



Ben Noordhuis作序

Node.js

硬实战

115个核心技巧

[美] Alex Young Marc Harter 著
承竹 慕陶 邱娟 达峰 译

Node.js in Practice

Node.js 硬实战 115个核心技巧

[美] Alex Young Marc Harter 著
承竹 慕陶 邱娟 达峰 译

Node.js in Practice

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

内 容 简 介

《Node.js 硬实战：115 个核心技巧》是一本面向实战的 Node.js 开发进阶指南。作为资深专家，本书作者独辟蹊径，将着眼点放在 Node.js 的核心模块和网络应用，通过精心组织的丰富实例，向读者充分展示了 Node.js 强大的并发处理能力，读者从中可真正掌握 Node 的核心基础与高级技巧。本书总共有三部分内容，第一部分是 Node.js 的基础核心，涉及 Buffer、流、网络 and 进程等相关知识；第二部分是项目实践，涉及测试、Web 开发、调试，生产环境等重要话题；第三部分则完整创建了一个 Node.js 模块。每部分涉及的技术都有详细讲解及注释详尽的示例代码，以帮助读者们更好地理解要点及其应用。

本书适合有一定 JavaScript 基础，追求在 Node.js 上更进一步的开发者。

Original English Language edition published by Manning Publications, USA. Copyright ©2015 by Manning Publications. Simplified Chinese-language edition copyright ©2016 by Publishing House of Electronics Industry. All rights reserved.

本书简体中文版专有出版权由 Manning Publications 授予电子工业出版社。未经许可，不得以任何方式复制或抄袭本书的任何部分。专有出版权受法律保护。

版权贸易合同登记号图字：01-2015-3992

图书在版编目（CIP）数据

Node.js 硬实战：115 个核心技巧 / （美）亚历克斯·荣（Alex R. Young），（美）马克·哈特（Marc Harter）著；承竹等译.—北京：电子工业出版社，2017.1

书名原文：Node.js in Practice

ISBN 978-7-121-30402-6

I. ① N…II. ① 亚…② 马…③ 承…III. ① JAVA 语言—程序设计 IV. ① TP312.8

中国版本图书馆 CIP 数据核字（2016）第 277756 号

责任编辑：张春雨

印 刷：三河市良远印务有限公司

装 订：三河市良远印务有限公司

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：30

字数：672 千字

版 次：2017 年 1 月第 1 版

印 次：2017 年 1 月第 1 次印刷

定 价：109.90 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 zltz@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。

译者序

JavaScript 从它诞生以来就在浏览器应用中发挥着愈来愈重要的作用,同时,热爱 JavaScript 的人们一直在努力地将其应用在服务器领域。2009 年, Ryan Dahl 带来了 Node.js, 从那个时候开始, JavaScript 社区出现了前所未有的繁荣。现在, Node.js 对开发者来说, 几乎已经是家喻户晓的了。对于传统的服务器端开发者们而言, Node.js 带来了浏览器端使用已久的异步编程相关的概念, 而对于前端开发者们来说, Node.js 则是带来了编写服务端程序方面的挑战。当我们这些对 Node.js 一知半解的开发者野心勃勃想要征服 Node.js 时, 我们需要一个经验满满的导师来指引我们前行。

我很幸运地有这样一个机会接触到这本书, 并且承担起翻译的任务。我第一次阅读原著时就感觉, 它是一本可以胜任 Node.js 导师这个角色的书籍。作者认真负责地介绍了 Node.js 相关的方方面面, 并且附带了相当详细的例子来帮助读者快速地理解其中的要点。我相信, 对于想要学习 Node.js 的开发者来说, 本书会是一个相当好的选择。

翻译书的过程, 也是一个学习的过程, 原谅我们水平有限, 书中翻译内容难免有疏漏。当这本书差不多翻译完成的时候, 我发现已经用了好长的时间, Node.js 技术更新得相当快, 我相信小部分内容的细节上和现在的应用可能会出现些许差异, 还请读者们谅解。

最后, 在此对支持本书翻译工作的所有人们表示感谢, 尤其是耐心容忍我们一再推迟交稿的编辑, 真的非常感谢。

衷心希望本书能够对您有所帮助, 请享受 Node.js 给您带来的一切, 谢谢。

序

在你手中的是一本将带你进入了解 Node 旅程的书。在接下来看到的页面中，Alex Young 和 Mare Harter 会帮助你深刻地把握 Node 的核心模块、网络应用等。

网络应用，是 Node.js 光芒闪耀的地方。亲爱的读者，你可能已经很清楚这一点，我敢说，这就是你购买本书的原因！对于读了序的这少部分人，让我来告诉你们，这一切是如何开始的。

在一开始的时候，有一个 C10K 的问题，这个问题是这样的：如果你要在现代的硬件中处理 10000 个并发连接，你会怎么做？

你可以看到，操作系统处理大量的网络连接在持续很长一段时间里都是很糟糕的。硬件在很多方面很糟糕，软件在另外一些方面也很糟糕，当硬件和软件集成在一起的时候……语言学家暂时还没有合适的词语，单纯用糟糕来形容对这一切并不公平。幸运的是，技术是一个进步的故事，硬件越来越好，软件越来越智能。操作系统，如用户软件等，在管理大量的网络连接上有了很大的进步。

在很久以前，我们征服了 C10K 问题，现在目标转移了，我们已经把眼光投向了 C100K、C500K、C1M 问题。一旦我们轻松地跨越这些界限时，我完全相信，C10M 会是下一个问题。

Node.js 是这个并发性不断增长的故事的一部分，它的未来是光明的。我们生活在一个日益互联的世界，这个世界需要一个强大的工具来连接一切。我相信 Node.js 会是那个强大的工具，我希望，在读完这本书时，你会有同样的感受。

Ben Noordhuis
Cofounder, StrongLoop, Inc.

前言

当 Node 在 2009 年出现的时候，我们知道有一些东西不一样了。在服务端 JavaScript 并不是什么新鲜的东西。事实上，服务端的 JavaScript 几乎和客户端的 JavaScript 存在的时间一样长。Node 中，JavaScript 运行时的速度，再加上基于事件的并行，这些很多熟悉 JavaScript 程序员都熟悉的东西，确实是让人感到不可抗拒。不仅仅是像我们这样背景的客户端 JavaScript 开发者——Node 吸引了从系统层面到各种服务端开发、PHP、Ruby 或者 Java 的开发者们。我们都可以发现自己身处于这变化中。

在那个时候，Node 有很多变化，我们受困于它，但是在这个过程中也学习了很多东西。从一开始，Node 关注于一个小的、低级别的核心库，来为大量多样化并且增长的用户提供足够的功能。值得庆幸的是，因为早期的一些设计决定，今天大量多样化的用户空间还存在着。Node 现在更加稳定，并且在许多初创企业和已成功企业生产环境中使用。

当 Manning 联系我们来编写一本中级的、关于 Node 的书时，我们想到了在过去和常见陷阱做斗争时，以及在 Node 社区中获得的经验教训。尽管我们非常喜欢那些提供给开发人员的大量真正优秀的第三方模块，但是，我们留意到越来越少的开发人员接触到 Node 核心基础的教学。所以我们着手编写了《Node.js 硬实战》，来以一种深入和彻底的方式，探索 Node 的根源和基础，并且解决很多我们个人遇见过的，或者看到他人处理过的问题。

致谢

我们要感谢很多人，没有他们的帮助和支持，这本书不可能完成。

感谢 Manning Early Access Program (MEAP) 的在作者在线论坛发布评论和纠正的读者们。

感谢那些在书的各个阶段提供了宝贵反馈意见的技术评审们：Alex Garrett、Brian Falk、Chris Joakim、Christoph Walcher、Daniel Bretoi、Dominic Pettifer、Dylan Scott、Fernando Monteiro Kobayashi、Gavin Whyte、Gregor Zurowski、Haytham Samad、JT Marshall、Kevin Baister、Luis Gutierrez、Michael Piscatello、Philippe Charrière、Rock Lee、Shiju Varghese 和 Todd Williams。

感谢帮助我们每一步的 Manning 团队，尤其是我们的开发编辑 Cynthia Kane、编辑 Benjamin Berg、校对 Katie Tennant，以及其他在幕后工作的人们。

特别感谢 Ben Noordhuis 为本书写序，还有 Valentin Crettaz 和 Michael Levin 在书出版前仔细的技术校对。

Alex Young

如果没有 DailyJS 社区的鼓励，我没法完成这样的一本书。感谢近几年来和我分享模块和类库的每一个人：没有你们就无法跟上 Node.js 社区的脚步。同时也要感谢我的同事允许让我在生产环境中使用 Node.js。最后，感谢 Yuka 让我相信我自己可以从事创业和写书这些如此疯狂的事情。

Marc Harter

我要感谢 Ben Noordhuis、Isaac Schlueter，还有 Timothy Fontaine，为了他们关于 Node 的各种 IRC 讨论：你们了解底层系统，以一种深刻的方式支持和影响着 Node，从你们身上我们获益匪浅。另外，我要感谢和我一起写书的 Alex，似乎很难能有像我这样和 Alex 一起写书的机会，更加有趣的是他是一个在美国中西部的使用英语的人。最后我要感谢的是我的妻子，她让这一切变得可能，我由衷地说一句，你是如此可爱，谢谢你。

关于本书

《Node.js 硬实战》提供给读者来深入了解 Node 的核心模块和包系统。我们相信这是成为一个多产和自信的 Node 开发者的基础。不幸的是，因为巨大并且充满活力的第三方生态系统几乎为所有的任务提供了预置好的模块，所以小小的核心很容易被错过。在这本书中，我们在官方文档的基础上来进一步实践和深入。我们想要读者能够仔细分析和研究他们编写的项目以及项目所包含的第三方模块的内部工作。

这本书不是一本 Node 的入门级别的读物。入门的话，我们建议阅读 Manning 的《Node 实战》(*Node.js In Action*)。这本书的受众是那些对 Node 有一定经验，正在寻求更进一步的读者。建议有一定的 JavaScript 基础的读者，最好也熟悉 Windows、OS X 或者 Linux 命令行的读者阅读。

同时，我们注意到很多 Node 开发者有来自客户端的 JavaScript 开发背景。因此，我们会花一些时间来解释一些不大熟悉的概念，如处理二进制数据、底层网络和文件系统的工作，以及和主机操作系统进行交互——所有这些都使用 Node 来作为教学指南。

章节路线图

这本书共分为三部分。

第一部分包括了 Node 的核心基础，我们关注于那些可能用到的 Node 核心模块（非第三方模块）。第 1 章简要概述了 Node 的目的和意义。第 2 到第 8 章每一章内容会深入 Node 核心的不同方面，从 Buffers 到流，从网络到子进程。

第二部分内容关注于真实世界的开发技巧。第 9 到第 12 章的内容，将帮助你掌握 4 种

非常实用的技能——测试、Web 开发、调试以及在生产环境运行 Node。除了 Node 核心模块之外，这些章节的内容也会包括许多第三方模块的使用。

第三部分将指引你以一种直接的方式来创建自己的 Node 模块，使用 npm 命令的各种方法来处理打包、运行、测试、基准测试和共享模块。它同时也包括进行有效的项目版本化的有用提示。

整本书有 115 个技巧，每一部分包括了一个特定的 Node.js 主题或者任务，包括了实际问题、解决方案和讨论部分。

代码约定和下载

在本书中的所有代码都是像这样的固定宽度的字体，如 `fixed-width font like this`，以和周边其他文本内容区分开来。在很多列表中，代码中的关键概念会有相应的注解，带有编号的符号有时候可以帮助在文本中找到其他额外关于代码的有用信息。

这本书的代码风格基于 Google JavaScript 的编码风格¹。这意味着我们把 var 声明独立开来，使用驼峰式来命名函数和变量，并且通常都带分号。我们的风格是在 Node 社区使用的多种 JavaScript 代码风格的混合。

大部分书中所示的代码可以以不同的形式在伴随着书的示例源代码中找到。示例源代码可以免费地从博文视点网站上下载：www.broadview.com.cn，也可以在 Github 上关注：<https://github.com/alexyoung/nodeinpractice>。

读者在线服务

购买《Node.js 硬实战》并注册账号，你可以在上边对本书进行评论、提交勘误或者咨询问题，从编辑或者其他用户那里获得帮助。

¹<https://google-styleguide.googlecode.com/svn/trunk/javascriptguide.xml>

关于封面插图

《Node.js 硬实战》的封面插图的标题是“来自 Ayvalik 的年轻人”，Ayvalik 是在土耳其爱琴海岸的一个小镇。这个插图是从奥斯曼帝国在 1802 年 1 月 1 日发布的伦敦 Old Bond 街的 William Miller 创作的一系列服饰中获取的。标题页在整个集合中丢失了，我们已经无法追溯其日期。在书的内容表格中带有英文和法文标识，每个插图都有两位艺术家的名字。两百年之后，你一定会惊讶地发现，他们的艺术作品竟让计算机编程的书变得如此优雅、有魅力。

这一图片系列是由 Manning 的一位编辑在 Manhattan 的西第二十六街 Garage 的一个跳蚤市场购买的。卖家是一个在土耳其 Ankara 的美国人，交易就发生在他正要收摊的时候。Manning 的编辑身上并没有足够的现金，信用卡和支票都被礼貌地拒绝了。当天晚上卖家就要坐飞机回 Ankara，好像没有希望了。结果是怎么解决的？仅仅只是握手，一个老式的 verbal 协议。卖家简单地提出将钱电汇给他就好，编辑带着一张写了银行信息的纸和胳膊下的公事包就走了。当然，第二天我们就把钱汇过去了。我们依旧很感谢这个陌生人对我们的信任。这是很久前的一段回忆。

在 Manning 的我们，为自己的创意和突破，以及基于带回来的两个世纪前的照片集的计算机书籍封面能如此丰富多彩而庆幸。

目录

第一部分 Node 基础

1	入门.....	2
1.1	Node 入门	3
1.1.1	为什么使用 Node	3
1.1.2	Node 的主要特性	5
1.2	构建一个 Node 应用	7
1.2.1	创建一个新的 Node 项目	8
1.2.2	创建一个流的类	9
1.2.3	使用流	10
1.2.4	编写测试	11
1.3	总结	13
2	全局变量: Node 环境.....	15
2.1	模块	16
技巧 1	安装与加载模块	16
技巧 2	创建与管理模块	17
技巧 3	加载一组相关的模块	20
技巧 4	使用路径	22
2.2	标准 I/O 以及 console 对象	23
技巧 5	标准 I/O 流的读写	24

技巧 6	打印日志消息	25
技巧 7	基准测试	27
2.3	操作系统与命令行	29
技巧 8	获取平台信息	29
技巧 9	传递命令行参数	30
技巧 10	退出程序	31
技巧 11	响应信号量	33
2.4	使用 timer 延迟执行	35
技巧 12	通过 setTimeout 延迟执行函数	35
技巧 13	通过定时器定时调用回调函数	37
技巧 14	安全的操作异步接口	38
2.5	总结	41
3	Buffers: 使用比特、字节以及编码	43
3.1	修改数据编码	44
技巧 15	Buffer 转换为其他格式	44
技巧 16	使用 Buffers 来修改字符串编码	46
3.2	二进制文件转换为 JSON	49
技巧 17	使用 Buffer 来转换原始数据	49
3.3	创建你自己的二进制协议	65
技巧 18	创建自己的网络协议	65
3.4	总结	71
4	Events: 玩转 EventEmitter	72
4.1	基础用法	73
技巧 19	从 EventEmitter 继承	73
技巧 20	混合 EventEmitter	76
4.2	异常处理	78
技巧 21	管理异常	78
技巧 22	通过 domains 管理异常	80

4.3	高级模式	82
	技巧 23 反射	82
	技巧 24 探索 EventEmitter	85
	技巧 25 组织事件名称	87
4.4	第三方模块以及扩展	88
	技巧 26 EventEmitter 的替代方案	89
4.5	总结	91
5	流：最强大和最容易误解的功能	93
5.1	流的介绍	94
	5.1.1 流的类型	94
	5.1.2 什么时候使用流	94
	5.1.3 历史	95
	5.1.4 第三方模块中的流	96
	5.1.5 流继承事件	97
5.2	内置流	98
	技巧 27 使用内置的流来实现静态 web 服务器	98
	技巧 28 流的错误处理	101
5.3	第三方模块和流	102
	技巧 29 使用流的第三方模块	102
5.4	使用流基类	105
	技巧 30 正确地从流的基类继承	105
	技巧 31 实现一个可读流	107
	技巧 32 实现一个可写流	111
	技巧 33 使用双工流转换和接收数据	113
	技巧 34 使用转换流解析数据	114
5.5	高级模式和优化	118
	技巧 35 流的优化	118
	技巧 36 使用老的流 API	121
	技巧 37 基于功能的流适配	123
	技巧 38 测试流	125

5.6 总结	128
6 文件系统：通过异步和同步的方法处理文件.....	129
6.1 fs 模块概述	130
6.1.1 POSIX 文件系统包装器	130
6.1.2 流	132
6.1.3 批量文件操作	133
6.1.4 文件监视	133
6.1.5 同步的替代方案	133
技巧 39 读取配置文件	134
技巧 40 使用文件描述	136
技巧 41 使用文件锁	137
技巧 42 递归文件操作	142
技巧 43 编写文件数据库	147
技巧 44 监视文件以及文件夹	151
6.2 总结	154
7 网络：Node 真正的“Hello, World”	156
7.1 Node 中的网络	156
7.1.1 网络技术	157
7.1.2 Node 网络模块	161
7.1.3 非阻塞网络和线程池	162
7.2 TCP 客户端和服务端	163
技巧 45 创建 TCP 服务端和客户端	163
技巧 46 使用客户端测试 TCP 服务端	165
技巧 47 改进实时性低的应用	168
7.3 UDP 客户端和服务端	170
技巧 48 通过 UDP 传输文件	170
技巧 49 UDP 客户端服务应用	174
7.4 HTTP 客户端和服务端	179
技巧 50 HTTP 服务器	179

技巧 51 重定向	181
技巧 52 HTTP 代理	186
7.5 创建 DNS 请求	189
技巧 53 创建 DNS 请求	189
7.6 加密	191
技巧 54 一个加密的 TCP 服务器	192
技巧 55 加密的 Web 服务器和客户端	196
7.7 总结	198
8 子进程：利用 Node 整合外部应用程序	200
8.1 执行外部应用程序	202
技巧 56 执行外部应用程序	202
8.1.1 路径和 Path 的环境变量	203
8.1.2 执行外部程序时候出现的异常	204
技巧 57 流和外部应用程序	205
8.1.3 外部应用程序的串联调用	206
技巧 58 在 shell 中执行命令	208
8.1.4 安全性和 shell 命令执行	209
技巧 59 分离子进程	210
8.1.5 父进程和子进程之间的 I/O 处理	211
8.1.6 引用计数和子进程	213
8.2 执行 Node 程序	213
技巧 60 执行 Node 程序	214
技巧 61 Forking Node 模块	216
技巧 62 运行作业	218
8.2.1 工作池	220
8.2.2 使用池模块	222
8.3 同步运行	223
技巧 63 同步子进程	223
8.4 总结	227

第二部分 实践中的技巧

9 网络：构建精简的网络应用.....	230
9.1 前端技术	231
技巧 64 快速的静态网站服务器	231
技巧 65 在 Node 中使用 DOM	236
技巧 66 在浏览器端使用 Node 模块	238
9.2 服务端技术	241
技巧 67 Express 路由分离	241
技巧 68 自动重启服务器	245
技巧 69 配置 web 应用	248
技巧 70 优雅地处理错误	253
技巧 71 RESTful web 应用	257
技巧 72 使用自定义的中间件	267
技巧 73 使用事件进行解耦	273
技巧 74 使用 WebSockets 来处理 sessions	276
技巧 75 升级 Express 3 到 4	281
9.3 web 应用程序的测试	285
技巧 76 测试路由	286
技巧 77 为中间件注入创建 seams	288
技巧 78 测试依赖远程服务的应用	291
9.4 全栈框架	297
9.5 实时服务	299
9.6 总结	300
10 测试：编写健壮代码的关键.....	301
10.1 Node 测试的相关介绍	303
10.2 使用断言编写简单的测试	304
技巧 79 用内置的模块编写测试	305
技巧 80 编写验证异常的测试	308
技巧 81 创建自定义的断言	312