

Vue2.0 升级 Vue3.0 成本评估报告

2021-4-28

前言

KOCA 前端技术体系使用 Vue 版本 V2.6.11

Vue3.x 正式发版时间 2020.09.18

Vue3.0 正式发版已有一段时间了,有一些团队已经开始启用 vue3.0 开展新的项目,但大部分朋友还是守着 vue2.0 老项目过日子,那么对于 KOCA 前端框架来说,做一次框架升级到底会带来哪些成本和收益?

关注要点

- ① vue3.x 新特性及这些新特性给开发者和产品带来的好处
- ② 升级 3.x 所面临问题
- ③ 是否有必要升级? 升级的计划安排
- ④ 现有 koca 前端支持分布式部署方案? 那么这个方案是否支持微应用使

用 vue3.0 框架

针对以上关注要点,我们做了一些调研

Vue3.x 新特性

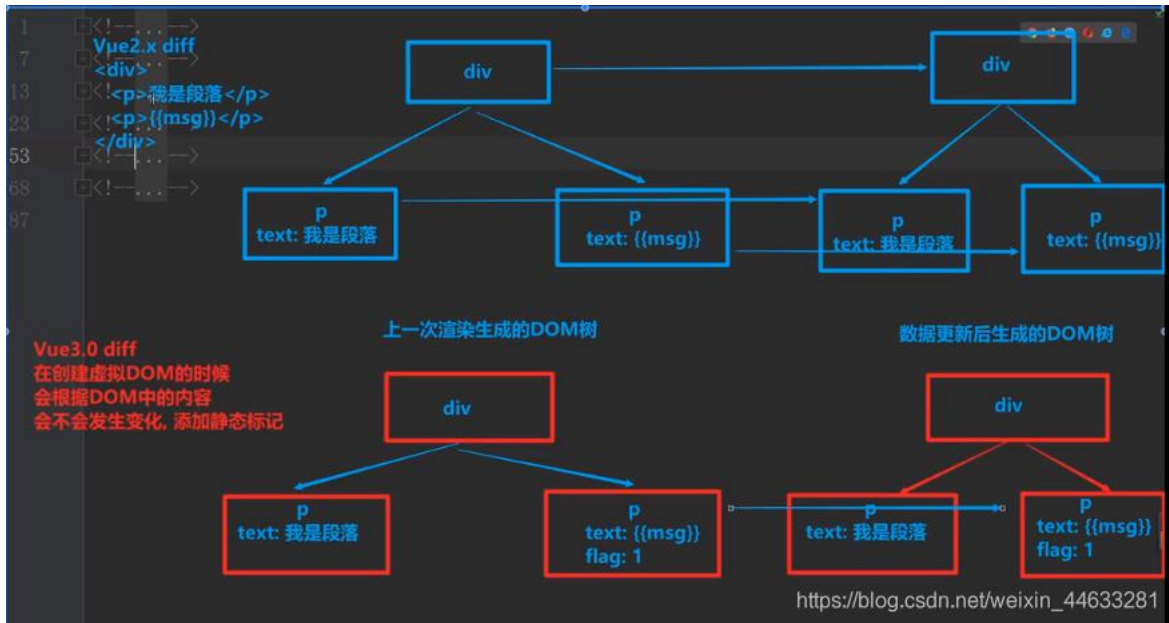
- 一、性能比 Vue2.x 快近 2 倍

- 二、按需编译, 体积比 vue2.x 更小
- 三、提供组合 API, composition api
- 四、更好的 ts 支持
- 五、新增 3 个组件: Frament, Teleport, Suspense
- 六、暴露了自定义渲染 api, Custom Renderer API

一、性能比 Vue2.x 快近 2 倍

| | Vue2.x | Vue3.x |
|---------------------------|---|---|
| 重写了虚拟 DOM 的实现, diff 算法的优化 | Vue2.x 中的 dom 是进行全量对比渲染 (对应的所有 dom 都对比一遍, 增加了对比次数和渲染速度) | 添加了静态标记 (与上次 dom 节点对比的时候, 只对比有静态) |
| hoistStast 静态提升 | vue2.x 无论元素是否参与更新, 每次都要重新创建, 然后渲染 | vue3.0 中对于不参与更新的元素, 会做静态提升, 只会被创建一次, 在渲染时直接复用即可 |
| cacheHandlers 事件侦听器缓存 | vue2.x 默认情况下 onClick 会被视为动态绑定, 所以每次都会去追踪它的变化; | Vue3.0 因为是同一个函数, 所以没有追踪变化, 直接缓存起来复用即可; |
| ssr 渲染 | vue2.x 当有大量静态的内容时, 这些内容会被当做纯字符串推进一个 buffer 里面, 即使存在动态的绑定, 会通过模板插值嵌入进去, 这样会比通过虚拟 dom 来渲染的快很多 | vue3.0 当静态文件大到一定量的时候, 会用 _ceratStaticVNode 方法在客户端去生成一个 static node, 这些静态 node, 会被直接 innerHtml, 就不需要创建对象, 然后根据对象渲染 |

表 1Vue2.x 与 Vue3.x 对比



```
<div>
  <p>知播渔</p>
  <p>知播渔</p>
  <p>知播渔</p>
  <p>{{msg}}</p>
</div>

export function render(_ctx, _cache, $props, $setup, $data, $options) {
  return (_openBlock(), _createBlock("div", null, [
    _createVNode("p", null, "知播渔"),
    _createVNode("p", null, "知播渔"),
    _createVNode("p", null, "知播渔"),
    _createVNode("p", null, _toDisplayString(_ctx.msg) + "", /* TEXT */)
  ]))
}
```

https://blog.csdn.net/weixin_44633281

二、按需编译, 体积比 vue2.x 更小

vue3 中的核心 api 都支持了 tree-shaking, 这些 api 都是通过包引入的方式而不是直接在实例化时就注入, 只会对使用到的功能或特性进行打包 (按需打包), 这意味着更多的功能和更小的体积。

三、提供组合 API, composition api

vue2 中, 我们一般会采用 mixin 来复用逻辑代码, 用倒是挺好用的, 不过也

存在一些问题：例如代码来源不清晰、方法属性等冲突。基于此在 vue3 中引入了 Composition API (组合 API), 使用纯函数分隔复用代码。和 React 中的 hooks 的概念很相似。

1. 更好的逻辑复用和代码组织
2. 更好的类型推导

四、新增 3 个组件: Fragment, Teleport, Suspense

①Fragment:模板可以有多个根元素

②Teleport: 提供一种将子节点渲染到存在于父组件以外的 DOM 节点的优秀方案。

③Suspense 让你的组件在渲染之前进行“等待”。

五、暴露了自定义渲染 API, Custom Renderer API

这个 api 定义了虚拟 DOM 的渲染规则，这意味着使用自定义 API 可以达到跨平台的目的。

自定义渲染器 api, vue 2 也可以做，之前 weex、nativescript vue 都是通过 vue 2 的一个非暴露的 api，去实现这个功能。但是 vue 2 引入了一些依赖上同步的问题。vue 3 就是一个内置的 api。

以上 3, 4 为比较明显的特性，但这两个特性解决的问题目前也不是很致命

组合 API. 新的 API 兼容 Vue2.x, 只需要在项目中，单独引入 @vue/composition-api 这个包

对 typescript 的友好支持，针对喜欢写 ts 的人，在 vue2.x 的基础上，额外引入一些包，加一些装饰器，基本也能用

升级 vue3.x, 可能面临的问题

- vue3.0 非兼容的改变与建议

<https://v3.cn.vuejs.org/guide/migration/array-refs.html>

- 当前项目生态中的几个库都面临巨大升级, 以及升级后要填的一些坑, 比如:

vue-router (vue-router@4)

Vuex (vuex@4)

ElementUI (Element plus, 不兼容列表

<https://github.com/element-plus/element-plus/issues/162>) 等。

这些生态在 2.x 上都已经很成熟, 在 3 上可能还有很长的路要走

- Vue3.x 没有默认对象 export default, 当前项目中所有直接使用 Vue.xxx 的语法全部得重写
- 放弃对 IE11 的支持 (原有的兼容版本, 因为微软推进 Edge 浏览器, 并打算放弃 IE11 所以兼容版本不会出)

Vue 2 的反应系统基于 ES5 getter / setter。Vue 3 利用 ES2015 代理获得了性能更高且更完整的反应系统, 该系统无法在 IE11 中进行多填充。这是主要障碍, 因为这意味着 Vue 3 要支持 IE11, 它实际上需要发布两个具有不同行为的不同版本-一个使用基于 Proxy 的反应系统, 另一个使用类似于 Vue 2 的基于 ES5-getter / setter 的系统。。

Vue 3 的基于代理的反应性系统提供了几乎完整的语言功能覆盖。它能够检测许多在 ES5 中不可能或不可行的操作, 例如属性添加/删除, 数组索引和 length 突变以及 in 操作员检查。为 Vue 3 的代理版本编写的相同代码在 IE11 版本中不起作用。这不仅给我们带来了技术上的复杂性, 也给开发人员带来了持续的精神负担。

尤玉溪在知乎上关于 vue3 的 IE11 支持

摘要

- Vue 3 将不会支持 IE11
- 原定投入 Vue 3 IE11 支持的精力将投入给 2.7，移植 3.x 兼容的新功能，包括：
 - Composition API
 - `<script setup>` 以及其它新的单文件组件特性
 - emits 选项
 - TS 类型改进
 - Vite 官方整合

Vue3.x 为什么有必要用

1. 兼容 Vue2.x 95%以上特性，所以其优点全部得到集成，上手快，门槛低
2. 对 Ts 支持更加友好，更加迎合行业趋势，技术潮流
3. Composition API 的添加，会让一部分 React 用户转投过来，以后群体会更大，生态更丰富

Vue3.x 与微前端的结合

KOCA 的微前端支持方案基于 vue2.x，即主应用是基于 vue2.x，问，微应用是否可以基于 Vue3.0 开发？

可以，但是微应用只能以 `iframe` 的形式嵌入。因为当一个微前端中同时存在两个 vue 版本时，vue 原型会被覆盖。

结论：

研发中心作为一个研发性质的部门，有必要去使用 Vue3.x，但基于 KOCA 目前一个正在使用的前端平台，可以尝试 Vue2.x + `@vue/composition-api` 过渡，不建议断崖式升级

前端团队计划安排（引用）：

1. Vue2.7 + @vue/composition-api, 支持 Vue3.x 的新语法, 同时制定严格的 Composition API 使用规范

2. 基于 Vue3.x 搭建新的基础框架 (全面 Ts) ! 不会基于老的框架升级!
同时制定详细的 Vue3.x 模板开发规范

3. 从此以后, 老的项目依然使用 Vue2.7+ @vue/composition-api 维护直至退役, 或者完全被新的 Vue3.x 框架替代成全新的工程

使用 KOCA 前端的产品

证券软件总部 7

金证财富 16

金证前海 3

智慧城市 3

睿服科技 2

创新合作部 2