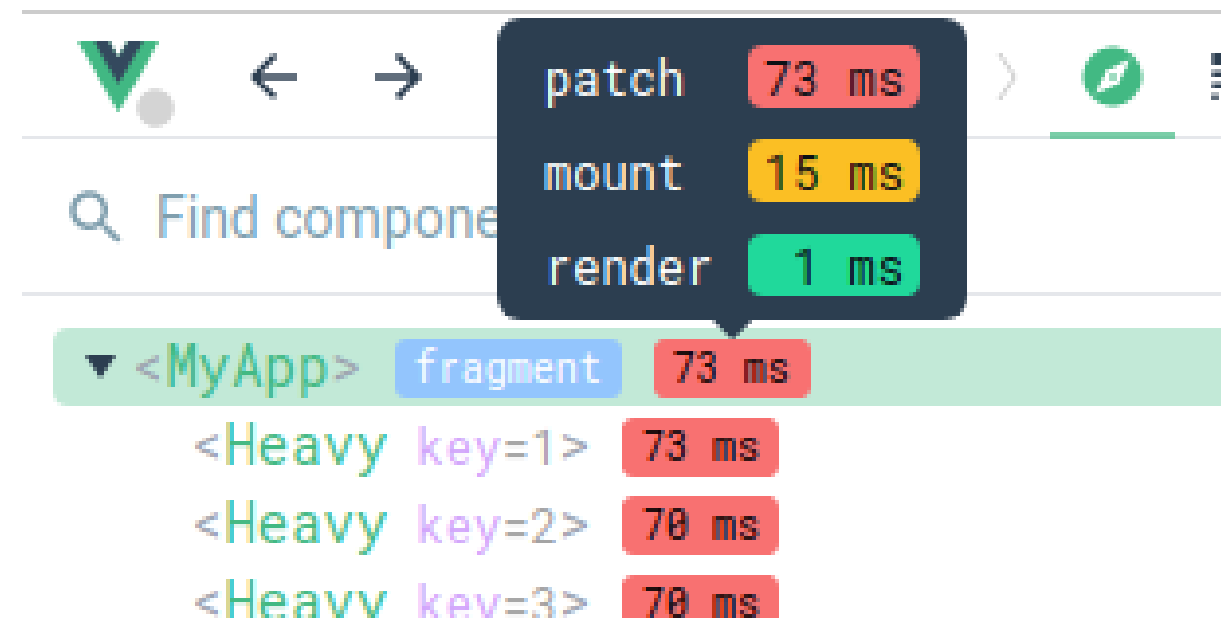
(property) value?: string | undefined

You, 2 months ago • workflow(sfc-playground): support multiple f

# DX - Devtools 6.0

- Neu: Performance Messung
- Neu: Events timeline
- Neu: Plugin API

# Performance Messung



# Plugin API

The screenshot shows the VitePressApp interface with the Pinia plugin API. The left sidebar lists categories: Mouse, Keyboard, Component events, Performance, and Pinia (selected). The main area displays a timeline of events for the Pinia plugin, including 'mutation a...' and 'fail end'. The right sidebar shows the details of the selected event, including 'event info' and 'group info'.

**Selected Group All**

Filter Pinia 🍌

fail start 16:00:11

mutation a... selected 16:00:12

fail ! end 16:00:13

**event info**

operation: add  
key: 0  
oldValue: undefined  
newValue: 127

**rawEvent(s):** Debugger  
effect: f reactive  
target: Array[6]  
key: "0"  
type: "add"  
newValue: 127  
oldValue: undefined  
oldTarget: undefined

**group info**

events: 3  
duration: 2005 ms

# RFCs - `<script setup>`

- Reduziert boilerplate
- Performance-Potentiale

[Link zum RFC](#)

```
<script setup>
import Foo from './Foo.vue'
import { ref, defineProps } from 'vue'

const props = defineProps({
  title: String,
})

const count = ref(0)
const inc = () => { count.value++ }
</script>

<template>
  <h1>{{ props.title }}</h1>
  <Foo :count="count" @click="inc" />
</template>
```

# Dieser Talk wurde mit Slidev gebaut



Slidev

[www.sli.dev](https://www.sli.dev)

- Vite
- Slides mit Markdown schreiben
- Vue 3 Components in Markdown
- Presenter Mode
- Recording im Browser
- VS Code Extension
- PDF Export
- Deployable als App

[vueday-2021.linusb.org](https://vueday-2021.linusb.org)