

2020 年前端开发的
未来是什么？

前端 的情况 2020

Experts 专家

Yan Cui

AWS 去服务器化领衔人物

Marek Gajda

The Software House 的首席技术官

Jessica Jordan

.cult 的开发者引领者

Luca Mezzalira

DAZN 的架构领域副总裁

Rocky Neurock

Honeypot.io 的
工程团队领导者

Tim Neutkens

Vercel 的 Next.js 负责人

Guillermo Rauch

Vercel 的首席执行官

Tomek Rudzki

Onely R&D 的领导者

Dylan Schiemann

Living Spec 的 CEO

Bartosz Skowroński

The Software House 的
设计领域领导者

Authors 编者

Patryk Mamczur

主编

Marcin Gajda

Tomasz Kajtoch

Wiktor Toporek

Andrzej Wysoczański

技术顾问

Joanna Swoboda

Magdalena Habarta

Kamil Glowiński

排版设计

PassionPenguin (霜羽 Hoarfroster)

原文译者、校对者

HumanBeingXenon (HumanBeing)

校对者

Table of Contents

目录列表

| | | |
|----|---|----|
| 01 | 开发者 Developers 4,500 名前端开发者专家们的日常开发生活 | 8 |
| 02 | 框架 Frameworks React是擂主，可谁是那个攻擂者呢？ | 12 |
| 03 | 托管 Hosting 传统的 DC 托管、云服务巨头以及聚焦于前端的托管服务 | 18 |
| 04 | JavaScript, APIs, Markup JAMStack 狂热于静态开发 | 23 |
| 05 | 微前端 Micro Frontends 在前端开发中我们真的需要来一场微服务革命嘛？ | 26 |
| 06 | 搜索引擎优化 Search Engine Optimization 看来您并不在乎搜索引擎优化，这是您应当这样做的原因 | 29 |
| 07 | 应用程序无障碍功能 Application Accessibility 让用户界面对所有用户都友好！ | 32 |

| | | |
|----|--|----|
| 08 | 开发团队 Development Teams 前端开发?这是个团队运动! | 35 |
| 09 | 设计 Design 努力追求设计师与开发者之间紧密协作 | 38 |
| 10 | 质保 Quality Assurance 将软件测试作为软件开发的奠基石 | 41 |
| 11 | 前端未来 Future of Frontend 前端开发在 2021 年的状态预测? | 45 |

这次一共有多少前端开发者
参与了这次调查？

调查问卷回答数：4500



Central and
South America
中美以及南美地区

107

Other
其他地区

1581

Eastern Europe
东欧地区

303

South
and East Asia
东亚与南亚地区

73

Middle East
中东地区

01.

开发者 Developers

4,500 名前端开发者专家们
的日常开发生活



Patryk Mamczur

调查报告主编

“ 这是一份有 4,500 名受访者参
加的调查报告,也因此,我们是
世界上规模最大的前端开发的
调查报告。

当我们开始谈及这个 2020 年的前端调查报告,我始终在脑海中坚定着拥有着这样一个目标:我一定要弄清楚一个前端开发者的日常生活究竟是什么样的。

鉴于成千上万的前端开发者如星星点点,散布世界,这个目标似乎相当不切实际,如海市蜃楼一般可望而不可及。但是诸位,前端社区又一次要为我们所惊艳了!!!

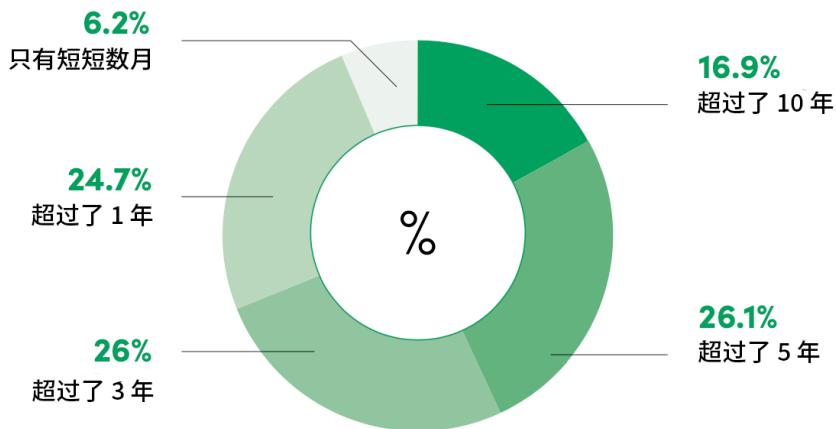
在短短几周中,整整 4,500 名前端开发者们参与到了我们的前端状况调查之中。我实际上拥有着这个一手数字,因为我确确实实就丝毫不敢松懈地,静静地端坐在屏幕之前,凝视着上升的数字,并且把握着最佳时机结束调查问卷的开放。可以说,就如同把手指放在核武器按钮上一般,我为这个数字负责。我不知道你的看法,但就我个人而言,我想,这个数字真的是出乎意料的完美了——一共有 4,500 名受访者填写了这张调查问卷。可能纵观历史,这一次的前端调查报告也会是规模最盛大的关注前端开发领域的年度报告。

总而言之,为我们成功开展了这次这个拥有四千余名前端开发者参与,告诉我们他们的日常工作,他们使用的框架以及他们希望使用的(但因为老板端着枪在背后盯着无奈不能使用的)框架,以及他们对于近来前端开发趋向的想法。这次的调查结果是真的完美,我们得到了不少令人惊讶的数据,甚至大部分的都能引人深思。所有的这些数据,完美的呈现出了前端的日常开发的现状。

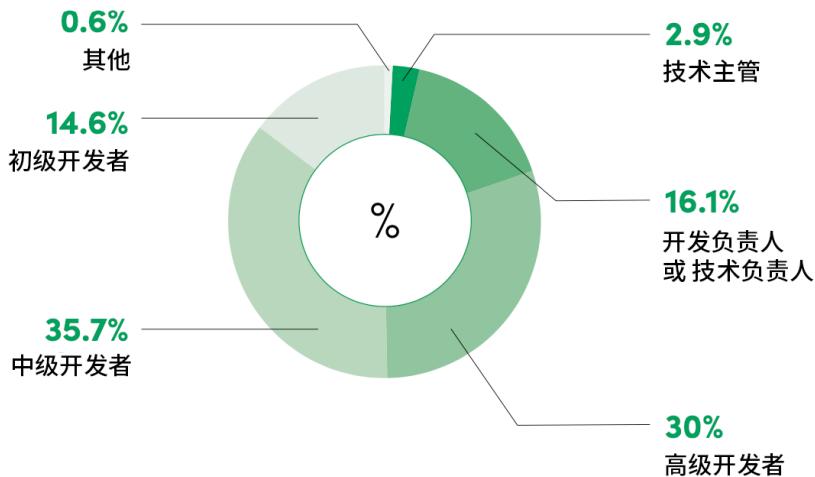
为了让所有的这些数据更直观,更生动形象,我们有幸邀请到了各个软件开发领头权威,让他们针对这些调查结果做出解释以及评论。他们凭借他们出色的专业知识和宏观视角,对《2020 年前端开发状况》进行了报告。而这正是这份最新的有关现代前端开发的报告的知识来源。

正因如此，我衷心建议您停止阅读这段夸张的简述，认认真真看一眼目录页面，选择一下你最感兴趣的话题并且找出开发者和权威的共同观点。请一定要相信我，相信这份调查报告一定不会让你失望！

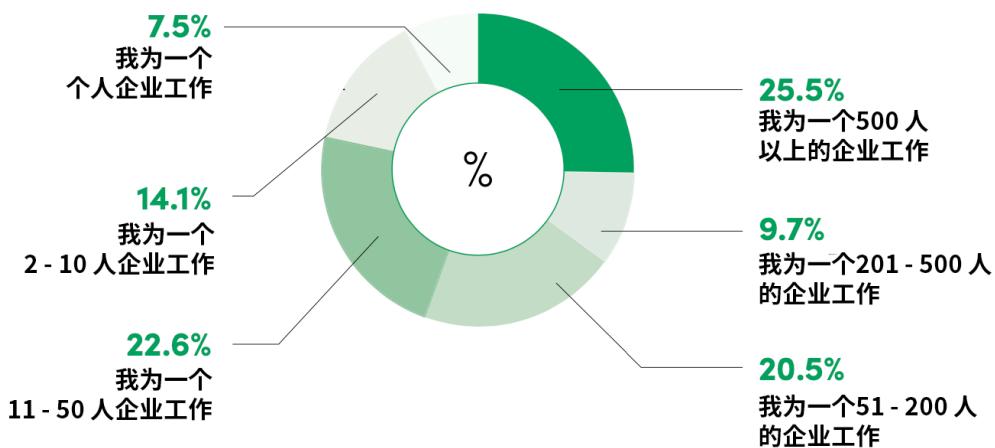
您从事前端开发工作多久了？



您觉得自己资历如何？



你现在所任职的企业的规模？



02.

框架 Frameworks

React是擂主，
可谁是那个攻擂者呢？



Dylan Schiemann

Living Spec 的 CEO、Dojo 的协助创作者

“ 现如今有一件事情非常明确：
React 在当今开发思路中占据
了主导地位

当你查看前端状况调查的结果的时候，你一定会发现一件很肯定的事情：React 主导了今天的 JavaScript 框架开发思路。但同时，看起来下一代的响应式框架正在逐渐的崛起，像是对 React 生态系统的一个倾斜的替代方案。同时他们都有与同时在逐渐收获更多使用者的 TypeScript 有着不少工作去做去协调。

对于过去的几年来，JavaScript 开发者们像是受到引力一般，被 React, Vue.js 与 Angular 这些引领开发的框架吸引着。相对来说，Angular 已经因为 Ivy 编译器的拖更而掉粉无数。类似的，对 Vue 的兴趣已经搁置在货摊中有一段时间，也只是因为 Vue 3.0 的发布版也在拖更。这都帮助了 React 以 74.2% 的占有率（调查数据中的框架使用比例）巩固了它的 JavaScript 框架主导地位。这个占有率远远超过了 Angular 与 Vue.js 两者之和。

但是，这并不意味着，React 社区会毫无顾虑的永远在世界上存在下去。现如今，一个影响深远的变化正在逐步发生在我们的世界：最近，当开发者开始抛弃 Redux 的时候，发生了巨大的变化。我们目前已经可以发现，在状态管理方面，比起 Redux (48.2%)，更多的人使用 React Context 接口和钩子 (49.6%)，超过了 Redux (48.2%) —— 当然，部分仍然在同时使用两者，但是趋势依旧极度显著。同时，作为一个小提示，当我们讨论大型 JavaScript 框架是重要的时候，我们必须不能忘记 jQuery 这个鲜有机会被谈及的，但是仍然是最广泛部署的 Web 端 JavaScript 库。

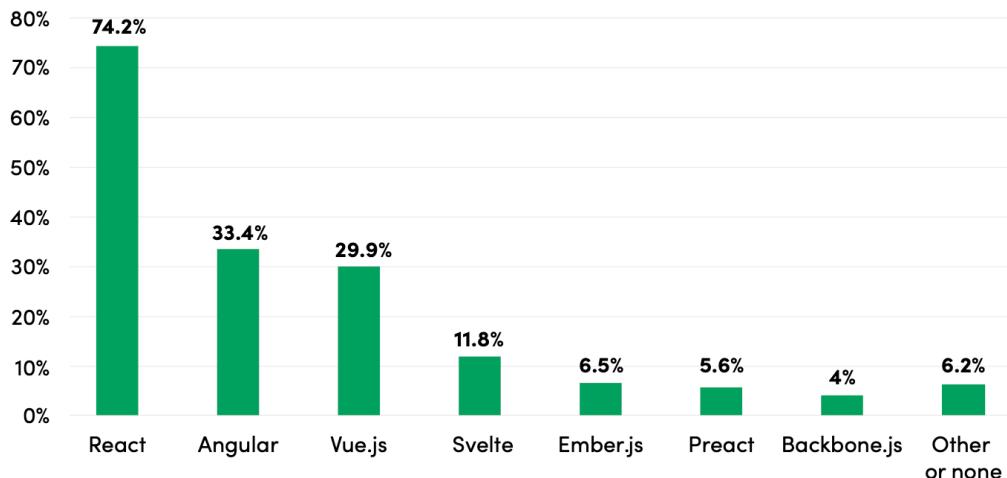
那么关于 JavaScript 的未来情况？我们可见对如 Svelte 的下一代响应式框架的极大兴趣，该框架致力于在常规 DOM 结构之上提供响应性。与此同时，另一名竞争者是 Stencil，一个专注于网络部件，以及高效编译（和 Svelte 一样）的框架。当然，Dojo 被重新注入了能量，作为一个响应式以 TypeScript 优先的框架，保证提供智能的默认功能，为用户提供拆盒即可用的更快的体验。有人认为，这些下一代框架可能更适合小型应用程序。换作构建大型应用程序时，需要额外的工作。的确，它们所提供的默认应用程序捆绑包大小要小得

多，因为它们不需要像有的框架那样需要兼容旧有的特性。此外，它们非常符合现代标准和语言功能。

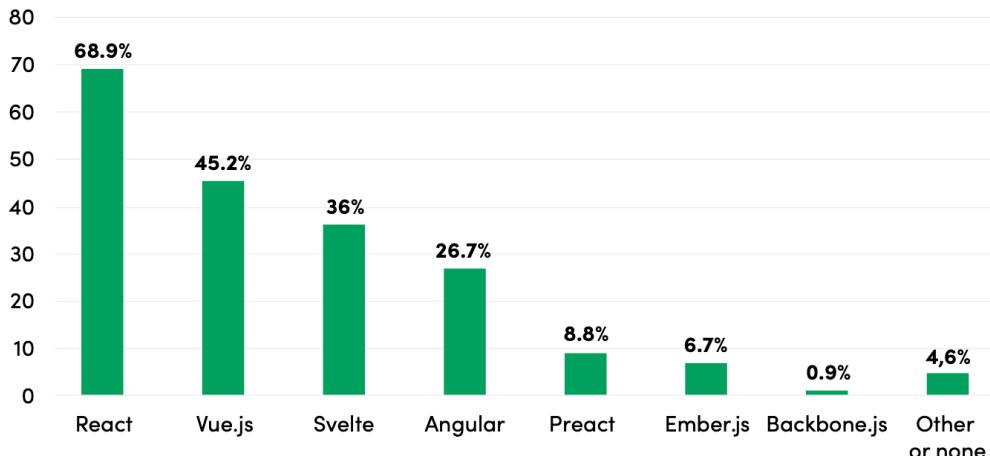
这就是为什么我们需要讨论 TypeScript 增长的重要性。目前已经有了 77.2% 的受访者使用 TypeScript 并且大多数的人与 JavaScript 相比，都对之更感兴趣。说实在的，不少框架正在提升他们对 TypeScript 的支持，并且不少项目开始使用 TypeScript 完全重写，同样适用于那些已经发布了的框架（例如 React 和 Angular）以及不少跨时代框架（尤其是 Stencil 和 Dojo）。

随着这些变化的发生，我真的很期待 JavaScript 领域将会如何发展。因为始终有一件事情是肯定的：React 是目前的当世无双的统治者，并且已经有不少有机会夺取 React 势力的竞争者正在为夺取框架领域的宝座摩拳擦掌。

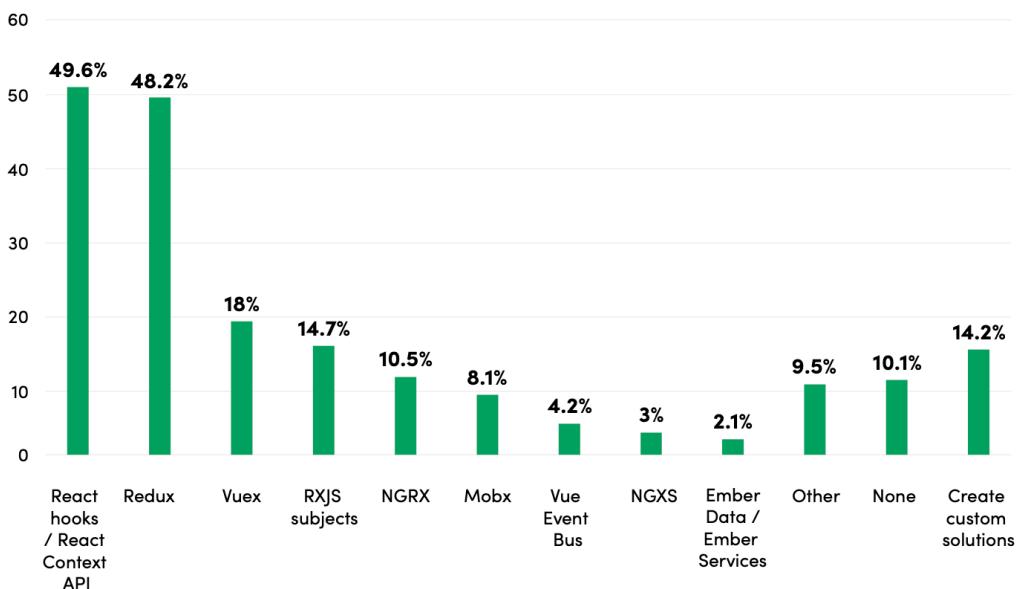
去年您都使用了以下哪些框架？



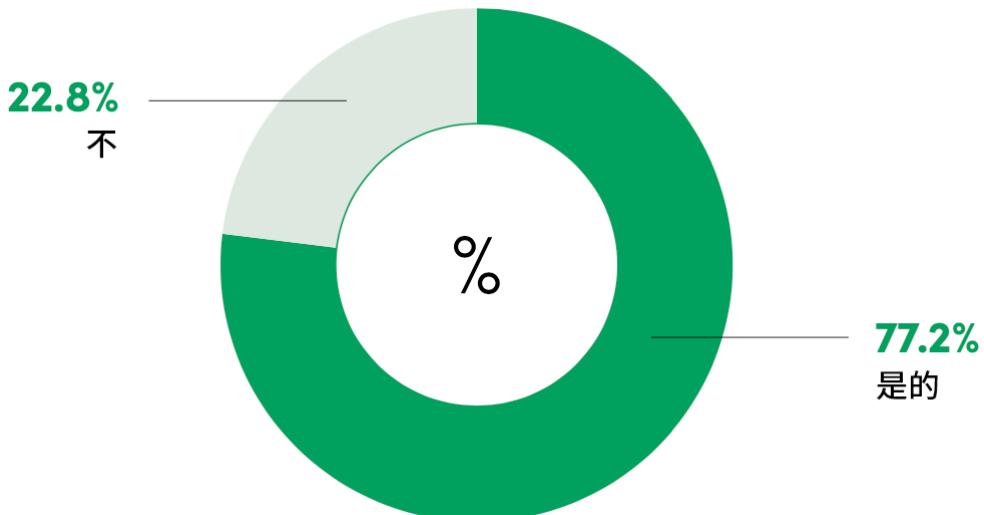
以下哪些框架您更愿意继续使用或者想要在未来学习？



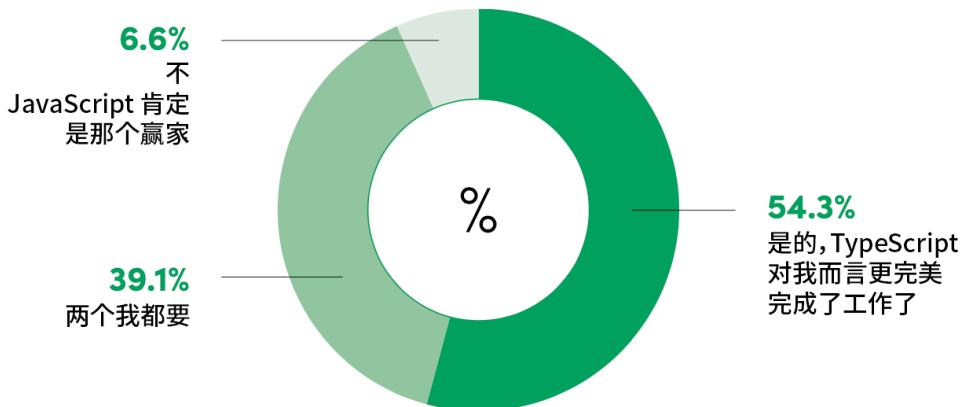
您会使用什么解决方案来进行状态管理？



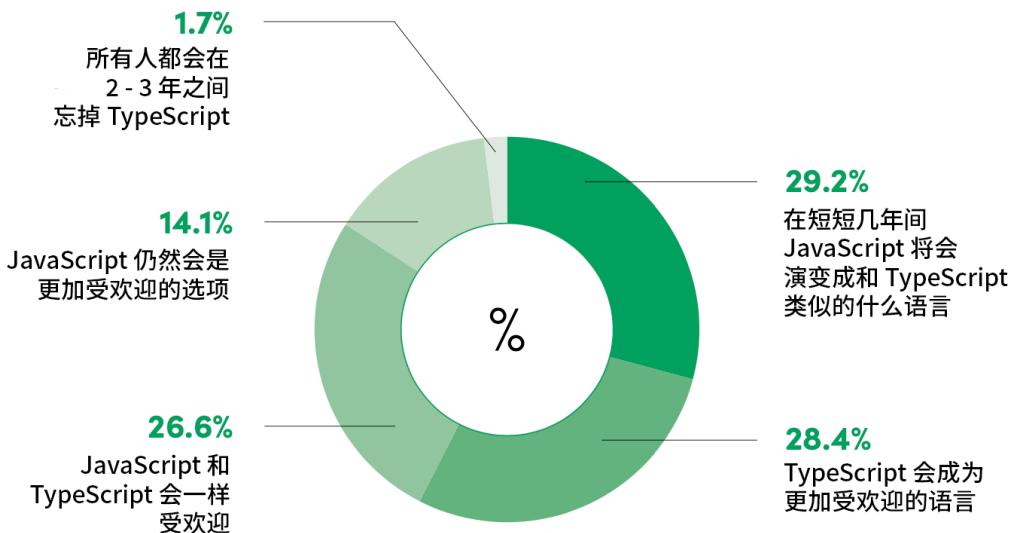
去年您是否使用了 TypeScript ?



您觉得 TypeScript 比 JavaScript 更好吗?



您觉得 TypeScript 的未来是什么样的？



03.

托管 Hosting

传统的 DC, 云服务巨头
以及聚焦前端的托管



Yan Cui

AWS 去服务器化领域的英雄人物以及 [独立顾问](#)

“ 前端开发的未来正需要像是
Netlify 或是 Vercel 那样为前
端开发团队提供简单使用的功
能的平台

2020 年前端状况调查的结果显示了不少有意思的数据，尤其当主题是托管服务的时候数据十分有趣。当你看到那些数字的时候，你会发现它们包含了所有的内容：从传统 DC 服务到云服务提供者到现在新生的 Netlify 和 Vercel。

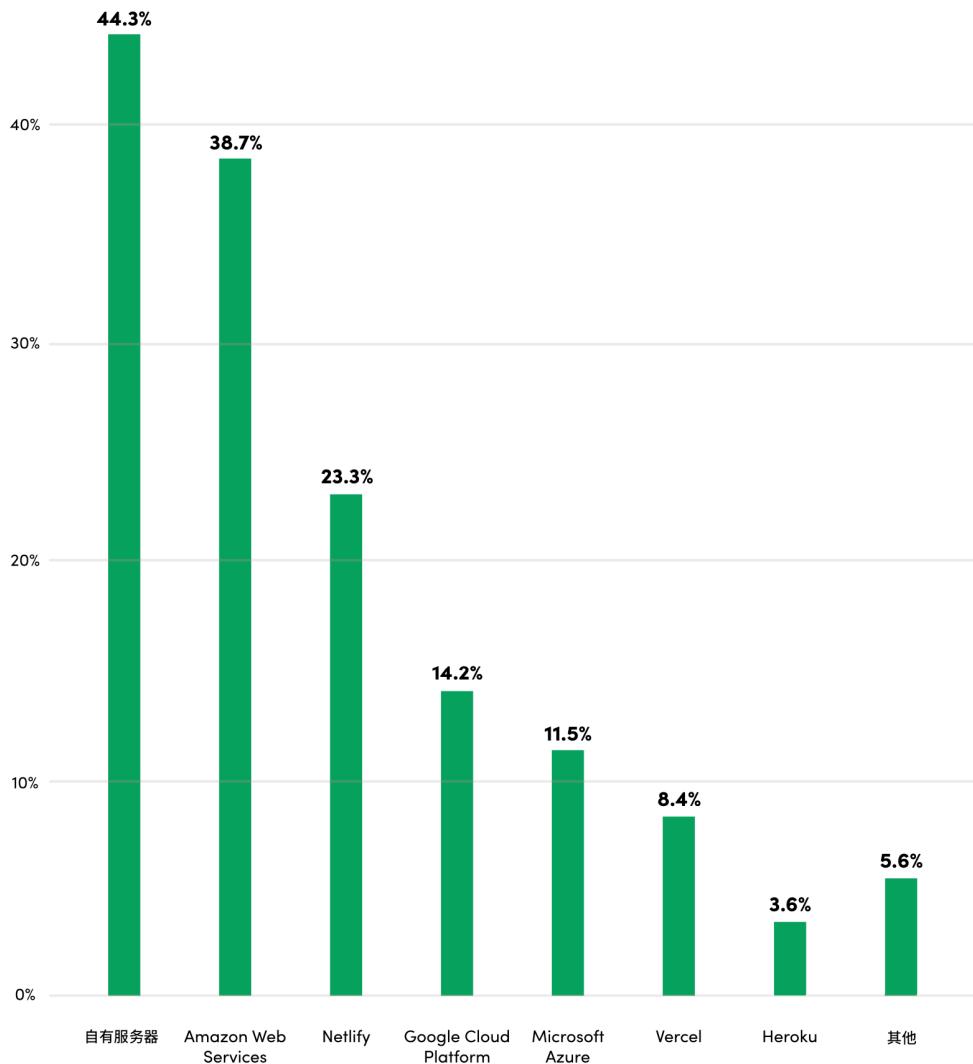
对于我而言，从这些数据中我获得的最大的惊喜莫过于 44.3% 的受访者仍然将他们的应用程序部署到自己的 Web 服务器上。再重复一遍，这是一个备忘小纸条，告诉我们现在对于传统 DC 服务提供者来说仍然有一个巨大的市场，并且对于公有云服务来说仍然有很大的发展潜力。

说实在的，在所有的云服务提供商中，亚马逊 Web 服务是最受欢迎的部署对象（38.7%），但这并不够令我惊喜。如果我们将目光转向另外一个方面：AWS 的分享功能的占有率远大于 Google 云（14.2%）和 Microsoft Azure（11.5%）占有率之和。

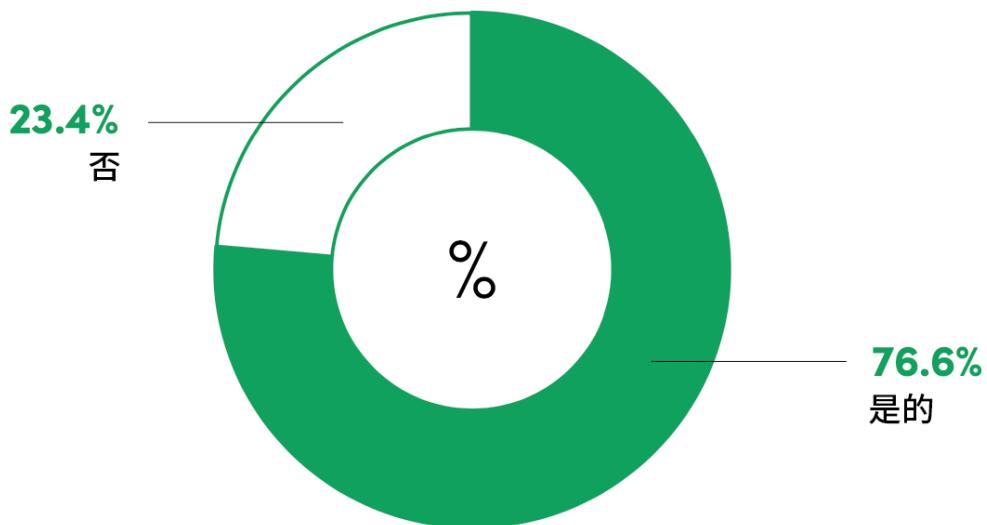
同时，Netlify 打入市场的比例如此之大（23.3%）甚至大于 GCP 和 Azure，而这很好的证明了 Netlify 的意义之大。越来越多的人认为，前端开发的未来需要诸如 Netlify 和 Vercel 之类的平台，因为这些平台可以为前端团队提供一个易于使用的、功能强大的后端基础架构抽象。

AWS 目前的继续推动 Amplify 的行为也可以告诉我们他们在其中看到了潜力与市场。另一个角度而言，我们不禁觉得即便在他们受够了 Firebase 多年之后 GCP 仍然失去了一个机遇，也失败于把握将其发展为市场的主导者的机会。

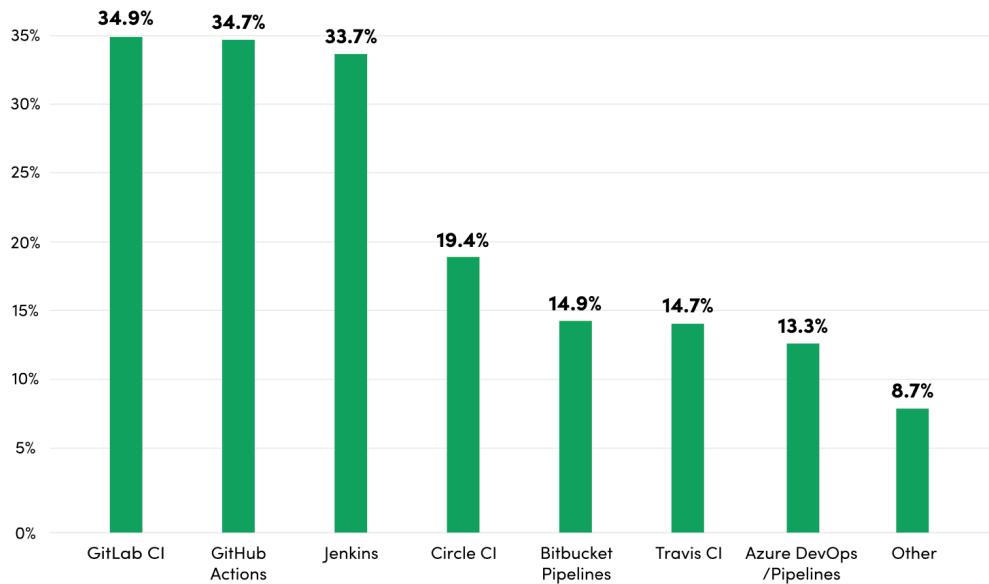
你平时通常将你的应用程序部署到？



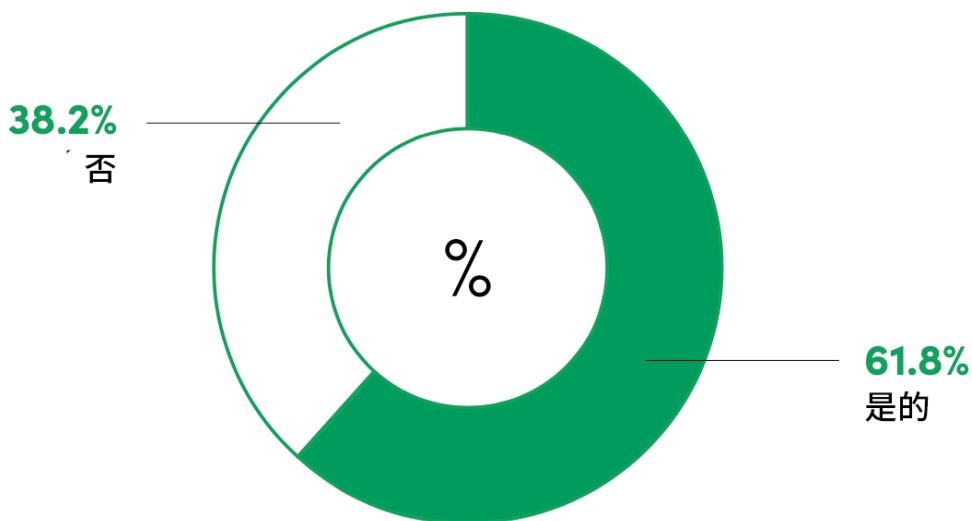
你使用 CI 吗？



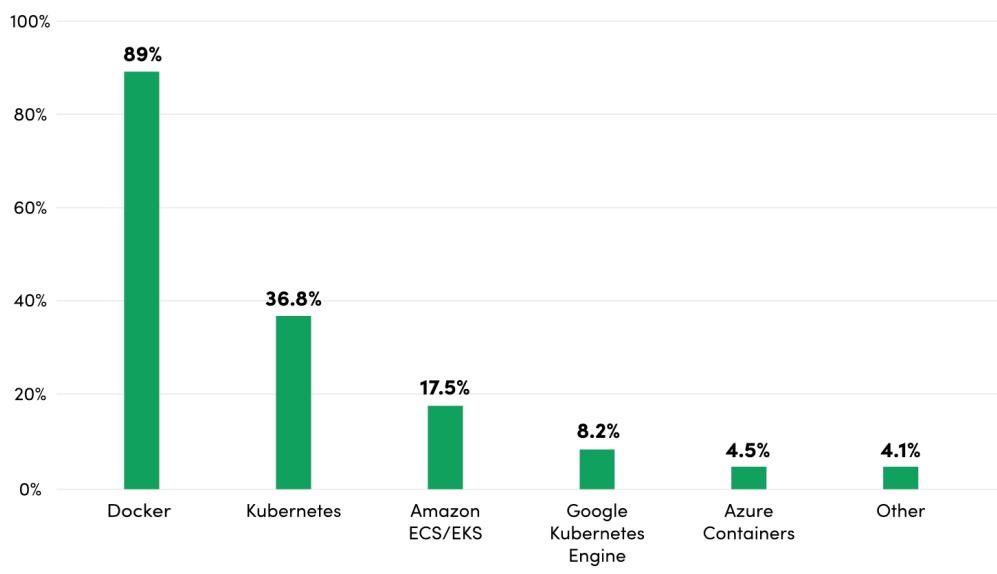
你去年使用过什么 CI ?



你使用过容器化解决方案吗？



你使用过什么容器化管理工具？



04.

JavaScript API Markup JAM

狂热于
静态开发



Tim Neutkens

Vercel Next.js 负责人

“让我们先搞个大的：我们预计在未来几个月中，会有更多开发者构建 JAMStack 网站”

说起来，大约三分之一的受访者近期已构建过一个 JAMStack (JavaScript、API 以及 Markup) 网站。同时，让我个人非常开心的是，大约一半的人已经使用过 Next.js，一款 Vercel 构建的 React 框架。因此，请让我们先爆料个的巨大新闻吧：我们预估在未来几个月中会有更多的开发者构建 JAMStack 网站。

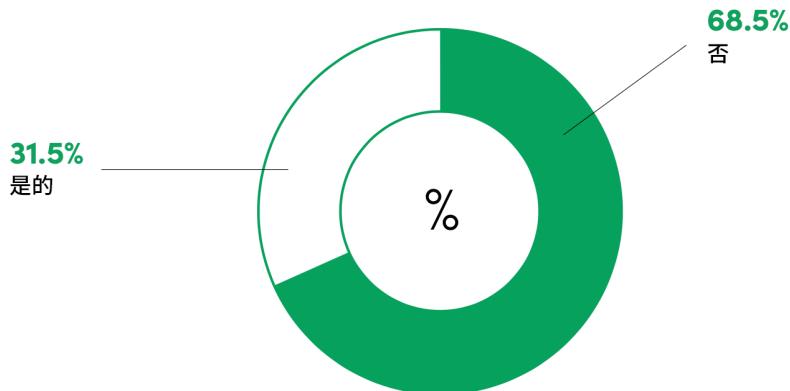
对我来说，JAMStack 最大的作用就是它能够帮我们做得更少，完成得更多。通过使用这个完美的 JAMStack，当有了请求的时候它不必要渲染所有的页面内容，它是一个服务端渲染策略——你的服务器会在请求回应之前预先渲染一个页面(静态生成)，也因此，各个 CDN 之间是可以完美地进行资源分享以优化性能 h ，获取更高的可能性以及更低的消耗，以及不会造成任何过度维护。

此外，JAMStack 框架是在静态一边以及动态的灵活自适应的一场巨变！举个例子，Next.js 允许你在产品 buildy 以后还能继续静态地生成额外的页面或者重新生成现有的页面(增量静态生成)。就算你的应用有上百万个页面，初始的 build 也会给这些页面增量生成的能力。

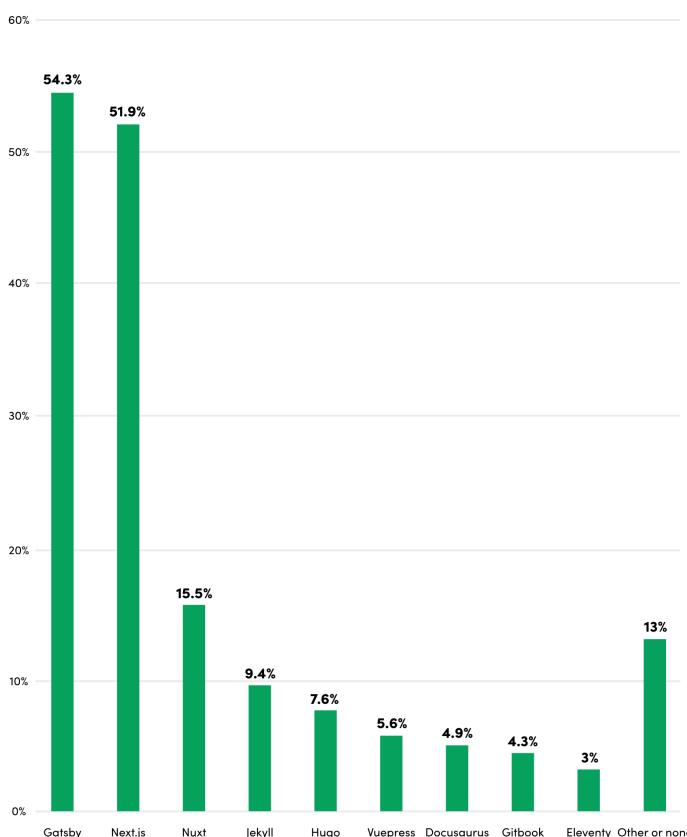
可重用的 API (JAM 的 A) 也是一个趋势：市场现在已经有了不少无头的 CMS、无头的 E 商城，无头的身份辨识等各种各样的功能。无需质疑，现在的各种框架也在向这些趋势靠拢。就我的经验而言，Next.js 拥有预览模式的功能，让我们在预览无头 CMS 的时候可以同时有条件地静态生成页面。

我现在真的对 JAMStack 的未来及其盼待。看到受访者正在使用各种 JAMStack 解决方案真是太棒了！这体现了开发者们正在体验不同的解决方案思路，并且推动 JAMStack 向一个简单而高性能的 Web 端框架发展！

你构建过 JAMstack 网站吗？



你去年使用了什么静态网页生成器？



05.

微前端 Micro Frontends

我们真的需要在前端来一场微服务革命吗？



Luca Mezzalira

DAZN 企业架构副总裁、[Building Micro-Frontends](#) 作者

“ 虽说现在过早, 尚需学习更多,
但我坚信微前端会不断发展, 逐
渐变得成熟, 就像微服务那样

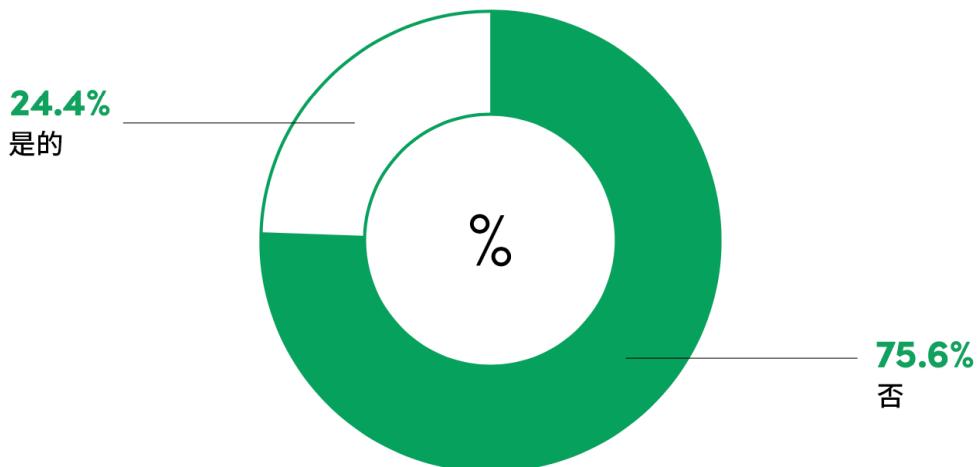
说实在的, 看到今天的人们如此热情拥抱微前端架构, 实在是太兴奋了! 我们早已知道世界上已经有不少企业正在使用微前端架构, 就例如 American Express、DAZN、宜家、Spotify 以及星巴克。现在, 我们拥有了前端状态调查的数据, 我们知道了已经有了 1/4 的前端开发者们用过了微前端服务。

我想 Web 组件是一个极佳的、入门级别的解决方案 —— 对于刚刚开始他们微前端开发旅程的人们来说尤其如此, 并且我们的调查结果似乎也证实了这一观点。换个方向考虑一下, 事实上已经有不少框架支持了服务端的渲染, 例如 Holocron、Podium 以及 Ara Framework, 同时也支持客户端的组装(例如 Module Federation 或 Single SPA)。但是, 你应该记住, 虽然这些框架是微前端社区的良好补充, 但是你应当仔细挑选, 始终检视你操作的上下文。

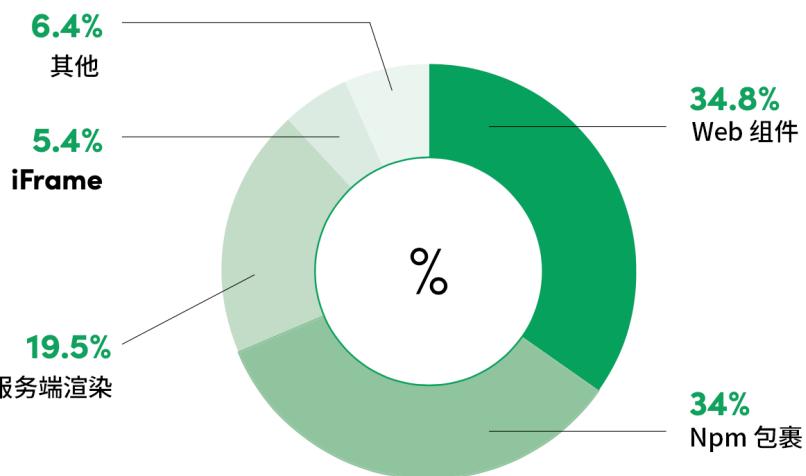
令我高兴的是, 只有 20% 的受访者认同微前端会在三年内消失的论断(见 [v11 章 前端的未来](#))。我同样相信未来会是微前端的天下 —— 他们一定会让微前端不断发展, 变得成熟, 就像前几年的微服务那样。事实上, TC 39 已经有了不少有趣的步伐, Realms 项目已经在 stage 2, 这将会为微前端开拓新的应用场景。

微前端并不是灵丹妙药, 但它绝对是其他架构(比如服务端渲染、JAMStack 以及单网页应用)的不错补充。但因为时日过早, 我们还有不少东西要完成, 并且需要更多的应用和找到更多东西去学习。然而, 我笃信, 这个架构可以为调整项目和团队带来好处, 只要用在合适的上下文。

你使用过微前端服务吗？



你怎么组合你的微前端服务？



06.

搜索引擎优化

Search Engine Optimization

看起来你并不在乎 SEO,
让我来告诉你为什么你需要 SEO 吧



Luca Mezzalira
Onely 研究与开发负责人

“当你明白了 Google 的观点，构建一个以用户为中心和对机器友好的网站一点都不难

由搜索引擎到来的流量对于任何在线生意来说都至关重要。依据 Wolfgang Digital 的 KPI Report 2020, 原始搜索制造了大约 43% 的流量, 远大于直接访问的流量以及付费搜索流量之和。同样的, 根据我们的前端状况 2020 的调查数据, 大约 52% 的开发者并不在乎 SEO。

在这里我并非要责怪他们, 我猜想你们当中的部分人是开发保密的内部应用程序, 不需要甚至不能够在搜索引擎可见。但是, 在其他的情况下, 如果你想要一个网站成功的在 Google 上完美地显示出来, 你需要进行搜索引擎优化。

因为 SEO 专家们不常常使用开发者的语言, 实现 SEO 有些困难。请允许我给予你帮助 —— 首先, 你应该总是确保 Google 可以正确的渲染你的页面的 JavaScript, 举个例子, 你可能意外的拒绝了搜索引擎对部分脚本对的访问请求 (在 robots.txt 中), 或者使用 JavaScript 并不被 Googlebot 所接受。我推介你们使用对移动端友好的测试或者 URL 倾测工具 —— 它们是免费的, 容易使用的由 Google 开发的工具。通过使用它们, 倾测 DOM 去确保你的所有的重要部分都能够被正确地由 Google 渲染。

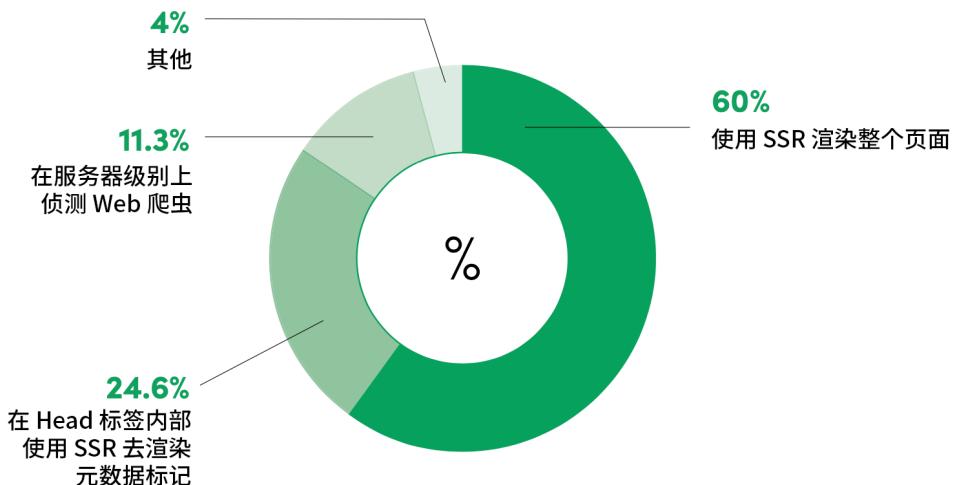
同样的, 根据调查数据, 超过 11% 的开发者使用动态渲染功能 (本质上通过判断是否是搜索引擎的爬虫来返回一个特殊的一个静态的页面)。谷歌称之为一个针对爬虫的解决方案。这个方案是有风险的, 因为动态渲染在部分情况下会失效, 导致这些页面的原始流量掉落到 0. 因此, 你总是需要保证你已经通过了测试并确保动态渲染正常工作。

SEO 对大多数的生意都极其重要, 并且渲染它是 SEO 的部分工作。你需要将注意力集中在使用恰当的 HTML 标签并在你选择服务端和客户端和动态的渲染之间构建一个有逻辑的网页结构。当你理解了 Google 的观点, 你会发现构建一个以用户为中心和对机器友好的网站一点都不难。

你在乎 SEO 吗？



你怎样实现 SEO 呢？



07.

应用程序无障碍设计

Application Accessibility

让用户界面对
每一个用户都友好



Rocky Neurock

Honeypot.io 开发团队负责人

“我相信增强无障碍能力和提供更优体验的关键在乎让我们这些明白方法的人真正地向同行们讲授无障碍设计的好处。

“不要割裂互联网！”我的朋友 Melanie Sumner 总是劝告我。

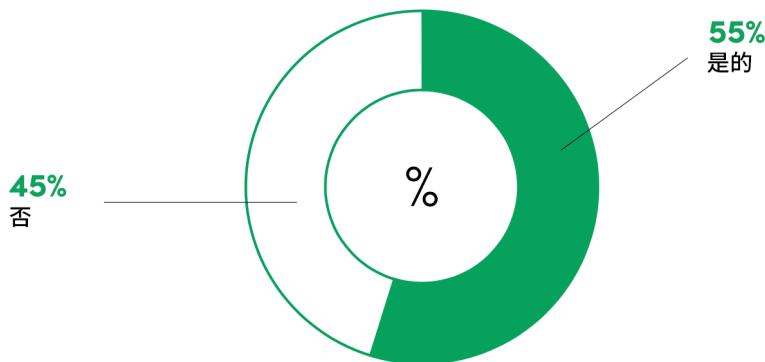
作为一名开发者，我们总在我们工作的最后一步才去考虑提升应用程序的无障碍能力。要逃离这种情况，我们需要在我们的思维中有一个转变。无障碍功能不会自己走过来，而是我们必须成为同辈们最好的老师并且教授他们无障碍设计，这样互联网才会为所有人服务。

这个主题是我心中最接近也是最重要的，不仅仅因为我真的关心用户体验，但是也因为我遭受着视力削弱的病症。我可以很肯定地说，互联网并不完美的为我工作。小的文字和低对比度的颜色最大程度的影响我的日常生活，但是我仍然每日例行遭遇别的令我沮丧的糟糕的设计。举个例子，对于非原生组件，例如重设计的选择列表或者单选框，他们可能会让用户更沮丧——因为他们并没有完美的完成任务。

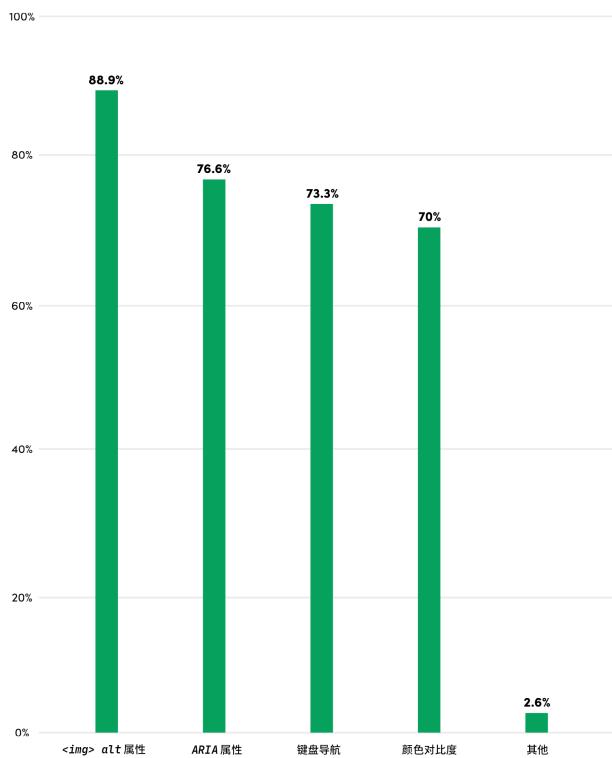
首先，请先考虑一下这些体验对那些喜欢在键盘旁或手机前过日子的用户来说是多么的令人心烦！然后，考虑一下用户使用着无障碍设备，他们的体验会更加糟糕。看到开发者们对基础 WCAG 网络内容无障碍指南 的理解熟悉与使用程度，我由衷感到开心。在未来，我们应该尝试去找出有多少人进行过无障碍测试。而就目前而言，自动化测试无障碍的工具的数量的逐步提升让我想到，如果这个数字与觉得有责任负责无障碍功能的开发者们有关，你那该会多好啊！

我真的认为，增强无障碍能力和提供更好的体验的关键在乎让我们这些明白方法的人真正地向同行讲授无障碍设计的好处。如果我们可以将我们的时间通过自动化测试来节省下来，那将会更棒啊！

你在乎无障碍设计吗？



你如何实现应用程序无障碍？



08.

开发团队

Development Teams

你在谈及前端开发?
那是一个团队的共同竞赛



Guillermo Rauch
[Vercel](#) CEO

“近来前端架构以及部署基础的趋势已经影响了前端开发者们与团队成员协作的方式

前端开发是一项团队间的竞赛——调查数据直观清晰地显示了 92% 的受访者在去年以一名团队成员的身份参与了前端开发。但是，近来在前端架构和部署基础上的趋势已经影响了前端开发者们与团队成员之间的协作，就比如说在 Vercel，我们就有眼前的例子。

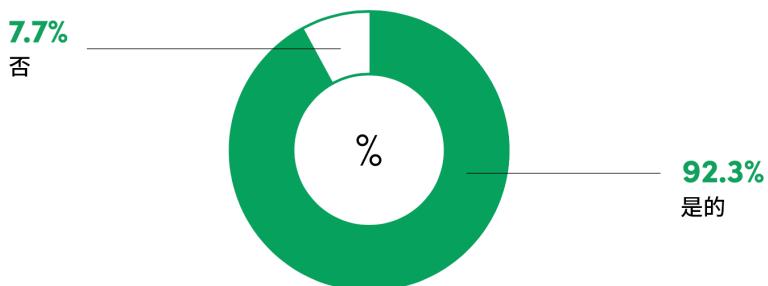
随着新型前端架构例如 JAMstack 的不断崛起，前端开发者现在可以将他们的前端项目简单部署，与后端独立。他们不再需要等待全套的后端测试系列完成运行，促进了快速迭代的发展。还有，现成的后端 API 例如无头 CMS、身份验证服务等正在崛起，并且能够简简单单地插入到你的前端项目中。这让后端开发者们可以关注在开发生意所需要的独特的 API。

另一个变化时设计师和项目所有者之间的关系。因为 JAMstack 应用程序可以很简单且便宜地部署到 CDN 的各个分支上，分配一个独特的预览 URL 给每一个分支每一个记录已经走向现实。就拿 Vercel 而言，现在我们的设计师和项目拥有者可以简单地点击预览链接去查看前端开发者们的更改是否符合预期，远比分享截图或 GIF 更有效。

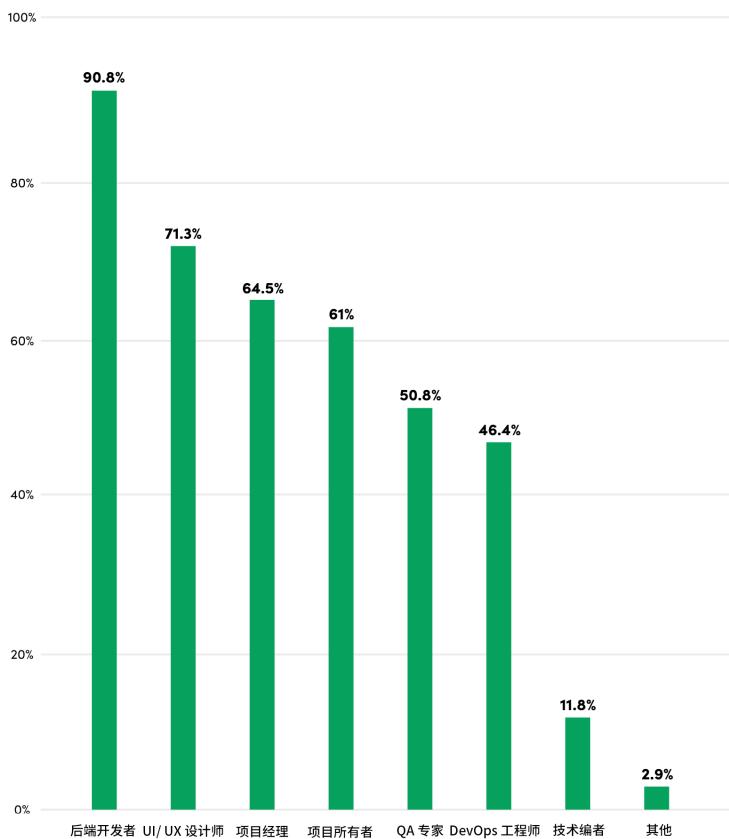
还有最后一个，软件测试。随着 Puppeteer 的发布，Chrome 的与无须服务的计算，端对端的测试相组合的无头版本浏览器现在运行地更快也更便宜。举个例子，你可以拥有服务像是 Checkly 去运行 Puppeteer 测试——由 QA 专家编写的，应对专门的预览链接的测试。并且随着 Vercel 和其他前端部署平台的崛起，他们可以简简单单地为你完成其他冗杂的苦难的工作，意味着 DevOps 工程师可以花费更少的时间去支持前端开发者们。

综上所述吧，我们实在是对前端架构以及部署基础导致的团队协作的变化很是激动。我们现在正在盼待着这个领域未来的更多的新方法与革新。

你去年以一名团队成员工作过吗？



你的开发团队的成员有下列中的哪些？



09.

设计 Design

努力追求设计师与开发者
之间紧密协作



Bartosz Skowroński

The Software House 设计负责人

“ 随着产品设计师的加入，软件公司越来越关注于创建与客户业务目标相结合的策略和产品。

图形设计师加入软件开发团队已经不是什么新鲜事了，犹依稀记得十年前或者可能更早就已经有了这个话题，但是现在我们终于来到了设计师与开发人员紧密协作并非什么新鲜而是标准的时代了，并且，我们现在还多了不少极棒的，能帮助这些协作变得更出色的工具。

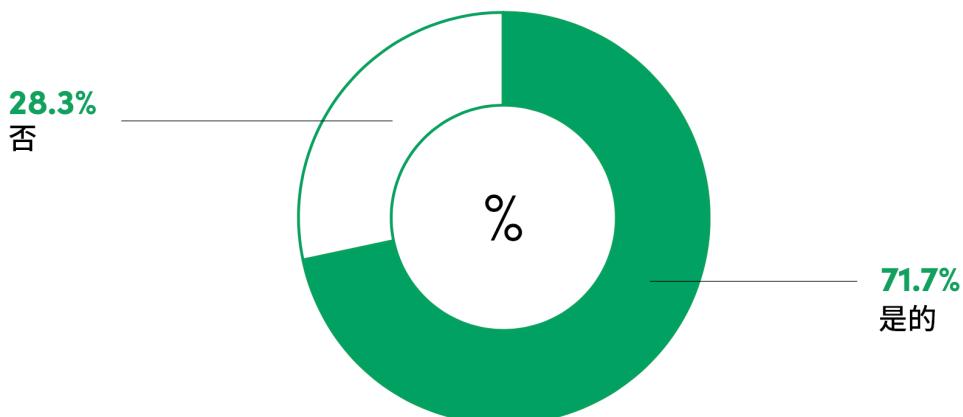
在软件开发中，设计类型的最基本分类可能是 UX（优化可能的用户体验）、UI（构建正确的用户界面）和产品设计（考虑客户业务以及业务目标）。现如今，软件开发公司雇佣有两种设计师已经成为了一种标准了——以用户体验为核心的 UI / UX 设计师以及以业务为中心的产品设计师。

而产品设计师的诞生让我尤其开心：这意味着作为软件公司，我们更加关心客户的需求，着力创建于业务相关的产品和策略。并且，客户们也开始意识到了这一点变化——全世界超过 70% 的开发团队至少配备了一名设计师（参阅 [第 8 章 开发团队](#)）。现如今，客户们无需再雇佣额外的自由设计师，而是可以选择一个团队，让开发人员和设计师（以及项目经理、软件测试人员和其他人员）更加紧密地协作。

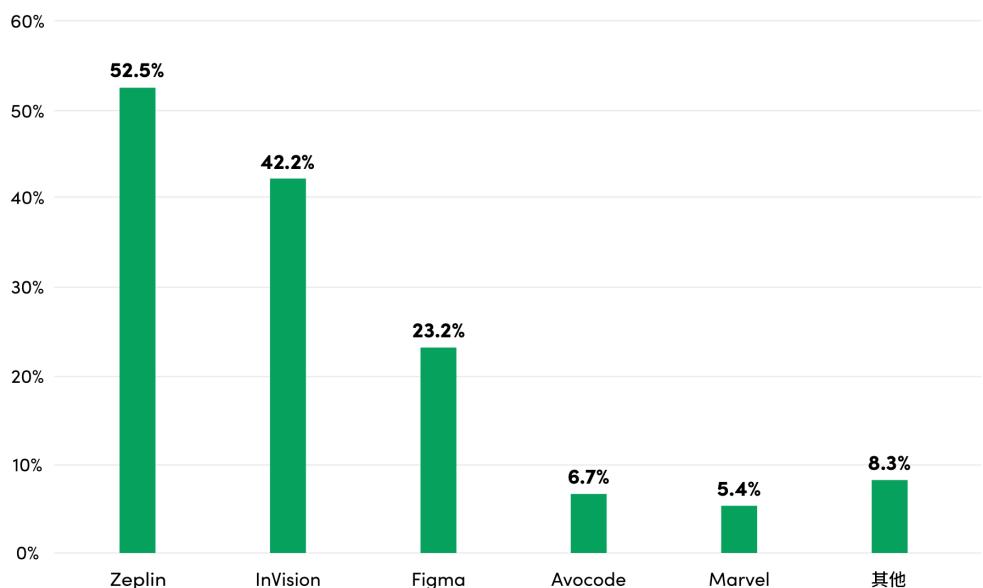
为了让协作更有成效，我们也需要更好的工具。多年来，设计师们一直在使用 Adobe Photoshop 之类的软件——挺难找到舍和软件开发业务的设计师需求的工具。幸运的是，现如今已经涌现了大量的这方面的工具，如 Figma、Invision、Sketch 和 Zeplin 等。这些工具让一切工作简化了：创建矢量图形、与其他设计师协作、设计提交给前端开发人员。有 71.7% 的开发团队使用这样的工具，实在是太棒了！

尽管设计师和前端开发人员之间的关系可能并非那么好，但是我相信，随着设计师的不断加入，以及更好的设计、提交设计工具的出现，未来可期！

你去年在与设计师工作的时候使用过任何一种交接设计的工具吗？



哪一个交接设计的工具你使用过呢



10.

QA Quality Assurance

软件测试是
软件开发的基石



Jessica Jordan

.cult 开发者倡导者

“软件测试和软件开发是不可分割的两个主题”

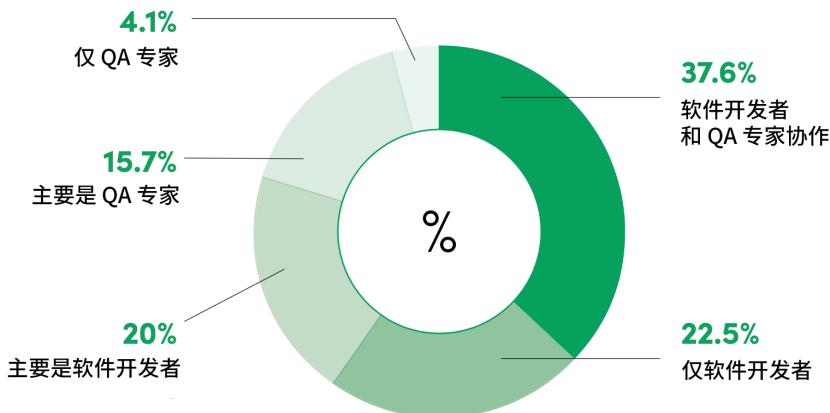
如今，客户端上已经实现了越来越多的数字铲平功能，令我们软件工程师和质量检测专家必须将测试作为我们的工作流程的一部分，以开发维护和拓展 JavaScript 应用程序。令人开心的事，多达 80% 的前端开发人员已经在执行软件测试，并且这个数字在近年来还在不断增加。

幸运的事，JavaScript 的生态系统为我们提供了大量的工具构建完美的测试套件，并为我们的应用程序提供足够的代码覆盖率的测试。

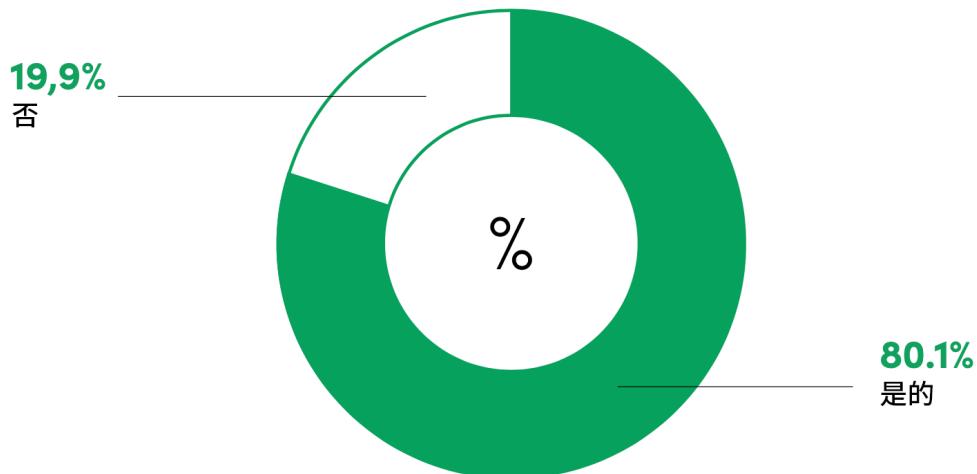
近年来，我们发现 JavaScript 测试的生态系统中的趋势是向简单方向迭代，借助人机工程学改善开发人员的开发环境，与其他测试解决方案集成以及其他许多方面。在 .cult，许多的软件项目，包括我们的强大的应用程序 Honeypot.io，都经过自动测试，并随着代码库的增长而获得更高的测试覆盖率。此外，我们的质保团队会手动测试并验证功能和设计是否完全满足功能需求，这是我们的发布产品的工作流程的重要组成部分，更好的确保了用户体验的出色。

尽管我们的团队在开发平台时候分配了更多时间进行手动自动测试，但我们一次又一次地发现软件测试是对我们的数字产品的稳定性的绝佳且必要的投资，最终影响到了整体工作流程的效率的提升。我们使用 Capybara、RSpec、Ember CLI 和 QUnit 等单元、集成和端对端的测试。当然，还有更多的解决方案供我们选择。在 .cult，我们坚信用于测试的工具的生态系统的持续发展将在短期内促使我们能够通过自动化工具覆盖更大范围的产品的开发。为什么我们要这么做？因为我们确信，软件测试和前端软件开发密不可分。

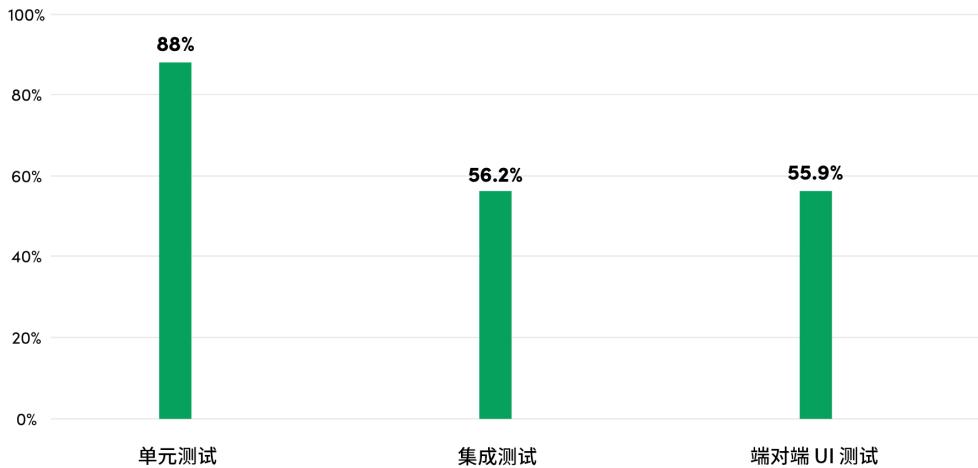
在你们团队中，谁负责软件测试



你在去年自己进行过软件测试吗？



你自己进行过什么类型的软件测试？



11.

前端的未来

Frontend Future

2021 前端状况？



Marek Gajda

The Software House CTO

“在前端开发社区，喜爱与厌恶之间的界限非常小”

我对你你说我对前端开发的调查结果以及最近的变化感到惊讶吗?并非如此。

我对这些变化发生之快感到惊讶吗?当然如此。

这就是为什么预测前端的未来并非一件容易的事。当你查看前端开发的状态是,你会发现一些完善的技术、工具以及优良的实践 —— 显而易见的选择 —— 我们应该向这些靠拢。让我们以 JavaScript 框架为例,你会发现 React 的使用人数比 Angular 和 Vue.js 的人多(参阅 [第 2 章 框架](#)),你会意识到 React 已经有了一个扎实的声誉,并且它很可能难以继续发展。

但是,在前端开发社区中,喜爱与厌恶之间的界限非常小,最好的例子就是 Redux。一两年前,使用 Redux 是“非常正常”的选择。但是前端开发者现在因为使用 Redux 导致一堆问题的发生而跳开 Redux 投奔 React Hook。在 2020 年的夏季,使用 Hook 的人数已经超过了 Redux(参阅 [第 2 章 框架](#)),并且多达 34% 的前端开发人员认为 Redux 会在 3 年后离开我们的视野。

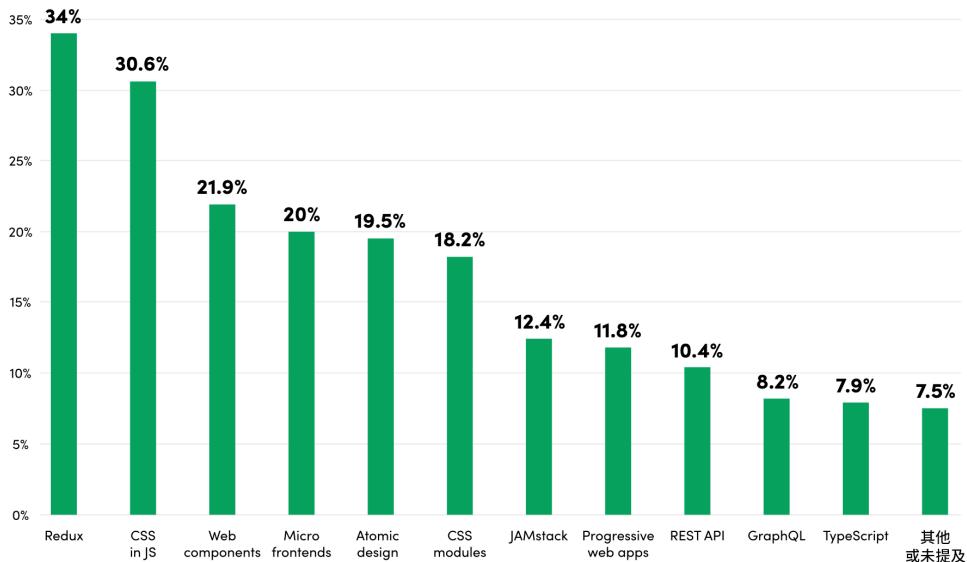
同样的,前端开发的世界也变得越来越复杂,一年两千,像是 CI 和容器化的解决方案被视为后端,但是现在前端开发者们意识到了它们的好处,有 77% 的前端开发者使用了 CI 而 62% 使用了容器化(参阅 [第 3 章 托管](#)),让这些解决方案成为前端开发的标准。

那么,前端开发的状态在未来 12 个月内会有什么变化呢?Svelte 会成为最热门的 3 种框架之一吗?微前端会成熟吗?

没有人能够给出肯定的答案,但我可以肯定,有些变化会很快发生,并且令我们惊讶!

让我们在《2021 年前端状态》报告中再见面!

以下哪一趋势 / 解决方案会在 3 年内消失?



软件开发已经 开始向着完美变化

我们给予那些理解这个转变，
并想充分利用它的技术经理以帮助



现代化
实时前端



技术栈迁移



云端迁移
去服务化



微服务架构

看看我们能够怎么帮你：