The Pragmatic Programmers

Broadview
www.broadview.com.cn

Serverless Single Page Apps
Fast, Scalable, and Available

Serverless架构
无服务器单页应用开发

[美] Ben Rady 著
郑美赞 简传挺 译

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Serverless Single Page Apps

Fast, Scalable, and Available

Serverless架构

无服务器单页应用开发

[美] Ben Rady 著
郑美赞 简传挺 译

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书讲授如何利用 Amazon 公司的 AWS Lambda 创建 Serverless 单页应用。这里，Serverless 的意思是应用开发者无须管理服务器，将应用构建在服务之上，而不是运行在需要人工配置和维护的服务器之上。这种新的开发方式带来很多好处，比如节省成本，可扩展性与可靠性高，以及开发者可以专注于实现应用的业务逻辑等。全书共 8 章，Ben Rady 带领读者采用这种新方法从零开始开发一个 JavaScript 解题应用，并且对其进行测试，最终完成部署。

对于创业者以及中小企业的开发者来说，本书讲述的 Serverless 设计是一个值得了解和学习的新方法，可以从中获得启示，抓住先机。

版权贸易合同登记号 图字：01-2016-7931

图书在版编目（CIP）数据

Serverless 架构：无服务器单页应用开发 /（美）本·雷迪（Ben Rady）著；郑美赞，简传挺译. —北京：电子工业出版社，2017.7
书名原文：Serverless Single Page Apps:Fast,Scalable,and Available
ISBN 978-7-121-31736-1

Ⅰ. ①S… Ⅱ. ①本… ②郑… ③简… Ⅲ. ①移动终端－应用程序－程序设计 Ⅳ. ①TN929.53

中国版本图书馆 CIP 数据核字(2017)第 124192 号

责任编辑：许 艳
印 刷：北京中新伟业印刷有限公司
装 订：北京中新伟业印刷有限公司
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编：100036
开 本：787×980 1/16 印张：14 字数：227.4 千字
版 次：2017 年 7 月第 1 版
印 次：2017 年 7 月第 1 次印刷
定 价：65.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819，faq@phei.com.cn。

本书赞誉

软件行业里聚集着最多的精英——上百万的开发者，他们带动技术朝着代码更容易测试、解决方案更简单、结果更可靠，以及维护起来更轻松的方向发展。有人看到了 Serverless 设计的未来，然后回过头来教授我们这些后知后觉者如何开发下一代应用，Ben 就是这样的开拓者。他的书就像是一位循循善诱的老师，教你理解 Serverless 设计模式，引导你自然地遵守部署和测试的最佳实践。

Tim Wagner
@timallenwagner

本书对于所有背景的开发者来说都是一份翔实而通俗易懂的指南。不管你是否使用 AWS，都能学到不少知识——从应用的安全到访问数据时不可或缺的身份认证。

Will Gaul

Ben 在本书中讲了很多内容：用 JavaScript 构建客户端逻辑、用 Cognito 进行认证和授权、用 Lambda 实现不能放心地交给浏览器处理的敏感功能。JavaScript 开发者会从中发现一些实现典型服务端功能的新方法，而且读完本书，你就会得到一个成本近乎为零的能运行的 Serverless 应用。

Ryan Scott Brown
serverlesscode.com 的作者
Serverless Framework 贡献者

未来你的应用不再运行在应用服务器上——而是运行在你公司某个机柜的机器里，运行在云上，由一组可靠的服务保护和管理。跟着本书开启全新的开发之旅吧！

Daniel Hinojosa
Testing in Scala 作者

本书对 Serverless Web 应用开发这种前沿技术做了精彩的介绍。它将带着你从零开始，直到部署 Serverless 应用。

Jake McCrary
Outpace Systems 公司软件开发主管

我读过很多技术图书，这一本是我今年读过的最好的书，也是我这些年读过的最好的书之一。Ben Rady 的讲述既轻松又实在，没有吹嘘自己的知识，也没有用不必要的内容凑篇幅。书中不仅告诉你要做什么，而且解释了为什么这么做，两者并重，十分清楚明了。Ben 的观点和技术选型有理有据，非常靠谱。建议你阅读本书。

David Rupp
RuppWorks LLC

谨以此书献给我的家人：

总是赐予我力量的妻子 Jenny

善良的女儿 Katherine

梦想着拯救世界的儿子 Will

致　谢

感谢我的编辑 Jackie Carter，以及 Dave、Andy 和 Pragmatic Bookshelf 的所有员工，感谢他们付出的时间，给予的支持和能量。没有你们，我不可能写这本书。

感谢我的技术审稿人，包括 Alex Disney、Cliton Begin、Daniel Hinojosa、David Rupp、Jake MacCrary、James Ross、Jeff Sacks、Joshua Graham、Lucas Ward、Rene Duquesnoy、Rob Churchwell、Ryan Brown 和 Sebastian Krueger。

感谢 Amazon 所有帮助我验证本书中想法的人，包括 Tim Wagner、Bob Kinney、Tim Hunt 和 Will Gaul。

另外，感谢所有在本书 beta 版本期间提供反馈的人，不管是私下还是通过勘误页面，是在论坛上还是在 Github.com 上提出意见的人，包括 Bill Caputo、Christopher Brackert、Christopher Mowller、Dennis Bruke、Ezequiel Rangel、Fred Daoud、Hal Wine、Jerry Tang、Ron Campbell 和 Timm Knape。

感谢所有帮助我写这本书的人！你们的反馈对我来说非常宝贵，真心感谢你们付出的时间和精力。

前　言

我做了几年单页 Web 应用，一直希望能摆脱应用服务器施加的限制，现在愿望终于实现了。Amazon（还会有更多的其他公司）开发出的技术让无服务器架构（serverless，在本书中简称为“无服”）成为可能，消除了很多构建和扩展 Web 应用的风险和成本。这个创意是如此令人叹服，如此具有变革意义，以至于我必须为它写一本书。

这本书是为那些希望在 Web 上构建一些东西的人而写的。有的人想搭建一个应用——一个他们认为将会是下一个惊世之作的东西。有的人只是刚着手 Web 开发，从未构建过任何类型的应用——不管是影响世界的还是其他什么样的。有的人则是已经用 Java、Ruby on Rails 或者 Node.js 构建过许多基于 Model-View-Controller 应用的资深 Web 开发者。随着这个无服技术的兴起，我希望能分享所学的一切，真心希望有人能够用它做出一些了不起的东西。

在前言中，你将了解本书的主题以及能用这些技术来实现些什么。

指导原则

写这本书时我脑子里有几条指导原则。这些原则的目的是让本书内容具有针对性、易

理解，不说废话。这里把它们列出来，帮助你了解我是如何写这本书的以及更好地理解本书的相关背景。

有些原则可能是有争议的，其中一些甚至与构建 Web 应用的主流思想相左。但是，这些原则将帮助你深入理解这个主题。与其波澜不惊，存在争议会更好。

使用 Web 标准和熟悉的工具

在这本书中，你将使用一小部分精选的工具来构建一个单页 Web 应用。在本书的某些节点上，你将会实现其他工具已有的功能，这些工具是我有意不在应用中引入的。你可能会好奇，既然它们提供了所需的功能，为什么我们不用呢？

阅读本书实际上是一个学习的过程。当学习知识时，使用熟悉的工具是有帮助的。不然，花在学习工具上的时间比花在学习技术上的还要多。我不希望对一个框架或者库的选择让我们偏离本书主题。这本书是关于无服 Web 应用，而不是关于框架或者类库的。为了保持未知性，我们将会使用那些 Web 开发者熟悉的工具（例如 jQuery）、Web 标准和 Web 服务，来实现一个无服设计。

这本书是关于 Web 服务，而不是 Web 框架的。

有可能读完这本书，你将会接触到一种客户端 Web 框架，比如 React 或者 Angular，来构建你自己的 Web 应用。这些工具近些年在 Web 开发者社区获得了很多关注，我希望能看到很多使用它们的成功项目。在本书中学到的所有知识与你希望使用这些框架做的事情都是兼容的。它们是互为补充而不是相互冲突的关系。

使用函数式 JavaScript

本书中你将不会创建 JavaScript 的任何类。创建类结构来解决问题对于一些具有富类型和面向对象类型系统的语言是合理的，但 JavaScript 不是这样的语言。相反，我们将使用更易于理解的函数式风格。

这意味着你将不会遭遇 this 关键字的范围问题。你将会避免原型和继承的共同使用。你不会使用 new 关键字和专门设计的函数来创建对象；相反，你将会使用一个字面意义的对象来创建它们：{}。

你可以自己决定这是否就是你希望采纳的风格。因为这种方式拥有一些实际、立竿见影的好处，最后许多软件设计决策实际上都会偏向这种风格。代码终究应该首先是给人写的，然后才是计算机。只要它可执行，对于计算机而言代码看起来如何无所谓。

避免无用功

做项目时，我的目标一直都是：持续交付有改进的结果，只要这个改进是朝着环境需要的方向在进行。实现这个目标意味着要避免任何能造成交付率被消磨的事情，比如剃牦牛（yak shaving）。

如果你对“剃牦牛”这个词不熟悉的话，想象一下你想要给朋友买一件毛衣。你走进一家商店却发现没有毛衣卖。幸运的是，那里的另一位客户知道街角有一个很好的裁缝，他可能可以为你做一件。所以你到了裁缝那儿，他有一个非常好看的毛衣图案和一台能编织这个图案的机器，但毛线供应商今天还没送货。所以你跑到供应商那儿……

然后这个过程一直继续，直到你发现自己在西藏的一片牧场上剃耗牛毛来编毛线。“我怎么来这里了？”你可能会问自己，“我所需要的只是一件毛衣。”当一系列看起来相关和必要的任务让你从真实目标上偏离时，你就是在剃耗牛。幸运的是，剃耗牛的解决办法很简单，就是意识到你自己一直在剃耗牛，然后转而买一顶帽子。

我希望你在阅读本书时不要剃耗牛。这就是为什么我用了一个预备好的工作空间以及尽可能少的工具的原因。你应该把时间花在学习上，而不是安装软件、配置和排错上。

通过测试来加快进度

你有没有曾经害怕过修改甚至只是修改一点点代码？也许你当时不确定应该做什么或者为什么应该做这个。“首先，不作恶”对程序员和医生而言均适用。这种情况会让你感

觉进退两难。

设想你做这些改动时，有一位可信的同事——这类系统的专家——就坐在你身边。如果你引入任何缺陷到系统中，他就会拦住你，清楚而明确地告诉你为什么这个改动是个坏主意。如果你有这样可信的同事，是否还会进退两难？

不确定性拖慢了我们的脚本，并且限制了解决方案的范围。为了快速搭建一个软件，你必须有自信。为了获得这份自信，你可以为自己创造一个自动化专家——它了解系统的所有细节，知道系统如何运转以及为什么这样运转。这位专家和系统相辅相成，随着它的改变而改变，互相影响。这个专家就是你编写一套测试，它们快到能持续运行，每秒能运行成百上千的测试，而且每次在你对代码进行修改之后都是如此。

一旦你有了测试套件，就拥有了新选择。而当你不再害怕修改代码时，就可以按照自己的设想来设计应用，而不是试图让它在一开始就要“正确”。这意味着你可以随着形势的变化快速适应这个世界，而不是试图预测你可能需要和不需要什么。你可以基于当下的信息而不是靠猜来做决策，应用就会自然而然地变成它应该的那个样子。

在本书中，你将使用测试驱动开发（TDD）技术来编写应用和它的测试。了解如何用测试来构建软件是一项技能，要获得它的益处，就必须真正地使用它。我在本书中引入 TDD 的目的不仅是向你展示如何测试 Web 应用的特定功能，还想证明用它来测试典型 Web 应用有多简单，如果你知道如何实现的话。

在实践 TDD 的过程中，有一个三步循环，经常描述为红—绿—重构。首先写一个测试，检查应用尚不具备的功能。如果测试如你预期的那样失败了，就可以相信它是在测试你希望加到应用上的功能。现在这些测试是红色的。一旦有一个失败的测试，添加几条最简单的代码来让这个测试通过，通常几行代码即可。现在这些测试是绿色的。

通过测试为应用添加一些功能后，就该后退一步，让自己看得更全面一些。是否在实现中引入了一些重复代码？所有的变量和函数是否都有描述性的准确名字？是否还有更简单的方式？考虑这些事情，然后开始重构。重构是在不改变功能的情况下修改代码。你之

前编写的测试会告诉你是否改变了功能，所以只能在这些测试都通过时重构，这一点很重要。有了它们作为你的安全防护网，可以在开始下一步测试之前清理完所有代码的问题。

TDD 用得越多，你的进展就会越快，从中获得的价值就越多。通过一遍又一遍地重复“红—绿—重构”过程，你将学会如何渐进式地构建一个设计合理、测试充分的应用。这样你不仅对自己的应用有信心，而且对它进行持续的修改也更容易。

边做边学

本书是一本教程，所以你可以跟着书中的引导边做边学。通过这本书，你将构建一个无服单页 Web 应用。本书的目的是用具体的方式解释无服架构的原理。本书的成果是一个可运行的应用[1]，所以你大可以放心，书中的技术就像广告中说的那样有用。

采用这种方式意味着篇幅有限，我对有些知识点不可能讲得很深入。因此，我在每章中安排了一节“下一步”来提供一些主题，如果你希望了解更多，可以查找相关的资料。

从预备好的工作空间开始

为了让你能快速上手，我提供了一个预备好的工作空间（prepared workspace），里面包括了起步需要的所有东西，并且不需要太多时间来设置。这就好比你在画一幅艺术品，我已经为你准备好颜料、画架和画布。你要做的就是创作。

为了能使用这个预备好的工作空间，你需要一台兼容 Bash shell 的计算机来使用工作空间里面的脚本和工具。可以是 OS X、任何*nix 版本或者 FreeBSD。如果你安装了 Cygwin，用 Windows 可能也行。你还需要一个带开发者控制台的 Web 浏览器。本书的大部分例子中使用的是 Google Chrome，但大多数情况下 Firefox 也能提供了类似的功能。

1 http://learnjs.benrady.com

如何阅读本书

阅读一本书可以有很多种方式，不仅是从头到尾的那种。至于如何阅读本书，取决于你想要获得什么。下面列出的是读本书的一些常见理由，以及我对于如何使用这本书来实现这些目标的建议。

目标 1：理解无服和传统单页应用的优劣

需要做什么

1. 阅读第 1 章的前 3 节理解两种方式的优劣。
2. 略读第 1 章剩下的部分和第 2～3 章。
3. 通读第 4～8 章，尽可能跟着教程实现应用。

如果你是一个有经验的 Web 开发者，搭建过单页 Web 应用，希望了解更多有关无服 Web 应用的知识，你可能不需要看前 3 章的所有内容。这些章节展示了搭建单页 Web 应用的 vanilla.js 方式。其意图是阐述一个单页 Web 应用的基础模块。我尽量定义哪些部分是必需的，并提供一些可供参考的基础实现。如果你没有给 Web 应用编写过很多测试，可以在第 2 章中实现一些测试例子来学习一些新技能。

读完第 1 章的前 3 节，你会希望集中精力细读第 4～8 章。我建议你搭建一个简单 Web 应用，或者至少一个原型，这样你可以在后面的章节中自己尝试这些技术。通过实验可以学到很多。

目标 2：搭建你的第一个单页应用

需要做什么

1. 在开始前，先读一读书中列出的必要的辅助阅读材料。
2. 从头到尾读完本书所有的章节，如果有必要，附录也要看。

如果你是第一次搭建 Web 应用，你会希望阅读整本书，从头到尾读一遍。另外，你可能也想跟上基础 Web 技术的脚步，包含 HTML、CSS 和 JavaScript，可以查阅下面的免费资源。等理解了这些主题，再来深入读本书。

免费资源

Learn to Code HTML & CSS [How14]，Shay Howe 著[2]

Eloquent JavaScript [Hav14]，Marijin Haverbeke 著[3]

目标 3：使用你喜欢的 Web 框架搭建一个无服应用

需要做什么

1. 通读第 1 章。
2. 用你喜欢的 Web 框架实现第 2 章和第 3 章的测试与功能。
3. 通读第 4～8 章。

正如前面说的那样，我不希望把这本书的重点放在 Web 框架上，但还是可以使用它们来搭建无服单页应用。如果你对单页应用的基本元素很熟悉，并且靠客户端 Web 框架来提供这些功能，可以用喜欢的框架轻松重建我们在第 2 章和第 3 章实现的功能。一旦有了这个作为基础，应该可以遵照本书余下的部分，用这个框架实现整个应用。

目标 4：创建一个最小可行产品（Minimum Viable Product，MVP）

需要做什么

1. 阅读第 1 章的前 3 节，理解无服开发与传统开发方式的优劣。
2. 阅读第 8 章来了解用这种方式构建一个 MVP 的成本。
3. 如果这个方式看起来行得通，阅读剩余的所有章节（如果有需要的话，

2 http://learn.shayhowe.com/html-css/

3 http://eloquentjavascript.net/

包括附录）。

3. 使用预备好的工作空间作为搭建 MVP 的起点。

当启动一个新产品或者一项新业务时，首要的挑战是搞清楚市场想要什么以及它愿意为什么付钱。许多人称之为产品/市场匹配（product/market fit），这是构建一个成功产品或者服务的关键。

找到产品/市场匹配的一个有效方式是先做出一个产品并试着销售。验证市场的需求和你通过销售或市场渠道连接这些客户的能力，这是你应该尽快跨过的关键门坎。当然，你可能又没那么多时间和资金来搭建一个完整的应用，所以一个替代方案是搭建一个 MVP 来验证产品的核心价值。

无服单页应用是尝试新主意、探索可能的新市场或者创建 MVP 的好方式。搭建这类应用来替代传统 Web 应用或者原生应用，意味着你可以更快接触到客户。你可以在若干小时内搭建一个初始版本并在几秒内部署。这些应用可以立即更新，轻松地被拆分测试，可以提供详细的使用指标，帮助你理解客户想要的是什么。

除了运行起来不贵、快速构建以及几乎所有用户都能访问之外，无服单页应用可以“无限”（正如 Amazon 宣传的那样）扩展，这样如果你的产品在市场上有旺盛的需求，你就可以满足这样的需求以及留住用户。

在线资源

你可以在 Progmatic Bookshelf 官网找到这本书中的应用以及代码[4]。你还能在上面看到社区论坛和提交勘误的表单，报告问题或者为未来的版本提供建议。

4 https://pragprog.com/book/brapps/serverless-page-apps

你可以在我的 Github.com 账号（benrady）[5]下找到预备好的工作空间。如果你还没有 Github 账号，那就新建一个，并且 fork 我的代码库。想知道更多操作细节，可以阅读第 1 章的内容。

Ben Rady
benrady@gmail.com

轻松注册成为博文视点社区用户（www.broadview.com.cn），扫码直达本书页面。

- **提交勘误**：您对书中内容的修改意见可在 *提交勘误* 处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **交流互动**：在页面下方 *读者评论* 处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：http://www.broadview.com.cn/31736

5 https://github.com/benrady/learnjs

目　录