

我是十三 Lv3

2020年09月21日 阅读 8248

关注

Vue 3.0 来了，我们该做些什么？

我曾七次鄙视自己的灵魂：

第一次，当它本可进取时，却故作谦卑；

第二次，当它空虚时，用爱欲来填充；

第三次，在困难和容易之间，它选择了容易；

第四次，它犯了错，却借由别人也会犯错来宽慰自己；

第五次，它自由软弱，却把它认为是生命的坚韧；

第六次，当它鄙夷一张丑恶的嘴脸时，却不知那正是自己面具中的一副；

第七次，它侧身于生活的污泥中虽不甘心，却又畏首畏尾。

 116 24 收藏

...

想必 Vue3.0 发布这件事，大家都知道了。



我也是从朋友圈的转发得知此事，博客平台、公众号、朋友圈基本都有这么一条新闻，可见 Vue3.0 的被期待程度，因为 React 16 发布的时候，我也没见大家这么追捧，让我有点震惊的是 Vue 有 130 万的使用者，这个体量真的是有点惊人。

此时，一位靓仔偶然路过~



Vue 3.0 来了，我们该做些什么呢？

- 学习，赶紧学习，学不动了也要学！
- 装不知道，我是一只快乐的鸵鸟，我不知道 Vue 更新了，继续摸鱼爽歪歪。
- 为了下半年的 KPI，冲冲冲！把手头的 Vue 项目进行版本升级和重构



Vue3.0 更新了啥

让我总结的话，就只有两个比较重要的更新（我目前还没有完完整整的体验过新版本，有些地方可能不够完善，希望大家多多包涵，后续会整理和分享一些实践的 demo）。

一个是 Composition API，另一个是对 TypeScript 的全面支持。

团队还会出一个 Vue 2.7 的版本，给予 2.x 用户一些在 3.0 版本中被删除方法的警告，这有助于用户的平稳升级。Nuxt3 好像还在路上，但是目前看来，市面上的各大组件库还没来得及针对 Vue3.0 进行改版升级。

周边的插件如 Vue-Router、Vuex、VSCode 插件 Vetur 等都在有序的进行中。

Vue3.0 具体更新了啥

来点阳间的知识，说点人话。



config 是一个包含 Vue 应用程序全局配置的对象。可以在挂载应用程序之前修改下面列出的属性。

- **devtools 类型：** boolean **默认值：** true **如何使用：**

javascript

```
app.config.devtools = true
```

是否开启 vue-devtools 工具的检测，默认情况下开发环境是 true，生产环境下则为 false。

- **errorHandler 类型：** Function **默认值：** undefined **如何使用：**

javascript

```
app.config.errorHandler = (err, vm, info) => {  
  // info 为 Vue 在某个生命周期发生错误的信息  
}
```

为组件渲染功能和观察程序期间的未捕获错误分配处理程序。

- **globalProperties ✨ 类型：** [key: string]: any **默认值：** undefined **如何使用：**

javascript

```
app.config.globalProperties.name = '十三'  
  
app.component('c-component', {  
  mounted() {  
    console.log(this.name) // '十三'  
  }  
})
```

若是组件内也有 name 属性，则组件内的属性权限比较高。

还有一个知识点很重要，在 Vue2.x 中，我们定义一个全局属性或者方法都是如下所示：

javascript

```
Vue.prototype.$md5 = () => {}
```

- **performance 类型：** boolean **默认值：** false **如何使用：**

javascript

```
app.config.performance = true
```

将其设置为 true 可在浏览器 devtool 性能/时间线面板中启用组件初始化，编译，渲染和补丁性能跟踪。仅在开发模式和支持 Performance.mark API 的浏览器中工作。

Application API

全局改变的动作，都在 createApp 所创建的应用实例中，如下所示：

javascript

```
import { createApp } from 'vue'
const app = createApp({})
```

那么 app 下这些属性：

- **component 参数：** 第一个参数 string 类型表示组件名，第二个参数 Function 或 Object **返回值：** 只传第一个参数，返回组建。带上第二个参数则返回应用程序实例 **如何使用：**

javascript

```
import { createApp } from 'vue'
const app = createApp({})
// 注册一个 options 对象
app.component('shisan-component', {
  /* ... */
})
```

```
// 检索注册的组件
const ShiSanComponent = app.component('shisan-component')
```

- **config**(上面第一段讲过了)
- **directive** 自定义指令变化不大，还是之前那些东西，如下：

javascript

```
app.directive('my-directive', {
```




```
beforeUpdate() {},  
// 更新后  
updated() {},  
// 卸载前  
beforeUnmount() {},  
// 卸载后  
unmounted() {}  
})
```

多数全局API都没变化，还是老的 2.x 的写法居多。

Composition API

Composition API解决了什么问题？使用传统的 Vue2.x 配置方法写组件的时候问题，随着业务复杂度越来越高，代码量会不断的加大。由于相关业务的代码需要遵循option 的配置写到特定的区域，导致后续维护非常的复杂，同时代码可复用性不高，你常常会发现一个页面组件，写着写着就写到了三四百行去了。

```
401 <style lang="less">  
402   .van-uploader__input-wrapper {  
403     width: 100%;  
404   }  
405 </style>  
406
```



有没有熟悉的感觉？没错这就是老的模式带来的弊端，一直憋了这么久，也没谁了~~ 而 Composition API 解决了这个令人头疼的问题。它为我们提供了几个函数，如下所示：

- reactive
- watchEffect
- computed
- ref



首页 ▾

探索掘金

登录

reactive

javascript

```
import { reactive, computed } from 'vue'

export default {
  setup() {
    const state = reactive({
      a: 0
    })

    function increment() {
      state.a++
    }

    return {
      state,
      increment
    }
  }
}
```

reactive 相当于 Vue2.x 的 `Vue.observable()` API，经过 reactive 处理后的函数能变成响应式的数据，类似之前写模板页面时定义的 data 属性的值。

watchEffect

javascript

```
import { reactive, computed, watchEffect } from 'vue'

export default {
  setup() {
    const state = reactive({ a: 0 })

    const double = computed(() => state.a * 3)

    function increment() {
      state.count++
    }
  }
}
```



116



24



收藏





首页 ▾

探索掘金

登录

```
    return {  
      state,  
      increment  
    }  
  }  
}
```

watchEffect 被称之为副作用，立即执行传入的一个函数，并响应式追踪其依赖，并在其依赖变更时重新运行该函数。

computed

javascript

```
import { reactive, computed } from 'vue'  
  
export default {  
  setup() {  
    const state = reactive({  
      a: 0  
    })  
  
    const double = computed(() => state.a * 3)  
  
    function increment() {  
      state.a++  
    }  
  
    return {  
      double,  
      state,  
      increment  
    }  
  }  
}
```

这就比较直观了，computed 在 Vue2.x 就存在了，只不过现在使用的形式变了一下，需要被计算的属性，通过上述形式返回。

116

24

收藏



hooks

与 2.x 版本相对应的生命周期钩子

Vue2.x 的生命周期	Vue3.x 的生命周期
beforeCreate	setup()
created	setup()
beforeMount	onBeforeMount
mounted	onMounted
beforeUpdate	onBeforeUpdate
updated	onUpdated
beforeDestroy	onBeforeUnmount
destroyed	onUnmounted
errorCaptured	onErrorCaptured

Vue3.0 在 Composition API 中另外加了两个钩子，分别是 `onRenderTracked` 和 `onRenderTriggered`，两个钩子函数都接收一个 `DebuggerEvent`：

javascript

```
export default {
  onRenderTriggered(e) {
    debugger
    // 检查哪个依赖性导致组件重新渲染
  },
}
```

[探索掘金](#)[登录](#)

前面我也开玩笑的讲了三条，要么装不知道，要么赶紧学，而已经学习了 Vue 3 的朋友可以用到自己的项目中，对项目进行优化和升级。这样，在年终总结就可以加上最重要的一条：**带领团队成员完成了项目的重构，包括逻辑重构 + 语法升级，全面适配 Vue 3！打包效率提升xx倍！页面响应速度提升 xx！**

不仅如此，对于学生党或者还在找工作的同学来说，可能在准备面试时又需要多准备一些内容了，冲冲冲！

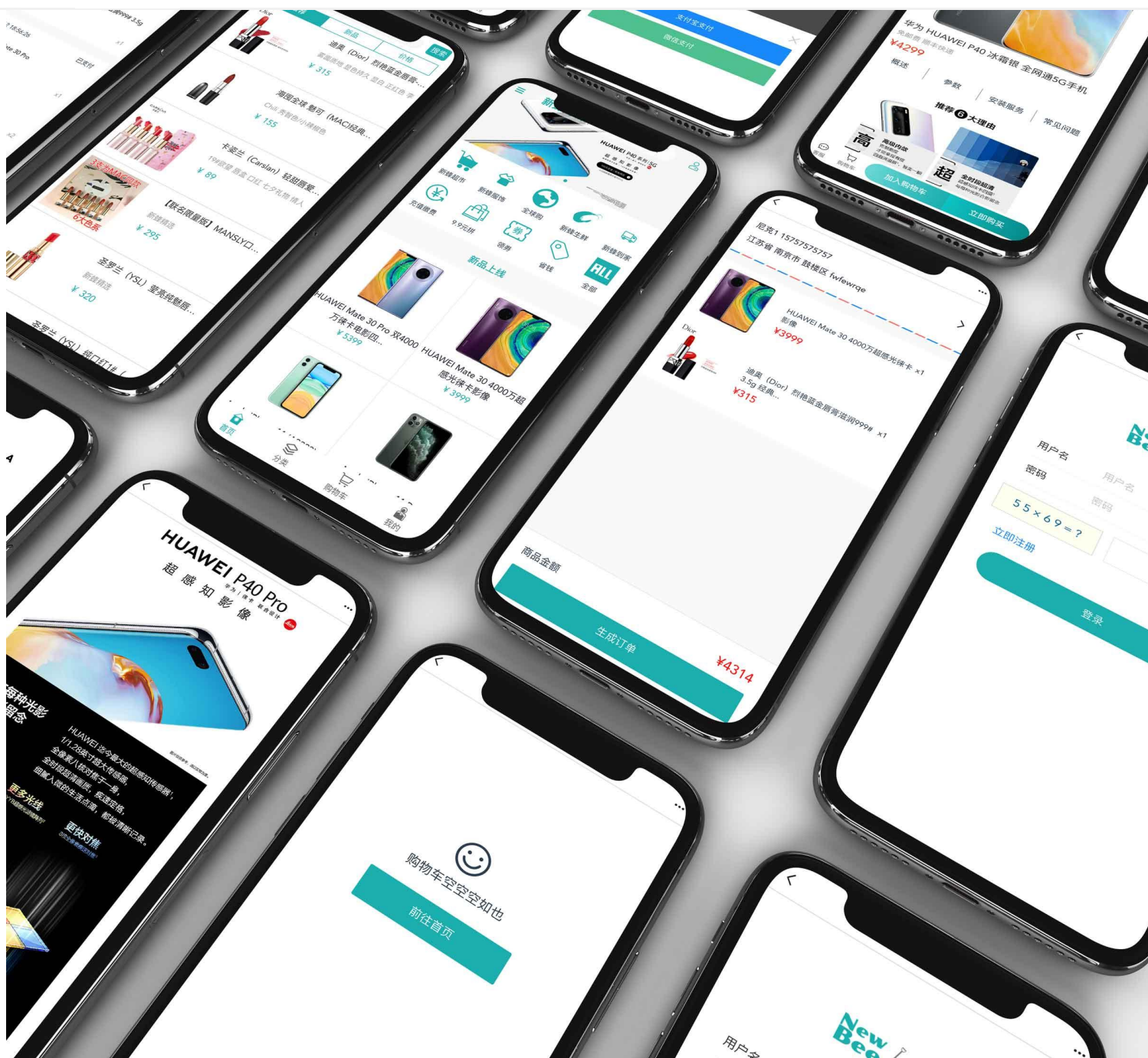
最后，我想了想我能够做些什么，首先装鸵鸟是不行的，然后不学习也是不行的，因为我上半年的时候写了一个 Vue 的商城项目并开源到 GitHub 网站上，页面效果如下所示：

 116 24 收藏



探索掘金

登录



newbee-mall 在 GitHub 和国内的码云都创建了代码仓库，如果有人访问 GitHub 比较慢的话，建议在 Gitee 上查看该项目，两个仓库会保持同步更新。

- newbee-mall in GitHub : <https://github.com/newbee-ltd/newbee-mall-vue-app>
- newbee-mall in Gitee : <https://gitee.com/newbee-ltd/newbee-mall-vue-app>

116

24

收藏



7天前



回复



我是十三 Lv3 (作者) 高级开发工程师
会的

这里，大家可以放心，我一直都回答会升级到 Vue3，并且代码依然全部开源，希望大家都去点个 star，你们越热情，我也会更有动力去重构项目到 Vue3 版本！所以，对我个人来说，其实一直在等着 Vue.js 3.0 版本的正式发布，之后我会抽时间把这个 Vue.js 实战商城项目用 Vue3 再写一下。大家可以给新蜂商城项目点个 star，star 数量越多，我也越有精神头儿去更新，哈哈哈哈哈。



这样，大家就有了 Vue3 的实战经验啦！

除注明转载/出处外，皆为作者原创，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文链接，否则保留追究法律责任的权利。

我是十三 Lv3 高级开发工程师
获得点赞 1,945 · 获得阅读 179,705

关注

 116 24 收藏