



まちにまった

Vue.js 3

Vue.js v-tokyo meetup #11

2020.08.25

@kazupon

自己紹介





kazupon

PLAID, inc.

Vue.js Core Team Member

Vue.js Japan User Group Organizer

Creator of Vue I18n & Intlify

WebAssembly Love ❤️



@kazu_pon



kazupon

はじめに

2020年7月18日

Vue.js 3.0 RC リリース！

Vue 3 is now in RC! #189

[Edit](#)[New issue](#)

yyx990803 opened this issue on Jul 18 · 94 comments



yyx990803 commented on Jul 18 · edited ▾

Member



Vue 3 is now in RC!

We are very excited to announce that Vue 3.0 has entered RC (Release Candidate) stage!

Entering the RC stage means that both the API and implementation of Vue 3 core have stabilized. In principle, we do not expect to introduce new major features or breaking changes before the final release. Most official framework parts also now have v3 support. Please check [here](#) for the latest status.

New Documentation

The Vue docs team have been updating our docs for v3 and it is now available at v3.vuejs.org! It has been a massive undertaking and many thanks to the hard work by the docs team: [@NataliaTepluhina](#), [@bencodezen](#), [@phanan](#) and [@sdras](#). The new docs has been meticulously migrated to cover differences between v2 and v3, runs on VuePress, and has improved code samples that can be edited inline.

For a quick overview on what's new and what's changed, please refer to the [Migration Guide](#).

Assignees



No one—assign yourself

Labels



announcement

Projects



None yet

Milestone



No milestone

Linked pull requests



Successfully merging a pull request may close this issue.

None yet


この issue で Evan 氏が今後について表明

<https://github.com/vuejs/rfcs/issues/189>

合わせて公式ドキュメントも公開 (β)

▲ Beta Version: Docs are in development and subject to change. Close


Vue.js Docs ▾ API Reference Ecosystem ▾ Support Vue ▾ GitHub ↗



The Progressive JavaScript Framework

[WHY VUE.JS?](#) [GET STARTED](#) [GITHUB](#)

Special Sponsor



Build app-to-app workflows and connect APIs

<https://v3.vuejs.org/>

公式プラグイン・ツールほぼ準備が整いつつある

- vue-router
v4.0.0-beta6
- vuex
v4.0.0-beta4
- vue-cli
v4.5.0 (vue 3 preset)
- vue-devtools
v6.0.0-beta.1
- eslint-plugin-vue
v7.0.0-beta.2
- @vue/test-utils
v2.0.0-beta.2
- vue-class-component
v8.0.0-alpha.6
- vue-loader
v16.0.0-beta.5
- rollup-plugin-vue
v6.0.0-beta.8

デシジョンツリーの公開



Vue 3: mid 2020 status update #183

yyx990803 opened this issue on Jul 1 · 54 comments

Decision Tree

It doesn't mean you cannot start using Vue 3 today though. Most of the framework parts are now in either beta or alpha, and the core itself has been extensively tested by our early adopters. The only thing that blocks us from going into RC is the browser devtools extension (which is being actively worked on at this moment). All the significant changes have been landed and documented in [RFCs](#) and there are no more planned breaking changes. If you've been waiting to get onboard with Vue 3, here is a decision tree to help you plan accordingly:

```
IWantVue3()

async function IWantVue3() {
  await read(`https://github.com/vuejs/rfcs/pulls?q=is%3Apr+is%3Amerged+label%3Acore+-label%3Arevoked+-label%3A2

  if (isTrue("I just want to play with Vue 3")) {
    // If you just want to try Vue 3 out - you can do it right now with Vite.
    // Vite (https://github.com/vitejs/vite) is a new dev/build tool that we
    // created that is lighter, faster and produces smaller bundles. It works
    // with Vue 3 out of the box.
    run(`npm init vite-app hello-vue3`)
    return
  }

  if (isTrue("I am planning to use Vue 3 for a new project")) {
    if (isTrue("I need IE11 support")) {
      await IE11CompatBuild() // July 2020
    }
    if (isTrue("RFCs are too dense, I need an easy-to-read guide")) {
      await migrationGuide() // July 2020
    }
  }
}
```

None yet

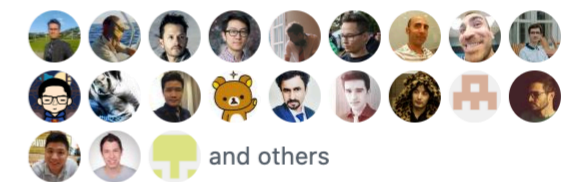
Notifications

Customize

Unsubscribe

You're receiving notifications because you're watching this repository.

54 participants



Lock conversation

Pin issue

→ Transfer issue

<https://github.com/vuejs/rfcs/issues/183>

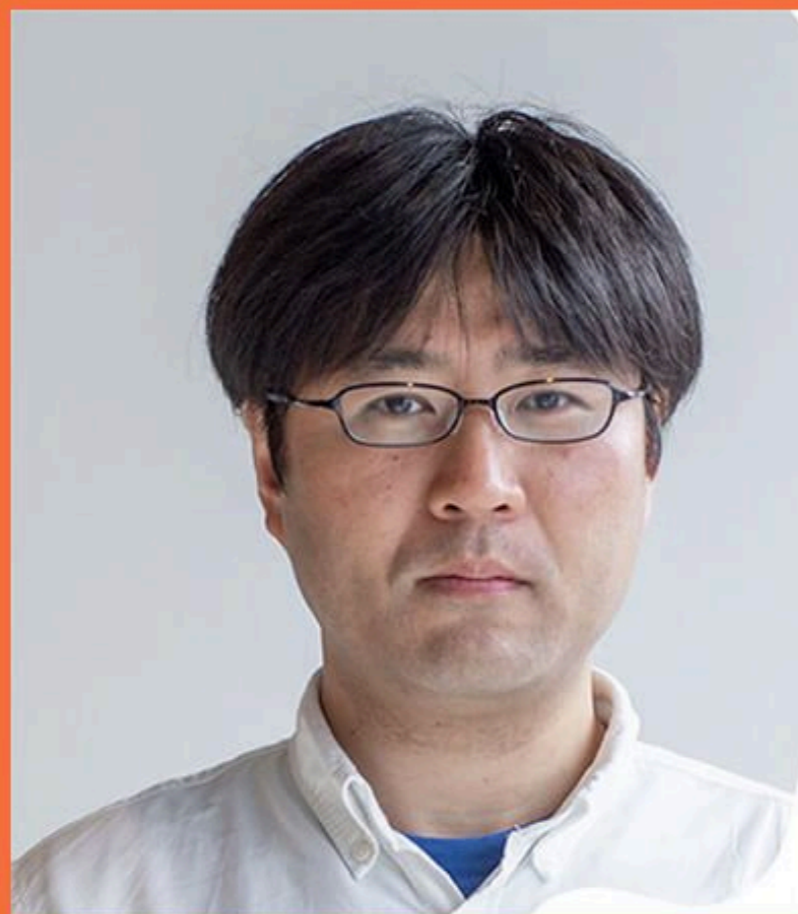
いよいよいよ

リリース間近！

今日話すこと

その前に

2020年2月1日



PLAID, inc.

川口 和也

PWA Night
CONFERENCE 2020

2020.2.1 Sat Abema Towers, Shibuya

Vue 3 について話した



スライドはこちら 🙌



The screenshot shows a presentation slide with a dark background. At the top left is the Speaker Deck logo (S), and at the top right is a search icon. The slide content includes the Vue.js logo (a green 'V' with a blue shadow), the title 'まもなくやってくる Vue.js 3' in large white text, and the subtitle 'PWA night conference 2020' and date '2020.02.01' in smaller white text. The presenter's name '@kazupon' is at the bottom. Navigation arrows are on the left, and share and full-screen icons are on the right. Below the slide, the user profile 'kazupon' and date 'February 01, 2020' are visible, along with a 'Programming' folder icon, a star icon with '35', an eye icon with '13k', and a download icon.

<https://speakerdeck.com/kazupon/mamonakuyatutekuru-vue-dot-js-3>

と、いいわけ

今日話すこと

- 2月の発表からの Diff (差分)
- Vue 3 関連周りのアップデート情報
 - SFC
 - Vite
 - @vuedx
 - その他

SFC

Evan 氏が SFC 改善 の PR を投下



 **Evan You**
@youyuxi

RFCs for improving the authoring experience of Vue Single File Components:

 SFC Improvements by yyx990803 · Pull Request #1...
This PR includes 3 RFCs related to improving the authoring experience of Vue SFCs (Single File ...
[github.com](#)

5:07 AM · Jun 30, 2020

489 likes 128 people are Tweeting about this

<https://twitter.com/youyuxi/status/1277695108268339208>

投下した RFC は 3つ

- 1. コンポーネントのインポート糖衣構文
- 2. Composition API の DX 改善
- 3. CSS 変数インジェクション

コンポーネント

の

インポート糖衣構文

<component>

- SFC に<component>ブロックを導入

```
<component src="../components/Foo.vue" />  
<component async src="../components/Bar.vue" />  
<component src="./components/Baz.vue" as="Qux" />
```

```
<template>  
  <Foo />  
  <Bar />  
  <Qux />  
</template>
```

<component>
で何が良くなるのか？

こういうコードを書く必要がなくなる！

- 一般的なコンポーネント

従来

RFC

```
<template>
  <Foo/>
</template>
```

```
<script>
import Foo from '../components/Foo.vue'

export default {
  components: {
    Foo
  }
}
</script>
```

```
<component src="../components/Foo.vue/>
```

```
<template>
  <Foo/>
</template>
```

コンポーネントを使うために
いちいち <script> ブロック内で
import する必要がなくなる

こういうコードを書く必要がなくなる！

- ・ 非同期なコンポーネント

従来

```
<template>
  <Foo/>
</template>

<script>
import { defineAsyncComponent } from 'vue'

export default {
  components: {
    Foo: defineAsyncComponent(
      () => import('../components/Foo.vue')
    )
  }
}
</script>
```

RFC

```
<component async src="../components/Foo.vue/>
```

```
<template>
  <Foo/>
</template>
```

非同期コンポーネントをロードするために
dynamic import で components オプションに
指定する必要がなくなる

コンポーネントを別名にできる

- `as` を使うことで別名にできる

```
<component src="../components/Foo.vue" as="Bar" />
```

```
<template>  
  <Bar />  
</template>
```

ES modules の `import` 構文の
`as` と同じことができる

Composition API

の

DX 改善

<script setup>

- <script> に setup が追加

```
<template>
  <button @click="inc">{{ count }}</button>
</template>

<script setup>
import { ref } from 'vue'

export const count = ref(0)
export const inc = () => count.value++
</script>
```

<script setup>

で

どうDXが良くなる？

より簡潔に実装できる！

従来

```
<template>
  <button @click="inc">{{ count }}</button>
</template>

<script>
import { ref } from 'vue'

export default {
  setup() {
    const count = ref(0)
    const inc = () => count.value++

    return {
      count,
      inc
    }
  }
}
</script>
```

RFC

```
<template>
  <button @click="inc">{{ count }}</button>
</template>

<script setup>
import { ref } from 'vue'

export const count = ref(0)
export const inc = () => count.value++
</script>
```

setup 関数内でロジックを実装して、レンダリングするためにコンテキスト返す必要がなくなる！

使い方

コンテキストでレンダリングするためには？

- export する

```
<template>  
  <button @click="inc">{{ count }}</button>  
</template>
```

```
<script setup>  
import { ref } from 'vue'
```

```
export const count = ref(0)  
export const inc = () => count.value++  
</script>
```

ES Modules の
export 構文を使うだけ

setup関数の引数はどう使う？

- setup の属性値に指定する

従来

RFC

```
<template>
  <p>msg: {{ msg }}</p>
</template>

<script>
import { watchEffect } from 'vue'

export default {
  props: {
    msg: String
  },
  setup(props, { emit }) {
    watchEffect(() => console.log(props.msg))
    emit('foo')
  }
}
</script>
```

```
<template>
  <p>msg: {{ msg }}</p>
</template>

<script setup="props, { emit }">
import { watchEffect } from 'vue'

export default {
  props: {
    msg: String
  }
}

watchEffect(() => console.log(props.msg))
emit('foo')
</script>
```

コンポーネントオプションは？

従来

```
<template>
  <p>msg: {{ msg }}</p>
</template>

<script>
import { watchEffect } from 'vue'

export default {
  props: {
    msg: String
  },
  setup(props, { emit }) {
    watchEffect(() => console.log(props.msg))
    emit('foo')
  }
}
</script>
```

props などの
コンポーネントオプションを
使う場合は
export default する必要がある

RFC

```
<template>
  <p>msg: {{ msg }}</p>
</template>

<script setup="props, { emit }">
import { watchEffect } from 'vue'

export default {
  props: {
    msg: String
  }
}

watchEffect(() => console.log(props.msg))
emit('foo')
</script>
```

TypeScript は？

- declare を使って型定義

```
<template>  
  <p>msg: {{ computedMsg }}</p>  
</template>
```

```
<script setup="props" lang="ts">  
import { computed } from 'vue'
```

```
declare const props: {  
  msg: string  
}
```

declare で props をすると、
SFCコンパイラが props 実際の定義コードと
TS の型推論されるようにコンパイルする

```
export const computedMsg = computed(() => props.msg + '!!!')  
</script>
```

<script> と <script setup> を一緒に使える

```
<script>
```

```
performGlobalSideEffect()
```

```
export const named = 1
```

```
</script>
```

```
<script setup>
```

```
import { ref } from 'vue'
```

```
export const count = ref(0)
```

```
</script>
```

用途:

- グローバルな処理を一度だけ実行したい場合
- SFC で named export なものを提供したい場合

CSS 変数

インジェクション

<style vars>

- style ブロックに vars が追加

```
<style vars="{ color }">
  .text {
    color: var(--color);
  }
</style>
```

<style vars>

で

どう良くなる？

script 側から値を渡せる！

```
<template>  
  <p class="text">hello</p>  
</template>
```

```
<script setup>  
import { ref } from 'vue'
```

```
export const color = ref('red')
```

```
</script>
```

バインディング！

```
<style vars="{ color }">
```

```
  .text {  
    color: var(--color);  
  }
```

```
</style>
```

vars に定義した変数を
Web標準のCSS変数の要領で
スタイル値として
割り当てることができる！
つまり、動的なスタイルを制御できる！

<style scoped>といっしょに利用可能

```
<template>
| <p class="text">hello!</p>
</template>

<script>
export default {
| data: () => ({ color: 'green' })
}
</script>

<style scoped vars="{ color }">
| .text {
| color: var(--color);
}
</style>
```

vars を scoped と
いっしょに利用したい場合は、
内部のCSS変数はローカルと見なす

つまり、このケースでは
--color が影響するのは
このコンポーネントにのみ

グローバルなCSS変数の利用は？

```
<template>
| <h1>hello!</h1>
</template>
```

```
<script>
export default {
| data: () => ({ color: 'green' })
}
</script>
```

```
<style scoped vars="{ color }">
h1 {
| color: var(--color);
| font-weight: var(--global:fontSize);
}
</style>
```

--global プリフィックスを使って
CSS変数を利用する

これらRFCの採択について

- `<script setup>`, `<style vars>` については RC に実装済み
- `<script setup>` は難なく使える
- `<style vars>` はCSS 変数が見えるモダンなブラウザで利用できる

これらRFCの採択について

- `<component>` についてはまだ未実装
- SFC に新たなブロックや書き方が加わるので、複雑性と学習コストが増すのでは？
- Vetur といったテンプレート解析をするツールへ影響が大きすぎるのでは？

というわけで、Vue 3.0 では...

- まだ RFC でいるいると議論、フィードバックがあるため SFC の正式機能ではない
- この SFC 改善は Experimental (実験的) な機能として提供される
- 問題なければ、v3.1? で採択される予定

Vite

Evan 氏が新しいビルドツール公開

- 読み方: ` /vit/`



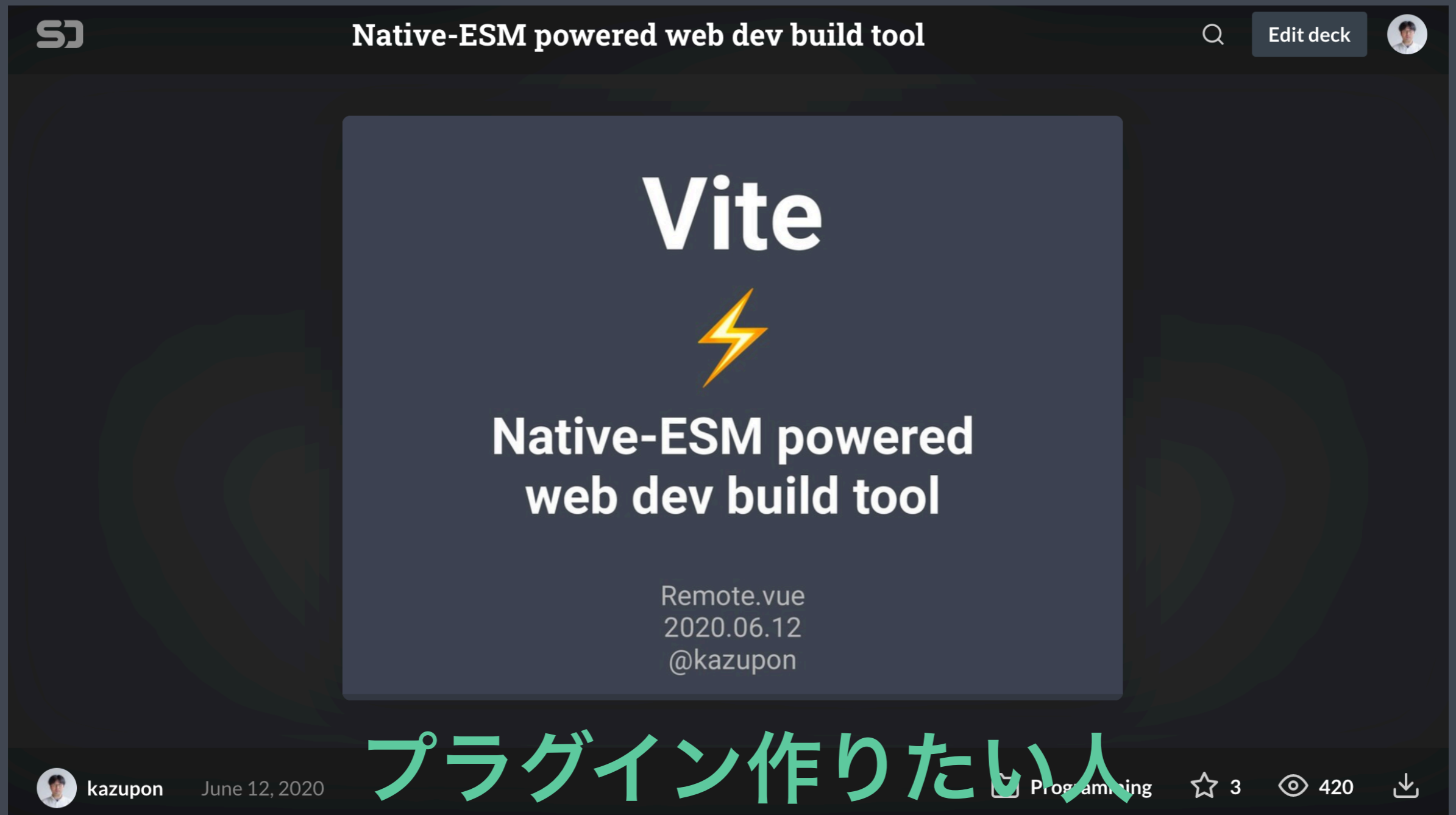
npm v1.0.0-rc.4 node >=10.16.0 circleci passing build passing

Vite is an opinionated web dev build tool that serves your code via native ES Module imports during dev and bundles it with [Rollup](#) for production.

- Lightning fast cold server start
- Instant hot module replacement (HMR)
- True on-demand compilation
- More details in [How and Why](#)

<https://github.com/vitejs/vite>

6月にViteについて話した



The screenshot shows a presentation slide for Vite. At the top left is the Speaker Deck logo (SD). The title of the slide is "Native-ESM powered web dev build tool". In the top right corner, there is a search icon, an "Edit deck" button, and a profile picture of the presenter. The main content of the slide is centered and consists of the word "Vite" in a large font, a yellow lightning bolt icon, and the text "Native-ESM powered web dev build tool". Below this, it says "Remote.vue", "2020.06.12", and "@kazupon". At the bottom of the slide, there is a green text overlay that reads "プラグイン作りたいたい人". The bottom of the screenshot shows the Speaker Deck interface with the presenter's name "kazupon", the date "June 12, 2020", a "Programming" tag, 3 stars, 420 views, and a download icon.

プラグイン作りたいたい人
内部実装について知りたい方はぜひ！

<https://speakerdeck.com/kazupon/native-esm-powered-web-dev-build-tool>

Vite の特徴

- ES Modules ベースの開発サーバを備えたモダンなビルドツール
- dev server は高速
 - 起動時にすべてバンドルしない
 - 高速な ESbuild による on the fly コンパイル
- Hot Module Replacement (HMR) もシンプルなため速い

機能

- Bare Module Resolving
- Hot Module Replacement
- TypeScript
- CSS / JSON Importing
- Asset URL Handling
- PostCSS
- CSS Modules
- CSS Pre-processors
- JSX
- Custom Blocks
- Config File
- Dev Server Proxy
- Production Build
- Modes and Environment Variables

<https://github.com/vitejs/vite#features>

機能

- Web Assembly
- Inline Web Workers
- HTTPS/2

生まれた背景

- VuePress の起動が遅くてイライラした
- SFC を ES Modules で動かしたいというアイデアが昔からあった
- Snowpack などの新しい ES Modules ベースのバンドラが出現してきた

そんな中アイデアが舞い降りてきて生まれた



Evan You

@youyuxi



As I was going to bed, I had an idea about a no-bundler dev setup (using native browser ES imports), but with support for Vue SFCs ****with hot reload****. Now it's almost 6AM and I have PoC working. The hot reload is so fast it's near instant.

♡ 1,705 6:54 PM - Apr 20, 2020



💬 190 people are talking about this



<https://twitter.com/youyuxi/status/1252173663199277058>

Vite の現状

- v1.0.0 RC としてリリースされているので、近々正式 v1.0 がリリースされる見込み
- プロダクション向けのビルドは、実質 rollup でビルドするので使えるはず！...

Vue と Vite との関係

- Vue 3 向けの公式ドキュメントで Vite を使ったインストールについて書いてある
- なので、Vue 向けにはサポートされていくのは間違いない

開発環境がすぐに整う！

- `npm init vite-app` のようなコマンド
TypeScript とか諸々の環境が整う！
- なので、みなさん、使ってください！

@vuedx

Vue developer experience


- Vue の DX をよくするためのツール
- コアチームの Rahul 氏 (znck) が作成中
- まだ Experimental



余談: ちなみに Rahul 氏は 2018 年日本に来ている

- Vue Fes Japan 2018
- コンパイラ周り が 得意
- rollup-plugin-vue の 開発者

SPEAKER



オープンソース・エンスージヤスト
Rahul Kadyan

[Twitter](#) [GitHub](#)


Rahul は開発者でオープンソース・エンスージヤストです。彼は Vue エコシステム向けにツール上で動作するコンパイラユーティリティを作っています。

彼は、Myntra (Flipkart Group) でシニアエンジニアとして働いています。

SESSION [会場A](#)

A deep dive in SFC compilation

Vue single file components empowers developers with ability to write declarative render functions using plain HTML-like syntax. The so called SFCs also provide features like collocation and scoped styles. We tend to use these SFCs effortlessly as any other JavaScript file but to reach that level of ease, Vue does the heavy lifting of transpilation. This talk takes a deep dive in the compilation process of Vue components and explains how to write a custom block processor. It also discovers some less known facts about SFCs.



Rahul Kadyan - A deep dive in SFC compilation : Vue Fes Japan 2018

後で見る 共有

<https://vuefes.jp/2018/speakers/znck/>

GitHub リポジトリ

- @vuedx は Rahul 氏の個人リポジトリで開発が進められている

The screenshot shows the GitHub repository page for `znck/vue-developer-experience`. The repository has 9 watchers, 318 unstars, and 6 forks. The main content area displays a list of files and folders with their commit history. The most recent commit is `znck chore: make tests slightly faster` (commit hash `237b9f6`) from 13 days ago, with 95 total commits. The file list includes:

File/Folder	Commit Message	Time Ago
<code>znck</code>	<code>chore: make tests slightly faster</code>	13 days ago
<code>.vscode</code>	<code>test: component tag renaming</code>	13 days ago
<code>examples</code>	<code>test: component tag renaming</code>	13 days ago
<code>extensions/vscode</code>	<code>chore: release</code>	13 days ago
<code>packages</code>	<code>chore: make tests slightly faster</code>	13 days ago
<code>scripts</code>	<code>fix: exclude .vue files from TypeScript program (only virtual files a...</code>	4 months ago
<code>types</code>	<code>feat: use TSX instead of render function</code>	29 days ago
<code>.editorconfig</code>	<code>refactor: remove unnecessary method overrides from ts plugin</code>	4 months ago
<code>.gitignore</code>	<code>test: generate grammar</code>	4 months ago
<code>.npmrc</code>	<code>Initial Commit</code>	5 months ago
<code>.prettierrc</code>	<code>refactor: remove unnecessary method overrides from ts plugin</code>	4 months ago
<code>.travis.yml</code>	<code>chore: add xvfb and libsecret-1-0 to run extension tests on CI</code>	4 months ago

The right sidebar contains the following information:

- About:** A collection of tools for vue ecosystem. Includes a `Readme` and `MIT License`.
- Releases:** No releases published.
- Packages:** No packages published.
- Used by:** 1 user, `@znck / vue-developer-experience`.

<https://github.com/znck/vue-developer-experience>

@vuedx はどのようなもの(DX)を提供するのか

@vuedx が提供するもの

- @vuedx のゴールとしては以下を提供すること
 - <template> 内での型チェック
 - コード補完
 - リファクタリング機能
 - SFC の ES Modules 化

<template> 内での型チェック

```
preview — znck@Rahuls-MacBook-Pro — ../znck/preview — -zsh — 143x48
=> ./browser/components/Browser.vue:20:23
Suggestion :: Parameter 'viewport' implicitly has an 'any' type, but a better type may be inferred from usage. (code 7044)
18 |     <Viewport :width="width" :height="height">
19 |       <template #default="viewport">
20 |         <slot v-bind="viewport" />
    |                   ^^^^^^^^^
21 |       </template>
22 |     </Viewport>

./browser/components/Mobile.vue
=====
=> ./browser/components/Mobile.vue:77:25
Error :: Property 'browser' does not exist on type vue component 'Mobile'. (code 2339)
75 |     <Browser :device="type" :width="config.width" :height="config.height">
76 |       <template #default="browser">
77 |         <slot v-bind="browser" />
    |                   ^^^^^^^
78 |       </template>
79 |     </Browser>

=> ./browser/components/Mobile.vue:75:32
Error :: Type 'number' is not assignable to type 'string'. (code 2322)
73 |     }"
74 |   >
75 |     <Browser :device="type" :width="config.width" :height="config.height">
    |                   ^^^^^
76 |       <template #default="browser">
77 |         <slot v-bind="browser" />
The expected type comes from property 'width' which is declared here on type 'IntrinsicAttributes & Props & VNodeProps & AllowedComponentProps & ComponentCustomProps'

=> ./browser/components/Mobile.vue:75:54
Error :: Type 'number' is not assignable to type 'string'. (code 2322)
73 |     }"
74 |   >
75 |     <Browser :device="type" :width="config.width" :height="config.height">
    |                   ^^^^^
76 |       <template #default="browser">
77 |         <slot v-bind="browser" />
The expected type comes from property 'height' which is declared here on type 'IntrinsicAttributes & Props & VNodeProps & AllowedComponentProps & ComponentCustomProps'

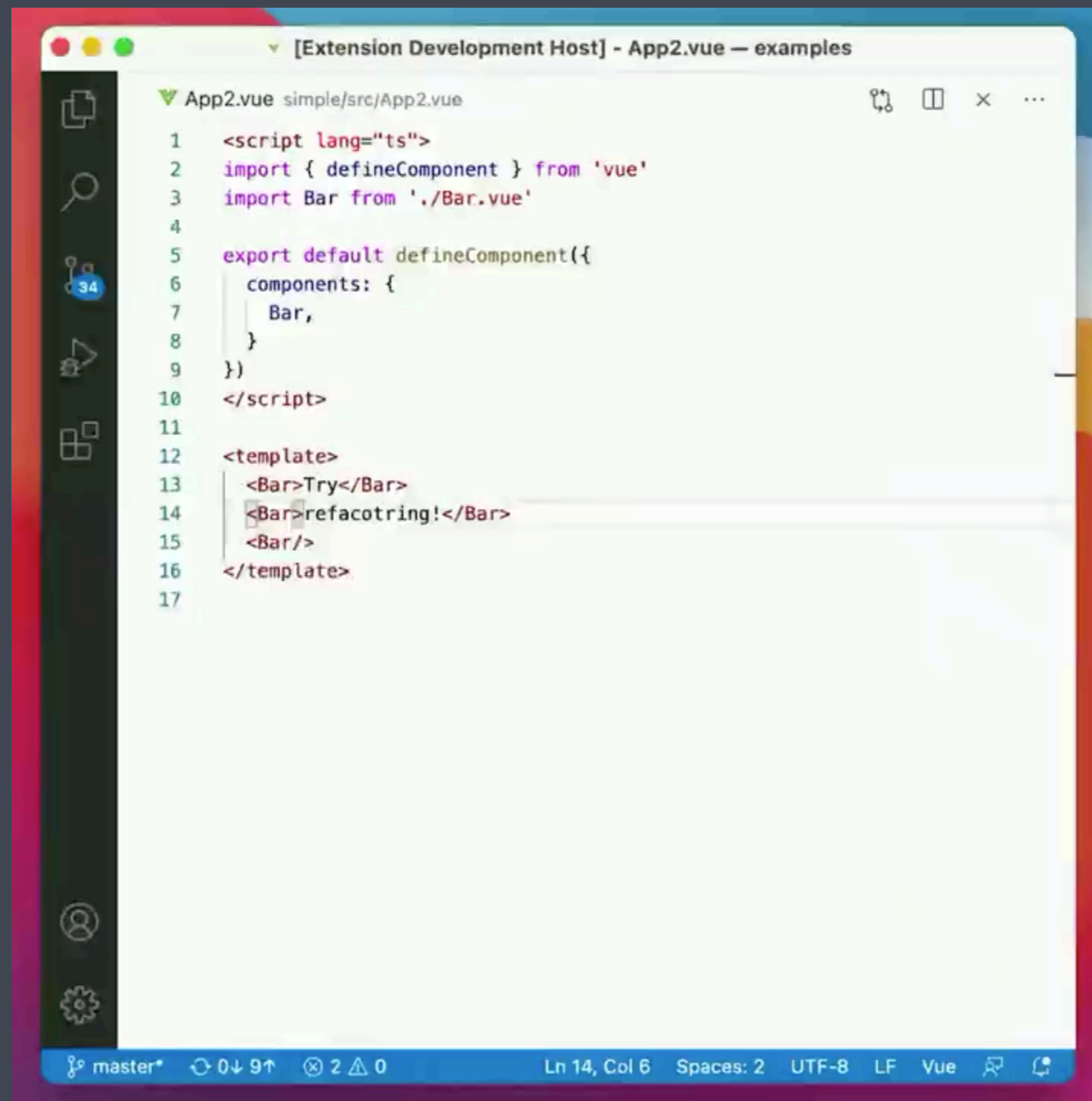
=> ./browser/components/Mobile.vue:2:27
Suggestion :: 'ref' is declared but its value is never read. (code 6133)
1 | <script lang="ts">
2 | import { defineComponent, ref, computed } from 'vue';
  |                   ^^^
```


リファクタリング機能

- `<template>`と`<script>`との同期
- Props の変更
- コンポーネントタグの変更
- ファイル名変更とコンポーネントの抽出
- 変数名の変更
- ... など

デモ

- 例: コンポーネントタグの変更



```
[Extension Development Host] - App2.vue — examples
App2.vue simple/src/App2.vue
1 <script lang="ts">
2 import { defineComponent } from 'vue'
3 import Bar from './Bar.vue'
4
5 export default defineComponent({
6   components: {
7     Bar,
8   }
9 })
10 </script>
11
12 <template>
13   <Bar>Try</Bar>
14   <Bar>refacotring!</Bar>
15   <Bar/>
16 </template>
17
```

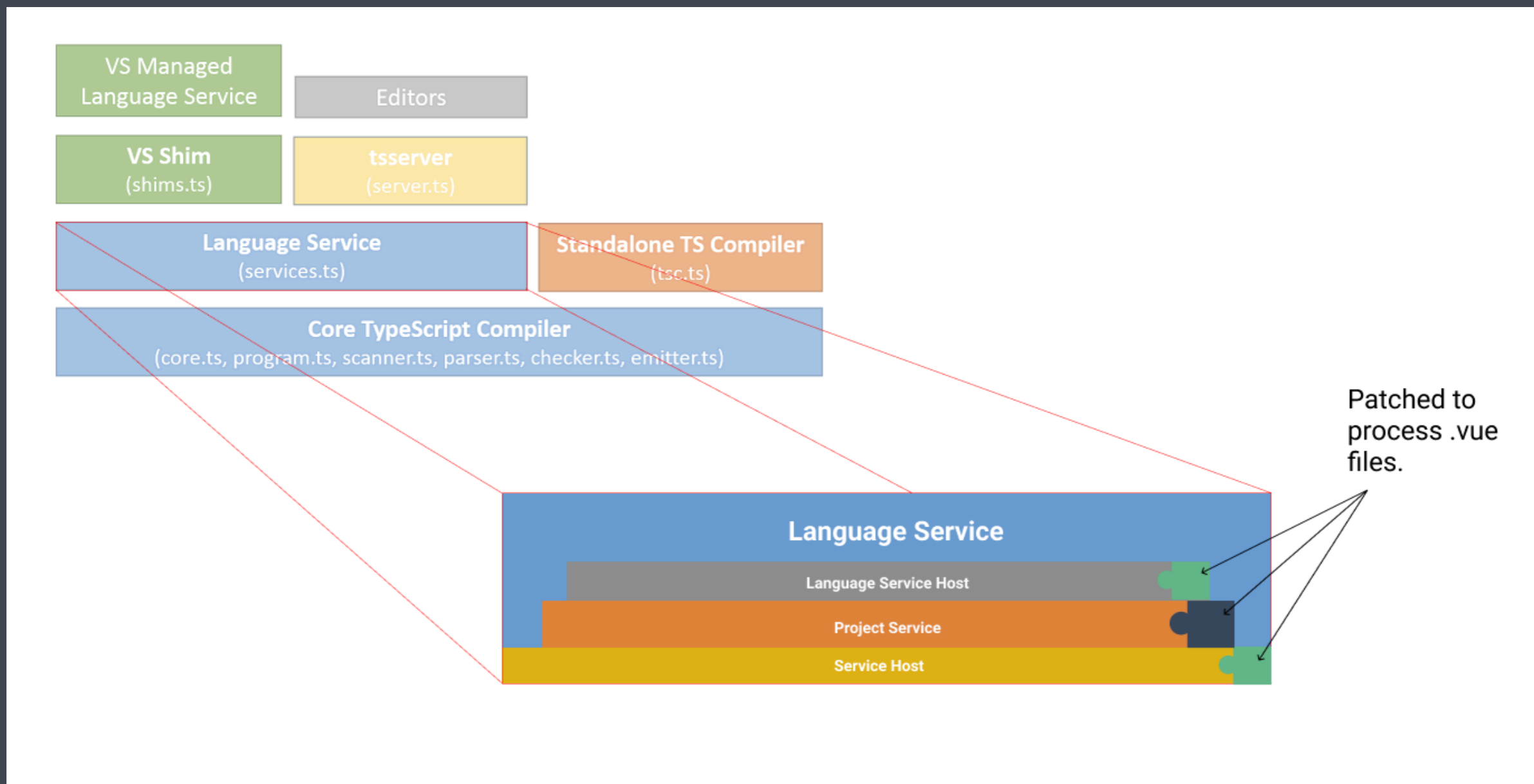
Ln 14, Col 6 Spaces: 2 UTF-8 LF Vue

<https://twitter.com/znck0/status/1290577650398846976>

@vuedxでは

これをどうやっているのか

アーキテクチャ



VS Code の Language Service にパッチする
感じで SFC を処理している

Vue Virtual TextDocument

- SFC 各ブロックを仮想ファイルとして処理

component.vue

```
<template>
  <div>{{ foo }}</div>
</template>

<script lang="ts">
import { defineComponent } from 'vue'

export default defineComponent({
  setup () {
    return { foo: 0 }
  }
})
</script>
```

component.vue__template.ts

```
-----
-----
-----
-----
-----
-----
-----
<div>{{ foo }}</div>
</template>
```

component.vue__script.ts

```
-----
import { defineComponent } from 'vue'

export default defineComponent({
  setup() {
    return { foo: 0 }
  }
})
```

component.vue__render.ts

```
import { h as _h } from 'vue'
import { JSX } from '<not decided yet>'
import _Ctx from './component.vue'

export function render(_ctx: _Ctx) {
  return h(JSX.intrinsic.div, null, [_ctx.foo])
}
```

TS のコード
として処理する

@vuedx、
Vetur と同じよう
なツールでは？



Vetur は今後どうなる？

今後は統合していく予定

Explore TS Plugin support in Vetur by integrating znck/developer-experience's refactor support #2016

New issue

Open octref opened this issue on Jul 5 · 2 comments



octref commented on Jul 5 · edited -

Member



Chatted with @znck, here's a summary:

[znck/vue-developer-experience](#) is a TS Plugin that uses a virtual file system to support Vue files. Currently it only supports Vue 3 but not Vue 2, because it creates virtual files without modifying AST (so you cannot wrap `export default {}` to `export default Vue.extend({})`). It adds most TS language features to Vue files.

Vetur currently has limitations that can only be solved by a TS Plugin. For example, rename/refactor in JS/TS files should generate changes across a project's JS/TS/Vue files, but with Vetur's current approach it won't support Vue files, so renaming might create compiler errors.

A lot of Vetur's static analysis code can live either in language server or in a TS Plugin, but both have their limitations:

- Language Server: cannot interfere with TS language features
- TS Plugin: has no access to VS Code API

Assignees

No one assigned

Labels

integ:typescript

plan

Projects

None yet

Milestone

No milestone

Linked pull requests

Vetur に @vuedx のTSプラグインを取り入れてリファクタリング機能を置き換える

<https://github.com/vuejs/vetur/issues/2016>

その他

Reactivity API

以前の2月の発表のときは

- reactive
- computed
- watch
- readonly
- ref
- isRef
- toRefs

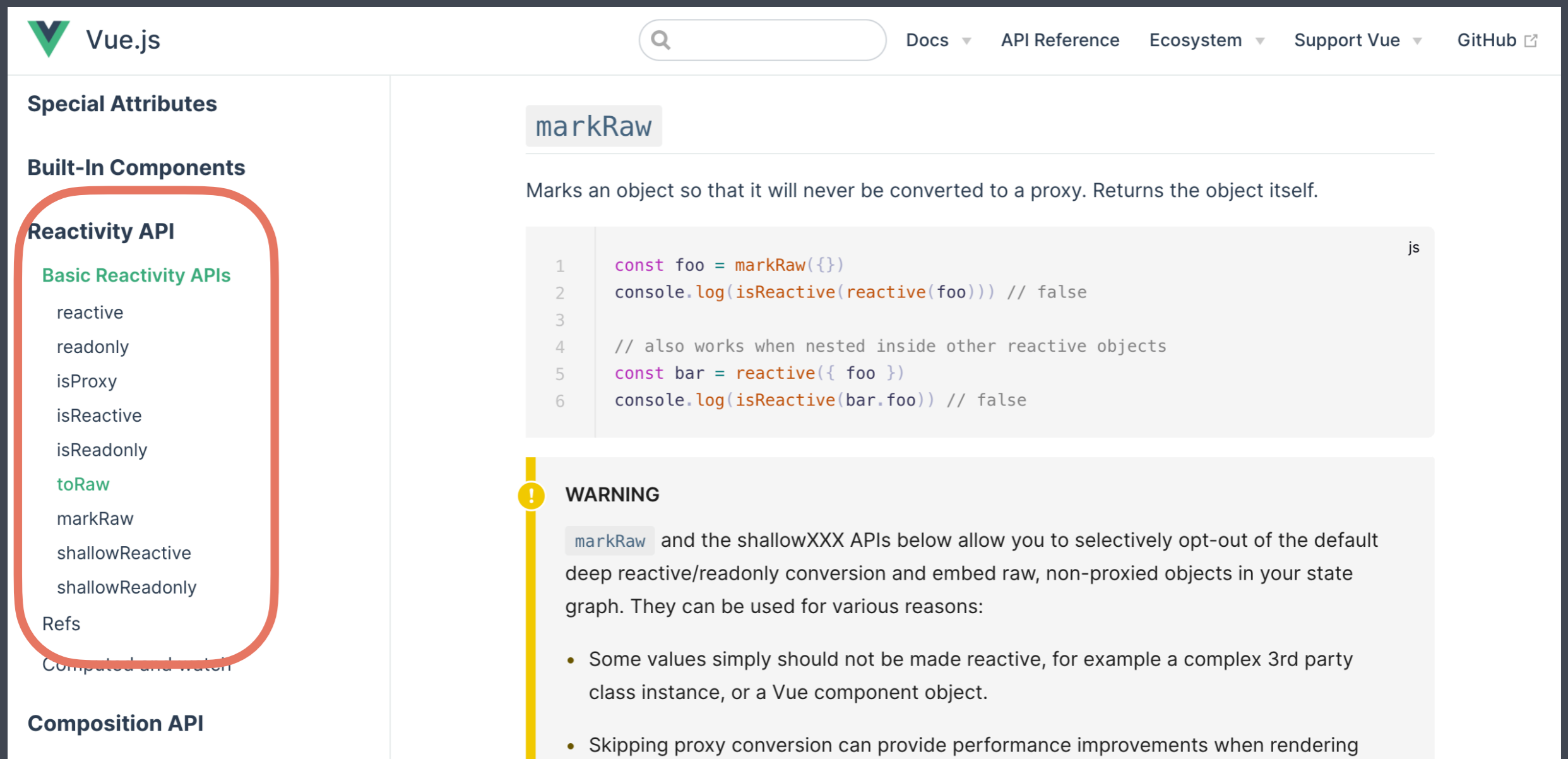
上記のような基本的なAPIしかなかったが...

さらに以下のAPIが増えた

- shallowReactive
- shallowRef
- markRaw
- triggerRef
- isReactive
- isProxy
- toRaw

主にパフォーマンスの最適化、高度なことを
したい人向けに公開された

詳しい詳細は公式ドキュメントで



The screenshot shows the Vue.js documentation website. The left sidebar contains a navigation menu with the following items: Special Attributes, Built-In Components, **Reactivity API** (highlighted with a red rounded rectangle), and Composition API. Under 'Reactivity API', there are sub-sections: Basic Reactivity APIs (with a list of reactive, readonly, isProxy, isReactive, isReadonly, toRaw, markRaw, shallowReactive, shallowReadonly), Refs, and Computed and watch. The main content area shows the 'markRaw' API page, which includes a description, a code example, and a warning section.

Vue.js

Docs ▾ API Reference Ecosystem ▾ Support Vue ▾ GitHub ↗

markRaw

Marks an object so that it will never be converted to a proxy. Returns the object itself.

```
1 const foo = markRaw({})
2 console.log(isReactive(reactive(foo))) // false
3
4 // also works when nested inside other reactive objects
5 const bar = reactive({ foo })
6 console.log(isReactive(bar.foo)) // false
```

WARNING

`markRaw` and the shallowXXX APIs below allow you to selectively opt-out of the default deep reactive/readonly conversion and embed raw, non-proxied objects in your state graph. They can be used for various reasons:

- Some values simply should not be made reactive, for example a complex 3rd party class instance, or a Vue component object.
- Skipping proxy conversion can provide performance improvements when rendering

Reactivity API の項目に載っています

<https://v3.vuejs.org/api/basic-reactivity.html#reactive>

Vue.js コアチームの記事も読むとよい

- Reactive の歴史を交えながら解説している

The screenshot shows a Dev.to article page. On the left is a sidebar with navigation icons: a heart with '36', a hand with '7', a bookmark with '41', and a menu icon. The main content area features the article title 'Understanding Reactivity in Vue 3.0' with tags for #vue and #javascript. The author is Jinjiang, with a bio '0.1.x Engineer + 🌐' and a 'Follow' button. The article text begins with 'This article is just written for my knowledge and understanding of the coolest part in Vue: the reactivity system.' and a 'Background' section starting with 'As we know, Vue.js team is working on 3.0 for a while. Recently it released the first Beta version. That means the core tech design is stable enough. Now I think it's time to walk through something inside Vue 3.0. That's one of my most favorite parts: the reactivity system.'

Understanding Reactivity in Vue 3.0

#vue #javascript

Jinjiang 5月27日 · 24 min read

0.1.x Engineer + 🌐

Follow

WORK
Frontend engineer

LOCATION
Singapore

JOINED
2020年5月26日

Trending on DEV 🔥

What chair do you use for coding?
#discuss #help

What would you do if you had a Time Machine?

<https://dev.to/jinjiang/understanding-reactivity-in-vue-3-0-1jni>

分かりやすい Reactivity API についての資料

- ・ Ref と Reactive を使い分けが日本語で分かりやすく書いてる



<https://speakerdeck.com/kawamataryo/ref-vs-reactive-vue-composition-api-deep-in>

@vue/

composition-api

v1.0.0 がリリースされた

- ・ ジョインした Anthony 氏や Carlos 氏によって開発が進んだ



Anthony 氏

- ・ vue-next から足りないAPIがバックポートされている



Carlos 氏

これはどういうこと？

初期の頃は注意書きがあった

Update README.md Browse files

master v3.0.0-beta.6 v0.5.0

yyx990803 committed on Feb 26 Verified 1 parent a5f5a86 commit bde4d1b1900c98e13d7ec3e555425e19afc1b3f1

Showing 1 changed file with 2 additions and 0 deletions. Unified Split

```
2 README.md
```

4	5
5	5
6	6
7	7
8	8
9	9
10	10
11	11

0 comments on commit bde4d1b Lock conversation

本番環境で使うのは推奨しない！と

開発が進むようになってからは...

- ・ バックポートするうちにこの注意書きはなくなった

```
... @@ -1,37 +1,19 @@
1 - # Vue Composition API
2
3 - Vue 2 plugin for Composition API in Vue 3.
4
5   [[npm](https://img.shields.io/npm/v/@vue/composition-api)]
   (https://www.npmjs.com/package/@vue/composition-api)
6   [[GitHub Workflow Status]
   (https://img.shields.io/github/workflow/status/vuejs/composition-
   api/Build%20&%20Test)](https://github.com/vuejs/composition-api/actions?
   query=workflow%3A%22Build+%26+Test%22)
7
8 - English | 中文文档(./README.zh-CN.md) / Composition API RFC
   (https://composition-api.vuejs.org/)
9
10
11 - Note: the primary goal of this package is to allow the community to
   experiment with the API and provide feedback before it's finalized. The
   implementation may contain minor inconsistencies with the RFC as the
   latter gets updated. We do not recommend using this package for production
   yet at this stage.
12
```

```
... @@ -1,37 +1,19 @@
1 + # @vue/composition-api
2
3 + Vue 2 plugin for Composition API
4
5   [[npm](https://img.shields.io/npm/v/@vue/composition-api)]
   (https://www.npmjs.com/package/@vue/composition-api)
6   [[GitHub Workflow Status]
   (https://img.shields.io/github/workflow/status/vuejs/composition-
   api/Build%20&%20Test)](https://github.com/vuejs/composition-api/actions?
   query=workflow%3A%22Build+%26+Test%22)
7
8 + English | 中文(./README.zh-CN.md) • Composition API Docs
   (https://composition-api.vuejs.org/)
9
10 + </p>
11
12 + ## Installation
```

つまり、本番環境で使えることを意味します

- Vue 3 は RC フェーズ
- Vue 2 と Vue 3 を互換を保つライブラリなど作っている作者によって実証されている



Anthony 氏



Carlos 氏

つまり、本番環境で使えることを意味します

- ・ 公式的には断言はしていないが、バックポートは成功したと！明言している

🔔 Open

Vue 3 is now in RC! #189

yyx990803 opened this issue on Jul 18 · 94 comments

Migrating non-trivial applications from v2 to v3 will likely be a much slower process. We will provide code mods and tools to help with such migrations, but in most cases, this will depend on how fast the project's dependencies can be upgraded to support Vue 3. It is also important to evaluate whether the risk and time investment of upgrading is worth it - since Vue 2 will continue to be supported. We plan to dedicate a focus period after 3.0 release to back-port features into v2 via compatibility plugins. We are already seeing success with this approach in [@vue/composition-api](#).

ご利用は計画的に

- 制限事項がある

Limitations

✔ Support ✖ Not Supported

Ref Unwrap

Unwrap is not working with Array index.

- ✖ Should NOT store `ref` as a direct child of `Array`
- ✖ Should NOT use `ref` in a plain object when working with `Array`
- ✔ Should always use `ref` in a `reactive` when working with `Array`
- ⚠ ``set`` workaround for adding new reactive properties

Template Refs

- ✔ String ref && return it from `setup()`
- ✔ String ref && return it from `setup()` && Render Function / JSX
- ✖ Function ref
- ✖ Render Function / JSX in `setup()`

<https://github.com/vuejs/composition-api#limitations>

ご利用は計画的に

- パフォーマンスが気にならないかどうか

Name Duration for...	vue-next- v3.0.0- beta.15- keyed	vue- v2.6.2- keyed	vue2- compositi on-api- v0.6.5- keyed
Issues for the implementation			
create rows creating 1,000 rows	149.4 ± 4.1 (1.00)	174.9 ± 3.4 (1.17)	188.1 ± 7.7 (1.26)
replace all rows updating all 1,000 rows (5 warmup runs).	146.7 ± 1.2 (1.00)	162.9 ± 2.3 (1.11)	168.7 ± 1.2 (1.15)
partial update updating every 10th row for 1,000 rows (3 warmup runs). 16x CPU slowdown.	299.2 ± 16.7 (1.00)	352.4 ± 16.1 (1.18)	369.9 ± 22.3 (1.24)
select row highlighting a selected row. (no warmup runs). 16x CPU slowdown.	216.5 ± 12.8 (1.00)	424.5 ± 18.4 (1.96)	508.5 ± 5.6 (2.35)
swap rows swap 2 rows for table with 1,000 rows. (5 warmup runs). 4x CPU slowdown.	66.5 ± 1.7 (1.00)	85.6 ± 3.2 (1.29)	93.9 ± 3.7 (1.41)
remove row removing one row. (5 warmup runs).	30.3 ± 0.1 (1.00)	34.1 ± 0.6 (1.13)	35.9 ± 0.3 (1.18)
create many rows creating 10,000 rows	1,356.9 ± 9.9 (1.00)	1,496.3 ± 21.3 (1.10)	1,681.4 ± 33.9 (1.24)
append rows to large table appending 1,000 to a table of 10,000 rows. 2x CPU slowdown	335.5 ± 5.8 (1.00)	371.0 ± 6.2 (1.11)	417.3 ± 11.0 (1.24)
clear rows clearing a table with 1,000 rows. 8x CPU slowdown	191.2 ± 4.2 (1.00)	222.0 ± 6.9 (1.16)	229.9 ± 7.7 (1.20)
slowdown geometric mean	1.00	1.22	1.33

Memory allocation in MBs ± 95% confidence interval

Name	vue-next- v3.0.0- beta.15- keyed	vue- v2.6.2- keyed	vue2- compositi on-api- v0.6.5- keyed
ready memory Memory usage after page load.	1.2 (1.00)	1.2 (1.04)	1.3 (1.07)
run memory Memory usage after adding 1000 rows.	3.4 (1.00)	4.0 (1.19)	5.1 (1.52)
update each 10th row for 1k rows (5 cycles) Memory usage after clicking update every 10th row 5 times	3.6 (1.00)	4.4 (1.23)	5.5 (1.55)
replace 1k rows (5 cycles) Memory usage after clicking create 1000 rows 5 times	4.0 (1.00)	4.6 (1.15)	5.8 (1.45)
creating/clearing 1k rows (5 cycles) Memory usage after creating and clearing 1000 rows 5 times	2.6 (1.00)	2.7 (1.01)	2.8 (1.07)
slowdown geometric mean	1.00	1.12	1.31

まとめ

まとめ


- 以前自分が発表から、Vue 3 RC がリリースされるまでにいる
いと進展があった
 - SFC 改善 RFC
 - Vite 爆誕
 - @vuedex の開発
 - @vue/composition-api 1.0 リリース
- 今後も正式リリースまでに進展があると思うので、情報
キャッチアップしていただければ。

最後に

GitHub Sponsors


- よりOSS開発の活動していききたいので、サポートしてくれるとうれしいです！

Become a sponsor to kazupon



kazuya kawaguchi
kazupon
Tokyo, Japan

Hello! 🐱
My nickname is kazupon! I'm [Vue.js Core Team!](#)
I work on many Open Source projects like [vue-i18n](#), [vue-cli-plugin-i18n](#), [intlify](#) and more!



Why GitHub Sponsors?


As for my contribution to open source as above, while I'm working as a job, I'm organizing [the Japanese community](#) and contributing to Vue.js official open-source projects, and I have personal some open source projects. So far, I have worked these contributions after finishing my regular work and finding hiatus.

Currently, It's improved, but it still has a long way to go until full-time. 🐱

By becoming a GitHub Sponsors, if my rewards goals are achieved, it'll be less dependent on current my work, so far my open source contribution times are surely maintained. it means that I can further focus on achieving the roadmap and

3% towards \$6,000 per month goal

posva and 13 others sponsor this goal

Sponsor as  kazupon

For each tier you'll receive a badge that shows @kazupon you're a sponsor. Hover over your avatar to see.

Select a tier

\$5 a month [Select](#)

Coffee Supporters ☕

- Show your support on your profile with a Sponsor badge.
- Buy me a monthly cup of coffee.

\$10 a month [Select](#)

Kazupon Fan 🐱

- All of the above
- You can chat at discord with me about Vue related topics and my open source projects. (1 session / month)

<https://github.com/sponsors/kazupon>

ご清聴

ありがとうございました！