

阿里云杜欢：云上Serverless开发能力将成为前端的“金手指”

原创

阿里云栖号  于 2020-01-06 12:42:56 发布  1666  收藏 1

文章标签：[云上Serverless](#) [前端工程化](#)

版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

本文链接：<https://blog.csdn.net/yunqiinsight/article/details/103854564>

版权

云 + 端模式成为当前前端开发的新风向，由此而来的 Serverless 正帮助前端工程师提升开发能力和效率。InfoQ 记者在近日有幸在 2019ArchSummit 全球架构师峰会北京站采访到了阿里高级前端技术专家杜欢（风驰），他为我们详细梳理了阿里这两年在前端工程化的过程中，使用云 + 端的 Serverless 来探索前端演进过程的经验和体会。

InfoQ: 杜老师，您好！请您介绍一下您的从业经历，以及目前在阿里云战略 & 合作部负责的工作。

杜欢（风驰）：目前我在阿里云战略合作部，负责阿里云的开发者业务，更多的是在考虑怎么在云的时代帮助整个广大的开发者社区和生态能够在成为云时代原住民开发者的状态下，有个更好的开发环境。

InfoQ: 您从事前端工作多久了？对这个行业有过哪些困惑与思考？

杜欢（风驰）：我其实进入到前端行业还是很有趣的一个过程，我最早是在 2001 年左右开始接触到 Web 开发。那个时候，就是做网站，做网站前端、后端、数据库，然后发布运维都要做。那个时候其实也没有现在这么多岗位，基本上就一个岗位——开发，所有的事情都做。

后来随着公司业务的拓展，开始去接一些 Browser 端的工作，当时有一个词叫做 BS，它和 CS 是对应的，CS 叫 Client Side，就是客户端。Client 和 Server，就是客户端和 Server。BS 是 Browser 和 Server。从那个时候开始，这种 BS 结构的应用出现，这种结构的出现其实当时是为了解决开发成本和部署成本的问题。就是有些企业想做一个系统，这个系统可以很容易地让整个企业内部不同的团队、不同的角色很好地利用，部署的成本不要那么高，开发成本也不要那么高。

所以那个时候开始有这种业务类型出现，这种业务类型操作的主要界面就在 Browser 端，那 Browser 端就会遇到一个很大的挑战，也就是说，你的操作行为、表现、习惯跟原来传统的客户端软件开发的那种操作体验是不太一样的。因为 Browser 是浏览器。浏览器里边就是很有限的几个元素 API。然后主要客户就会提一些要求说，你需要帮我把传统的那种体验交互保留下来。因为对我而言，我只是换了一个软件提供商，但是我和我的同事在使用的时候不能有什么感知，对他们来讲应该是一样的。

那个时候遇到的挑战是，在浏览器里如何实现和传统的开发软件里的 UI、组件一样的行为。举个例子，比如说你在搜索框里输入任何一个字符，它会有下拉提示，这是非常常见的一个 UI，但在那个时候是没有的。这个 UI 至今也是没有原生提供的，都是前端去模拟出来的。

所以那个时候我做的就是这些事情。做着做着，我发现挑战非常大，相当于你要完全模拟出一套传统的开发体系里的整个 UI 体系。那个时候就想，我能不能把这个做得更好一点。所以慢慢地开始加入到前端的一些社区。认识了当时的一些朋友。就这样进入到前端行业，一直做到今天。

InfoQ: 前端的发展很快，在研发体系的升级上，阿里云是如何部署的？

杜欢（风驰）：前端升级确实是让人又爱又恨的。而且这种升级，在我看来，比如说框架层，它可能要解决的是一些新的研发形态，但是对业务而言，它其实并没有很大规模地解决上一个阶段遇到的问题。

举个研发效率的例子，比如我们现在做工程化、框架的演进，和最早用 jquery 的时候，相对业务而言，有什么变化吗？没有什么变化。而且有时候反而使你的整个协同成本、交付成本、人力成本在一定程度上变高了。因为你引入了工程的概念，你就要去做工程化，工程化不是所有人都能做得很深入。因为工程化本身就是一个领域，所以你又得为了把工程化做好去准备一些特定的候选人，组建一个团队。相对来讲，你又多了一批做业务的人，业务流程又要变慢。原来可能你在 UI 上做完，JS、HTML、CSS，你怎么做马上就可以看到。现在你是看不到的，你写完之后要编译，工程化和编译完之后，你可能才能看到。

我想表达的是，前端不断地在演进，它其实是更精细化了，质量更有保障，在一定程度上效率可能也有提升。但是从更宏观的角度，从业务的角度来看，它可能不一定真正解决了业务痛点。比如说今天我们提到，有的业务期望是，人一进来马上就能干活，干完活马上就能上线。从业务的视角来看，前端这几年的演进可能还不是一个终态，它处在一个摸索的阶段。

InfoQ: 所以就像您说的，工程化现在还没有达到它的预期效果？

杜欢（风驰）：我们认为，工程化的出现和持续演进未来一定是能帮到业务的，但是它还在摸索的阶段。本身做工程化需要消耗人力、资源，包括流程的新增，这些其实在这个阶段是会降低业务交付效率的。所以我们也不能说它不对，因为它毕竟有一个发展的过程。只是在它还没有到达终态之前，不管是框架还是这种工程化的这种演进，相对来讲都是比较痛苦的。

但是未来如果终态来临，随着未来结合云原生 Serverless，从写代码到最终发布一体化的时代来临，可能所有的问题就迎刃而解了。

InfoQ: 我之前采访过一个专家，他说，前端工程化就是在做“消灭”自己的工作，您怎么看？

杜欢（风驰）：我是这么理解，如果是消灭自己，那意味着，前端这个岗位目前做的事情未来会有一个东西替代它。

那今天前端的岗位在做什么事情呢？核心是在做用户交互行为的开发，在普遍的基础上，如果加上业务的特性，用户交互行为就会有很多定制化的东西。再加上，因为每一个业务都要差异化才能生存，尤其是 to C 的产品类型，它一定会在用户侧寻找和竞品的差异化，用户侧更多的表现就是怎么让用户看起来更舒服，操作起来更舒服，整个体验更好。这些往往会表现在真正的用户交互行为上的差异。

这里有一个矛盾的点，抽象出来的那些东西，通过工程化确实能以一定的手段来替代，但是差异化的东西怎么做，是不是能够完全替代，这个还很难说，至少今天还没有一个大家都觉得可信的方案说能够替代掉。就像今天的企业级定制开发也是类似，之所以叫定制开发，就是它至少在提定制的这个时间点，没有一个可抽象、可覆盖它的一个通用的东西，要不然它就不需要定制了，就用通用的就好了。

所以我觉得工程化能够消灭那种通用抽象的东西，但是定制的东西至少目前来看还不能，除非未来机器学习演进到能够理解真正不同的需求，并且能够把这种需求跟现有的技术体系、科学体系完整地链接起来的时候，那我觉得是会有机会的。

InfoQ: 阿里经济体的前端技术架构是什么样的？它经历了哪些发展阶段，可否提取几个重要的时间节点谈谈？

杜欢（风驰）：阿里经济体的前端在一定程度上，至少能代表国内的前端行业发展的阶段。首先，据我所知，在国内，前端这个岗位最早就是在阿里出现的。那个时候为什么会出现前端？已经从原来的所有的应用由一个人开发变成一种用户需求导向，用户觉得你这个应用虽然好，但是操作起来很差，或者整个体验不好，所以能不能有人把这块做得更好？所以在业务的需求下产生了职业精细化的要求。这个精细化的需求在前端岗位诞生的时候，它的核心是把结构、表现、行为这三者做精细化的处理或演进。这是这个岗位诞生之初阿里前端在做的事情。

后来随着业务体量逐渐增大，开始覆盖到的人群，以及人群所在的地理位置都不太一样的时候，越来越多的来自网络比较差的环境的用户会说，打开特别慢，体验不好，那个时候又经历了做性能优化的时代。性能优化主要的目的是，让不同的地理位置的用户都能够以最好的速度访问到我们的业务，让大家的体验尽量是最好的。

第三个阶段，Node.js 的出现，为我们前面谈到的工程化提供了基础。因为做工程化意味着你要去做编译、文件处理，操作一些事情，这些东西需要有一种能力让它能够跑在本地，跑在系统里面，不只是在 Web 页面上。Node.js 当时帮助前端有能力做这件事情，然后开始演进前端如何进一步地把行为、样式、结构分离，如何做模块化的设计、模块化的开发。拆开之后，这个页面你就看不到了，你想看到，怎么把拆开的东西重新聚合起来？那个时候就是通过 Node.js 做这种整体的工程化。

第一更精细化，第二更精细化之后，能够把它编译在一起，能够看到，同时去解决或优化和原来后端的协同方式。其实在这个阶段之前，前端和后端的协同方式是比较粗暴的，是那种交接式的。就是前端做完页面，然后把产物交接给后端，后端拿着前端做的页面，在那些特定的区域操作，比如说一个表格，表格里面应该有数据，前端会填一些假的数据在里面占位，后端再把真实的数据塞在里面，那是最早的阶段。有了工程化体系之后，前端和后端的衔接就可以通过 API 的方式来做。在有 API 之前，前端可以去模拟这个假数据，通过约定的 API 规范之类。这是第三个阶段，就是工程化带来的这种更精细化的设计、模块化的设计，以及这种前后端协同的演进。

再往后的演进就是无线时代，前端开始向混合开发模式演进，比如几个框架的诞生。阿里内部也诞生了一些框架，比如大家知道的 Weex，最近的 Rax 等等。这是在 all in 无线业务背景下前端的演进。阿里内部有很多中后台的业务，它有很多相对固定的结构形态，其实我们在工程化上又进一步演进了，就是诞生了这种中后台的研发模式。这种低代码的研发模式，更多地体现在页面的搭建，当我们有足够多的设计资源，已经抽象好的、比较通用的、设计好的模块，那么就可以简单地通过一些框架，把这些模块组装在一起，而不用写代码，或者写很少的代码。这是中后台的演进。

现在，结合云的到来，企业希望通过云去提高效率。这只是一个愿望，一定要经过一个技术的演进才能落地。相应地，从今年年初到“双 11”，我们整个阿里再一次演进了自己的技术体系，升级到了 Serverless 的研发体系。它不仅可以帮助前端完成面向用户交互的开发，还能够完成整个应用的开发，整个应用的开发基于云计算的实时弹性的能力能够快速做好，并且能够真实地在线上服务好“双 11”这么大的流量，真正帮助企业实现用云来快速商业化、节约成本的初衷。

InfoQ: 阿里是什么时候开始采用 Serverless 的？

杜欢（风驰）：2017 年，阿里就开始讨论这个事情，正式启动是在 2018 年。阿里内部由于开发环境、网络的客观原因，暂时不能直接使用阿里云的公共资源，所以我们要内部实现一套公共云上有的 Serverless 的能力，所以我们在 2018 年自己建设了这么一套能力。2019 年，我们开始做上层研发的架构和模型，到今年“双十一”我们正式投入使用。

InfoQ: 云 + 端是一个老生常谈的话题，阿里云的云 + 端和其他企业的云 + 端有哪些不同之处？

杜欢（风驰）：为什么今天云出现了这么久，大家提云 + 端也提了这么久，提 Serverless 也提了有一段时间，但是真正的实践那么少呢？因为在研发实践当中还是需要很挑战的一些东西去帮助它推动。

第一个是顶层的设计，因为你是研发生态，而不是简单地利用云的能力去做一个任务，这是不一样的两件事情。如果是利用云的能力去完成一个任务，这个很简单，很多人都在用。但是现在真正利用云 + 端，利用 Serverless 的能力去帮助自己提升研发能力是没有的。问题就在于大家都缺乏对整个研发架构的改变。因为你要真正利用它，研发模型要发生改变，研发的流程链路也要发生改变，这个大家没有参考。

今天阿里作为前期的实践者，愿意分享自己的设计，为大家提供参考，未来真正要在自己的研发体系里实践，大概要怎么设计，有哪些环节，哪些关键节点，哪些特征等等。第二个是真正的实践，如果阿里巴巴也只是停留在设计上，没有拿自己的业务去实践，我相信大家也缺乏信心，也可能会认为这只是我们的想象，但是今天我们真正地通过“双十一”这个很大的场景来考验。

其实我相信这能够给到整个行业一些信息，我们不仅在思考和设计整个 Serverless，整个云 + 端落到真正的研发模式上，同时我们也通过自己的业务去验证了我们的设计是可行的。最后我们也希望，不仅是分享我们的架构设计，未来我们自己内部的整个研发平台能有机会通过阿里云开放给整个行业，让外面整个开放的生态也能够使用。大家都使用一样的方式、一样的平台、一样的架构。

InfoQ: 正如您所说，今年“双 11”是 Serverless 在阿里的第一次大检验，取得了振奋人心的效果，但是这个过程当中肯定会有一些坎坷，您能分享一下这方面的经验吗？

杜欢（风驰）：最痛苦的还是 Serverless 底座的建设。我花了比较多的时间和大家讲为什么不要去自建这一层的原因是，因为落地和实践 Serverless，不是一个技术诉求，而是一个业务诉求。为什么？因为云本身是帮助企业用低成本高效快速地实现商业化，技术只是为了让这个业务诉求落地，是这样的一个关系，所以说如果没有 Serverless 底座是很痛苦的一件事。并且如果它的能力不行，基本上也是不可用的，因为落地 Serverless 意味着你的所有服务都是跑在上面的。如果它挂了，你的业务也挂了，没有人愿意这样。

InfoQ: 所以说，小企业可能不太适合做 Serverless？

杜欢（风驰）：小企业最好不要自己去建设 Serverless 底座资源能力。第一，存在技术上的挑战；第二，存在资源规模化的挑战。因为 Serverless 的核心要素是，它是按量使用的，按量使用意味着如果今天的量很小，你就用很少的资源；如果今天的量很大，就会给你调很多资源。“双十一”的时候，流量都是亿级的流量，如果你的企业内部没有按亿级做单位的这种流量的机器资源，你怎么去调度这些资源给他人使用呢？你没办法实现按量调度。所以小企业，或者不具备这种资源规模化的企业，不需要去自建 Serverless 能力，不是说不去实践 Serverless，可以用公共云，比如说用阿里云或其他的云。

我们遇到的最困难的也是这个事情，就是内部研发网络环境和生产运行网络的问题。我们也是不互通的，我们内部也很难直接在公共云环境去使用阿里云这些已有的能力。我们其实花了一年的时间，在阿里内部推动不同的团队去建设 Serverless 底座。这是第一个我认为比较挑战的点。

第二，整个研发模型对研发体系带来了挑战。其实很多时候这种东西一出来，看起来是帮助前端拓展了边界，拓展了价值能力，但是相应来讲，后端同学可能第一反应就是，那这是不是把我革命了？我就不需要干活了？其实不是这样的。比如阿里的导购业务就是取数据展示的场景。这种事情让一个后端来做，没有任何技术价值、技术沉淀、技术成长，但是现有的研发模式就是需要有后端同学进来开发。所以其实对他们来讲，Serverless 研发模式的演进有助于帮助他们往更底层演进，让他们聚焦于真正需要做技术研究的部分。比如，这些数据的能力、服务的能力，怎么做得更好、更扎实，这是我们期望看到的。但是这个研发模式乍一看，如果大家没有深入了解的话，就会认为对整个研发模式、研发流程挑战很大，那么就需要去和大家沟通、布道，讲它对每个岗位会带来的价值。

第三，回到前端来讲，这个东西虽然看起来很美好，但如果你真正下决心要进去，对每一个前端来讲，是撕裂的成长。因为我们要开始知道这个业务是什么，为什么要做这个业务，这个业务到底服务谁，关键的指标是什么，怎么做。这个时候他已经从前端变成一个业务的功能，整个业务都是他去开发、交付。

这是从技术准备、研发体系的协同，到前端岗位的挑战三个层面的难点，是我觉得印象比较深刻的，可能是未来大家在实践当中都会遇到的。

InfoQ: 我在网上了解到，有人说 Serverless 存在不适合长时间的运行应用，完全依赖第三方服务，缺乏调试，还有构建复杂等缺点。您认同这些观点吗？对于那些还没有涉足 Serverless 的人，您可以帮助他们辨清这些概念吗？

杜欢（风驰）：我觉得没有什么对错。它只是提到了一些特征，但是我也想从特征的角度给大家鼓鼓劲。比如说今天 CNCF，就是云原生在推的事情，核心就是 Serverless。Serverless 的核心特征是什么呢？第一，按量。也就是说，先不要站在技术的角度去看，站在业务角度，它是按量的，按量就意味着，对于业务而言，它是最好的资源使用方式，既不会带来浪费，也不会不足。第二，计费方式。现在很多的方式是你买的多浪费，买的少就不够，而且需要再补买，很难把它和你的应用扩容上去。Serverless 的计费方式是按量走的，用多少付多少。另外，它是平台承载的，因为平台的实时弹性，帮助用户实现按量诉求。

第二，关于技术实践上的复杂，我觉得也只是一个阶段性的现状而已。今天整个行业还没有一个开箱即用，或者说比较成熟的研发框架或研发体系、研发平台出来，大家都在一个摸索的阶段，就包括我们自己也是刚刚摸索实践出来，并且也还不算是成熟，我们也还遇到很多要去继续推动解决的问题。所以我是这么看，先从业务的角度去看，它一定是一个最佳的路径。阶段性的痛苦肯定是有，所以没有什么对错。

InfoQ: 目前国内外 Serverless 实践存在怎样的差距？

杜欢（风驰）：相对来讲，国外的整个开发生态就时间上要比国内领先一点，原因在于国外的主流云厂商对整个 IT 行业，对整个开发生态的布道做了很多工作。国外的开发生态对云原生，对 Serverless 的接受度和实践比我们要好很多，并且也早很多。对他们而言，这是一个先发优势。提供的早，就意味着实践的多，然后大家对整个 Serverless 的通用性的东西，比如通用的研发环节能够去做一些沉淀和抽象，所以诞生了一些像 Serverless.com 这些 Serverless 的开发框架。他们更多地是站在一个第三方的公共框架的角度来看，你可能既可以用这个云厂商，也可以用那个云厂商，基于我的框架可以快速地去做，基于我的框架，框架自然会有些约束，你跟着这个框架的要求去做一些动作，然后你可以去实践，真正实践这个 Serverless 在业务里面落地，这是一些现状。

那我们今天在做得也有点类似这个事情，但是我们可能不仅仅是一个开发框架，而是希望把整个开发平台都开放出来。所以大家不仅仅是说云层面，函数层面可以按照我们提的建议去做，你甚至可以直接在我们上面去做，我们希望是这样。

InfoQ: 明年 Serverless 有哪些更细粒度的技术值得关注？

杜欢（风驰）：当 Serverless 整个研发模式大概成形之后，接下来就是实践。在实践的过程当中，对渲染层、服务层、函数运行时、框架这几层可能会有一个更深入的实践，产生更细节的一些需求。我理解从明年开始，可能就是非常 detail 的垂直分层演进了，可能会有更多的这类内容产生，比如服务编排是如何演进的，函数运行时是如何演进的，性能是怎样提升的，稳定性是怎样进一步保障好的，就是又会回到一个大的运维架构演进阶段。

InfoQ: 最后一个问题，您预测未来 5 年，前端行业会有什么变化？您所在团队目前有没有针对这些技术判断做出一些布局？

杜欢（风驰）：今年阿里经济体的其他几个大的方向，比如前端智能、搭建等，这些都有可能串联起来，成为影响整个前端行业发展趋势的一些因素。但我今天讲的更多的可能是整个研发生态的变化，未来的研发模式使前端可以供整个业务，具体到每一个环节，比如前端可能通过 UI 的智能化，让自己释放出来，通过一些成熟的视觉物料、前端物料，以及服务的物料，通过 AI 的辅助，快速地把一些原本需要前端去开发的一些模式化页面模块，通过 AI 的方式自主生成。

运维这块可能随着云原生能力的不断增强，工程化能力的补充，未来有可能进入到 NoOps，就是不需要运维，只需要关注好一些数据。因为整个弹性会跟这些数据运行的实时数据关联起来，去做不同的变化。

所以整体而言，未来五年对前端而言是能力价值进一步放大的五年，云上 Serverless 开发能力将成为前端的“金手指”，企业愿意去组建一个由云端的应用开发工程师构成的研发团队，通过研发团队结合整个云的研发体系，快速地交付它的业务。同时在这个过程当中，结合智能化进一步提高生产效率，可能是这样一个趋势。

[原文链接](#)

本文为阿里云原创内容，未经允许不得转载。