

TURING

图灵程序设计丛书

MANNING

Node.js in Action

# Node.js

## 实战

Mike Cantelon

[美] Marc Harter 著

T. J. Holowaychuk

Nathan Rajlich

吴海星 译

Node.js核心框架贡献者代表作

Node.js项目负责人、Node包管理器  
(NPM) 作者力荐



人民邮电出版社  
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

Node.js in Action

# Node.js

## 实战

Mike Cantelon

[美] Marc Harter 著

T. J. Holowaychuk

Nathan Rajlich

吴海星 译



人民邮电出版社

北京

## 图书在版编目 (C I P) 数据

Node.js 实战 / (美) 坎特伦 (Cantelon, M.) 等著 ;  
吴海星译. — 北京 : 人民邮电出版社, 2014. 5  
(图灵程序设计丛书)  
ISBN 978-7-115-35246-0

I. ①N… II. ①坎… ②吴… III. ①JAVA语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2014)第063303号

## 内 容 提 要

本书是 Node.js 的实战教程, 涵盖了为开发高级 Node 应用程序所需要的一切特性、技巧以及相关理念。从搭建 Node 开发环境, 到一些简单的演示程序, 到开发复杂应用程序所必不可少的异步编程。书中还介绍了 HTTP API 的应用技巧等。

本书适合 Web 开发人员阅读。



- 
- ◆ 著 [美] Mike Cantelon Marc Harter T. J. Holowaychuk  
Nathan Rajlich
  - 译 吴海星
  - 责任编辑 李 璞
  - 责任印制 焦志炜
  - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号  
邮编 100164 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
三河市海波印务有限公司印刷
  - ◆ 开本: 800×1000 1/16  
印张: 22.25  
字数: 525千字 2014年5月第1版  
印数: 1-3 500册 2014年5月河北第1次印刷
- 著作权合同登记号 图字: 01-2013-8986号
- 

定价: 69.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

# 版权声明

Original English language edition, entitled *Node.js in Action* by Mike Cantelon, Marc Harter, T.J. Holowaychuk, Nathan Rajlich, published by Manning Publications. 178 South Hill Drive, Westampton, NJ 08060 USA. Copyright © 2014 by Manning Publications.

Simplified Chinese-language edition copyright © 2014 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Manning Publications授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。



# 序

写一本关于Node.js的书是一项很有挑战性的任务。这是一个相对新的平台，最近才刚刚趋于稳定。Node.js的核心一直在进化，并且社区中由用户创建的模块数量也呈现出了爆炸性的增长，其发展速度没人能跟得上。社区也仍然在找寻自己的声音。写书阐释这样一个还在不断发展的主题，唯一的办法是理解Node的本质，以及它为什么这样成功。这些Node.js老兵们就是这么做的。Mike Cantelon在Node社区中极其活跃，用Node做实验，谈论Node。关于Node适合做什么——可能更重要的是不适合做什么，他有着极深的洞见。T. J. Holowaychuk是最多产的Node.js模块作者之一，其中包括大规模流行的Web框架Express。Nathan Rajlich，也就是著名的TooTallNate，已经做了一段时间的Node.js核心代码的提交者，他也是平台发展到当前这种成熟状态的积极推动力量。

本书吸取了他们丰富的经验，带着你从最初的Node.js安装，到创建应用、调试程序和部署产品，一路走下去。你将了解到是什么让Node如此有趣，并从中窥见各位作者的理解，这样Node项目将来的发展方向也变得更好理解了。最重要的是，本书内容由浅入深、循序渐进，每一阶段都以之前所学的内容为基础。

Node是一个正在升起的火箭，作者们成功地将你带上了这一旅程。请将本书作为跳板，从这里出发，开拓你自己的视野吧。

Isaac Z. Schlueter  
Node包管理器（NPM）作者  
Node.js项目负责人

# 前 言

2011年初，Manning出版社找到我们，说想出一本关于Node.js的书，那时Node社区的状态和现在很不一样，圈子还很小。尽管已经有很多人表现出了对它的兴趣，但Node仍然被主流开发社区看做是一项有风险的技术，还没有人写过关于Node的书。尽管写书的想法令人生畏，但我们还是决定去大胆一试。

鉴于我们各自的开发方向不同，我们想不仅要把这本书的重点全放在Node的Web程序开发上，还要探索其他有趣的潜在用途。我们想给Web开发人员指出一条道路，用现有技术将异步开发带入服务器这一Node愿景。

这本书我们写了两年多，在写作过程中，这门技术已经进化了，所以我们也相应地做了更新。它现在变得更大了，很多成熟的公司也开始拥抱Node。

对于想做些不同尝试的Web程序开发人员，现在是学习Node的好时机，希望这本书可以帮到你，让你能迅速学会这门技术，并在其中找到乐趣。

# 致 谢

感谢Manning出版社那些优秀的人们在本书出版过程中所发挥的作用。Renae Gregoire在其中扮演了重要角色，在他的督促下，我们才能写出雄辩、清晰、高品质的内容。Bert Bates帮助定义了本书的视觉感受，跟我们一起设计了书中的各种图形来表示不同的概念。Marjan Bace和Michael Stephens给予了我们充分的信任，委托我们来写这本书，并协助推动项目前行。还有Manning出版社的编辑、生产和技术职员们，我们合作得非常愉快。

在成书的各个阶段，很多人参与了书稿的评审工作，我们也要对他们的反馈表示感谢。包括在本书的在线论坛中发表评论及指出书中错误的MEAP读者，还有下面这些评审者，他们多次阅读书稿，其见解和评论让本书变得更好，他们是：Alex Madurell、Bert Thomas、Bradley Meck、Braj Panda、Brian L. Cooley、Brian Del Vecchio、Brian Dillard、Brian Ehmann、Brian Falk、Daniel Bretoi、Gary Ewan Park、Jeremy Martin、Jeroen Nouws、Jeroen Trap pers、Kassandra Perch、Kevin Baister、Michael Piscatello、Patrick Steger、Paul Stack和Scott Banachowski。

还要感谢Valentin Crettaz和Michael Levin，就在这本书即将出版之前，他们对最终书稿做了认真的技术校对。最后同样重要的，我们还要感谢Node项目的负责人Isaac Schlueter为本书作序。

## MIKE CANTELON的致谢

我要感谢我的朋友Joshua Paul，他将我带入开源的世界，给了我在科技领域的第一次突破，并鼓励我写本书。

还要感谢我的伴侣Malcolm，她在我写书期间一直给我鼓励。当我因为写书一直闷在家里时，她很耐心地陪伴我。还要特别感谢我的父母，培育了我乐于创造和勇于探索的热情，并忍受了我那发展不太均衡的童年时期对8位机的痴迷。还要感谢我的祖父母，送给我一台让我一生热衷于编程的机器：Commodore 64。

在编写本书的过程中，T. J.和Nathan的专业知识是无价之宝，他们的幽默感更是值得赞扬。感谢他们如此信任我们，并同意一起合作。Marc Harter也提供了巨大的帮助，他参与了编辑、校对和内容的撰写这些艰巨的任务，这些任务加起来真的很耗费精力。

## MARC HARTER的致谢

感谢Ryan Dahl，几乎在四年前就激励我认真对待服务器端JavaScript编程。感谢Ben Noordhuis，

给了我Node内部运作的宝贵资源。感谢Bert Bates，信任我、挑战我，写作过程中总是愿意施以援手。感谢Mike、Nate和T. J.在关键时刻欢迎我加入，跟他们合作是我的荣幸。特别要感谢我的妻子，同时也是我的好朋友Hannah，她的鼓励让我得以加入并顺利完成这次新的合作。

### NATHAN RAJLICH的致谢

首先要感谢Guillermo Rauch，他带我进入了Node.js社区，并帮我找到了自己的位置。还要感谢David Blickstein鼓励我加入本书的创作项目。感谢Ryan Dahl开启了Node.js之门，还要感谢最近几年一直在出色掌舵的Isaac Schlueter。也要感谢我的家庭、我的朋友，还有我的女朋友，忍受了我在这一过程中的不眠之夜和各种各样的情绪。当然，特别要感谢我的父母这么多年来对我痴迷于计算机的巨大支持。如果不是他们陪在身边，我不会取得今天的成绩。



# 关于本书

本书的主要目的是教你学会如何创建和部署Node程序，重点是Web程序。本书中有相当一部分内容集中介绍了Web程序框架Express和中间件框架Connect，主要是因为它们的用途和社区的支持。你还会学到如何创建自动化测试，以及如何部署你的程序。

本书面向希望用Node.js创建响应式、可伸缩程序的有经验的Web程序开发人员。

因为Node.js程序是用JavaScript写的，所以需要你掌握这门语言。此外最好还要熟悉Windows、OS X或Linux命令行。

## 路线图

本书被分为三部分。

第一部分介绍了Node.js，教授了一些用它做开发所需要的基础技术。第1章阐述了Node的特征，并给出了一些示例代码。第2章指导读者创建了一个示例程序。第3章阐述了Node.js开发的困难之处，以及可以用来克服这些困难的技术，并给出了组织程序代码的办法。

第二部分在本书中所占比重最大，主要讨论Web程序开发。第4章讲了基于Node创建Web程序的基础知识，第5章讨论了如何用Node存储程序数据。

然后第二部分继续深入Web相关框架。第6章介绍了Connect框架，阐述了它的好处和它的工作机制。第7章讲述了Connect框架内置的各种组件，以及如何用它们给Web程序添加功能。第8章介绍了Express框架。第9章指导读者体验Express的高级用法。

涵盖了Web开发的基础知识后，第二部分又探索了两个相关的主题。第10章指导读者使用各种Node测试框架，第11章讲了在Node Web程序中如何用模板将数据展示从逻辑中分离出来。

第三部分转而讨论了可以用Node完成的Web开发之外的事情。第12章讨论了如何把Node程序部署到生产服务器上、如何维护在线时间，以及如何将性能提升到最优。第13章阐述了如何创建非HTTP程序，如何用Socket.io框架创建实时程序，以及如何使用一些便利的Node内置API。第14章讨论了Node社区的工作机制，以及如何用Node包管理器发布Node作品。

## 代码约定及下载

本书代码遵循通用的JavaScript编码规范。用空格，而不是制表符做代码缩进。避免一行代码超过80个字符。代码清单中，很多代码都加上了注释，指出了其中关键的概念。

每行一条语句，并在简单语句末尾添加分号。对于代码块，一条或多条语句放在大括号里，左边大括号放在块开头行的末尾，右边大括号缩进到跟块开头行竖直对齐的位置。

你可以从Manning出版社的网站上下载书中的示例代码，下载地址在[www.manning.com/Node.jsinAction](http://www.manning.com/Node.jsinAction)。

### 作者在线

购买本书英文版的读者可以免费访问由Manning出版社维护的专用Web论坛，并在论坛中对该书发表评论、询问技术问题、从作者和其他用户那里得到帮助。要访问并订阅该论坛，请访问[www.manning.com/Node.jsinAction](http://www.manning.com/Node.jsinAction)。这个页面介绍了注册后如何访问论坛、可以得到什么帮助以及论坛上的行为准则。

Manning致力于为我们的读者提供一个场所，让读者之间，以及读者和作者之间进行有意义的对话。但我们并不会强制作者参与，他们在论坛上的贡献是自愿而且不收费的。我们建议你尽量问作者一些有挑战性的问题，以激发他们的兴趣！

只要本书英文版仍然在售，读者就能从出版社的网站上访问作者在线论坛和之前讨论话题的归档。<sup>①</sup>

---

<sup>①</sup> 本书中文版请访问<http://www.ituring.com.cn/book/1061>

## 关于封面图片

本书封面上的画像标题为“城镇里的男人”，摘自19世纪法国出版的沙利文·马雷夏尔（Sylvain Maréchal）四卷本的地域服饰风俗纲要。其中每幅插图都是手工精心绘制并上色的。马雷夏尔这套书展示的丰富服饰，令我们强烈感受到200年前乡村与城镇的巨大文化差异。不同地域的人山水阻隔，言语不通。无论奔走于街巷，还是驻足于乡间，通过他们的服饰，一眼就能看出他们的生活场所、职业，以及生活境况。

时过境迁，书中描绘的那些区域性服饰差异到如今已经不复存在。即使是不同国家，都很难再看出人们着装的区别，再不必说城镇和乡村了。或许，我们今天多姿多彩的人生，正是从前那些文化差异的体现。只不过，如今的生活更加多元，而且技术环境下的生活节奏也更快了。

今时今日，计算机图书层出不穷，Manning就以马雷夏尔这套书中多样性的图片，来表达对IT行业日新月异的发明与创造的赞美。

# 目 录

## 第一部分 Node基础

第 1 章 欢迎进入 Node.js 世界	2
1.1 构建于 JavaScript 之上	3
1.2 异步和事件触发：浏览器	4
1.3 异步和事件触发：服务器	5
1.4 DIRT 程序	6
1.5 默认 DIRT	8
1.5.1 简单的异步程序	9
1.5.2 Hello World HTTP 服务器	10
1.5.3 流数据	10
1.6 小结	11
第 2 章 构建有多个房间的聊天室程序	12
2.1 程序概览	12
2.2 程序需求及初始设置	15
2.2.1 提供 HTTP 和 WebSocket 服务	15
2.2.2 创建程序的文件结构	16
2.2.3 指明依赖项	16
2.2.4 安装依赖项	17
2.3 提供 HTML、CSS 和客户端 JavaScript 的服务	18
2.3.1 创建静态文件服务器	19
2.3.2 添加 HTML 和 CSS 文件	21
2.4 用 Socket.IO 处理与聊天相关的消息	23
2.4.1 设置 Socket.IO 服务器	24
2.4.2 处理程序场景及事件	25
2.5 在程序的用户界面上使用客户端 JavaScript	29

2.5.1 将消息和昵称/房间变更请求传给服务器	29
2.5.2 在用户界面中显示消息及可用房间	30
2.6 小结	34
第 3 章 Node 编程基础	35
3.1 Node 功能的组织及重用	36
3.1.1 创建模块	37
3.1.2 用 module.exports 微调模块的创建	39
3.1.3 用 node_modules 重用模块	40
3.1.4 注意事项	41
3.2 异步编程技术	42
3.2.1 用回调处理一次性事件	43
3.2.2 用事件发射器处理重复性事件	46
3.2.3 异步开发的难题	53
3.3 异步逻辑的顺序化	54
3.3.1 什么时候使用串行流程控制	55
3.3.2 实现串行化流程控制	56
3.3.3 实现并行化流程控制	58
3.3.4 利用社区里的工具	60
3.4 小结	61

## 第二部分 用Node开发Web程序

第 4 章 构建 Node Web 程序	64
4.1 HTTP 服务器的基础知识	64
4.1.1 Node 如何向开发者呈现 HTTP 请求	65

4.1.2	一个用“Hello World”做响应的 HTTP 服务器	67	6.3.1	中间件什么时候不调用 next()	114
4.1.3	读取请求头及设定响应头	67	6.3.2	用中间件的顺序执行认证	115
4.1.4	设定 HTTP 响应的状态码	67	6.4	挂载中间件和服务器	115
4.2	构建 RESTful Web 服务	68	6.4.1	认证中间件	116
4.2.1	用 POST 请求创建资源	69	6.4.2	显示管理面板的中间件	117
4.2.2	用 GET 请求获取资源	71	6.5	创建可配置中间件	118
4.2.3	用 DELETE 请求移除资源	72	6.5.1	创建可配置的 logger 中间件组件	119
4.3	提供静态文件服务	73	6.5.2	构建路由由中间件组件	120
4.3.1	创建一个静态文件服务器	73	6.5.3	构建一个重写 URL 的中间件组件	122
4.3.2	处理服务器错误	77	6.6	使用错误处理中间件	123
4.3.3	用 fs.stat() 实现先发制人的错误处理	77	6.6.1	Connect 的默认错误处理器	124
4.4	从表单中接受用户输入	78	6.6.2	自行处理程序错误	124
4.4.1	处理提交的表单域	78	6.6.3	使用多个错误处理中间件组件	125
4.4.2	用 formidable 处理上传的文件	81	6.7	小结	129
4.4.3	计算上传进度	84	第 7 章	Connect 自带的中间件	130
4.5	用 HTTPS 加强程序的安全性	85	7.1	解析 cookie、请求主体和查询字符串的中间件	131
4.6	小结	86	7.1.1	cookieParser(): 解析 HTTP cookie	131
第 5 章	存储 Node 程序中的数据	87	7.1.2	bodyParser(): 解析请求主体	134
5.1	无服务器的数据存储	88	7.1.3	limit(): 请求主体的限制	135
5.1.1	内存存储	88	7.1.4	query(): 查询字符串解析	137
5.1.2	基于文件的存储	88	7.2	实现 Web 程序核心功能的中间件	138
5.2	关系型数据库管理系统	91	7.2.1	logger(): 记录请求	138
5.2.1	MySQL	91	7.2.2	favicon(): 提供 favicon	140
5.2.2	PostgreSQL	99	7.2.3	methodOverride(): 伪造 HTTP 方法	141
5.3	NoSQL 数据库	100	7.2.4	vhost(): 虚拟主机	143
5.3.1	Redis	101	7.2.5	session(): 会话管理	144
5.3.2	MongoDB	105	7.3	处理 Web 程序安全的中间件	148
5.3.3	Mongoose	107	7.3.1	basicAuth(): HTTP 基本认证	148
5.4	小结	109	7.3.2	csrf(): 跨站请求伪造防护	150
第 6 章	Connect	110	7.3.3	errorHandler(): 开发错误处理	150
6.1	搭建一个 Connect 程序	111			
6.2	Connect 的工作机制	112			
6.2.1	做日志的中间件	112			
6.2.2	响应“hello world”的中间件	113			
6.3	为什么中间件的顺序很重要	114			

7.4 提供静态文件服务的中间件	152	9.4.1 处理 404 错误	215
7.4.1 static(): 静态文件服务	152	9.4.2 处理错误	217
7.4.2 compress(): 压缩静态文件	154	9.5 小结	220
7.4.3 directory(): 目录列表	156	<b>第 10 章 测试 Node 程序</b>	221
7.5 小结	157	10.1 单元测试	222
<b>第 8 章 Express</b>	158	10.1.1 assert 模块	222
8.1 生成程序骨架	160	10.1.2 Nodeunit	225
8.1.1 安装 Express 的可执行程序	161	10.1.3 Mocha	227
8.1.2 生成程序	162	10.1.4 Vows	232
8.1.3 探索程序	162	10.1.5 should.js	234
8.2 配置 Express 和你的程序	164	10.2 验收测试	235
8.3 渲染视图	166	10.2.1 Tobi	236
8.3.1 视图系统配置	167	10.2.2 Soda	237
8.3.2 视图查找	169	10.3 小结	239
8.3.3 把数据输出到视图中	171	<b>第 11 章 Web 程序模板</b>	240
8.4 处理表单和文件上传	175	11.1 用模板保持代码的整洁性	240
8.4.1 实现照片模型	175	11.2 嵌入 JavaScript 的模板	244
8.4.2 创建照片上传表单	176	11.2.1 创建模板	245
8.4.3 显示上传照片列表	178	11.2.2 用 EJS 过滤器处理模板数据	246
8.5 创建资源下载	179	11.2.3 将 EJS 集成到你的程序中	249
8.5.1 创建照片下载路由	179	11.2.4 在客户端程序中使用 EJS	250
8.5.2 实现照片下载路由	180	11.3 使用 Mustache 模板语言与 Hogan	251
8.6 小结	182	11.3.1 创建模板	251
<b>第 9 章 Express 进阶</b>	183	11.3.2 Mustache 标签	252
9.1 认证用户	184	11.3.3 微调 Hogan	254
9.1.1 保存和加载用户	184	11.4 用 Jade 做模板	255
9.1.2 注册新用户	189	11.4.1 Jade 基础知识	256
9.1.3 已注册用户登录	194	11.4.2 Jade 模板中的逻辑	258
9.1.4 用户加载中间件	197	11.4.3 组织 Jade 模板	260
9.2 先进的路由技术	199	11.5 小结	264
9.2.1 校验用户内容提交	199	<b>第三部分 在 Node 中更进一步</b>	
9.2.2 特定路由由中间件	202	<b>第 12 章 部署 Node 程序并维持正常运行时间</b>	266
9.2.3 实现分页	205	12.1 安置 Node 程序	266
9.3 创建一个公开的 REST API	208	12.1.1 专用的和虚拟私有服务器	267
9.3.1 设计 API	208	12.1.2 云主机	268
9.3.2 添加基本的认证	209		
9.3.3 实现路由	209		
9.3.4 启用内容协商	212		
9.4 错误处理	214		



12.2	部署的基础知识	269	13.3.3	繁衍外部进程	296
12.2.1	从 Git 存储库部署	270	13.4	开发命令行工具	301
12.2.2	让 Node 保持运行	270	13.4.1	解析命令行参数	301
12.3	让正常运行时间和性能达到最优	271	13.4.2	处理 stdin 和 stdout	302
12.3.1	用 Upstart 维护正常运行时间	272	13.4.3	添加彩色的输出	304
12.3.2	集群 API: 利用多核的优势	273	13.5	小结	306
12.3.3	静态文件及代理	275	<b>第 14 章 Node 生态系统</b>		<b>307</b>
12.4	小结	277	14.1	给 Node 开发人员的在线资源	308
<b>第 13 章 超越 Web 服务器</b>		<b>278</b>	14.1.1	Node 和模块的参考资料	308
13.1	Socket.IO	278	14.1.2	Google 群组	309
13.1.1	创建一个最小的 Socket.IO 程序	279	14.1.3	IRC	309
13.1.2	用 Socket.IO 触发页面和 CSS 的重新加载	281	14.1.4	GitHub 问题列表	310
13.1.3	Socket.IO 的其他用法	283	14.2	GitHub	310
13.2	深入 TCP/IP 网络	284	14.2.1	GitHub 入门	311
13.2.1	处理缓冲区和二进制数据	284	14.2.2	添加一个项目到 GitHub 中	312
13.2.2	创建 TCP 服务器	286	14.2.3	用 GitHub 协作	314
13.2.3	创建 TCP 客户端	289	14.3	为 npm 库做贡献	316
13.3	跟操作系统交互的工具	290	14.3.1	准备包	317
13.3.1	单例的全局 process 对象	291	14.3.2	编写包规范	317
13.3.2	使用文件系统模块	293	14.3.3	测试和发布包	318
			14.4	小结	320
			<b>附录 A</b>	<b>安装 Node 和社区附加组件</b>	<b>321</b>
			<b>附录 B</b>	<b>调试 Node</b>	<b>329</b>
			<b>附录 C</b>	<b>Express 的扩展及配置</b>	<b>336</b>

# Part 1

## 第一部分

# Node 基础

学习编程语言或框架时，经常会碰到新概念，让你以一种新的方式思考问题。Node也不例外，因为它对应用程序开发的几个方面采取了一种全新的方式。

本书的第一部分会勾勒出Node与其他平台的差异，并且讲解它的基本用法。你会看到用Node创建的应用程序长什么样子，如何组织，以及如何处理Node特有的开发困难。你在第一部分所学的知识将成为本书后续内容的基础，即第二部分详细介绍的如何用Node创建Web程序，以及第三部分讨论的如何创建非Web程序。

### 本章内容

- Node.js是什么
- 服务端JavaScript
- Node的异步和事件触发本质
- Node为谁而生
- Node程序示例

Node.js是什么? 你很可能已经有所耳闻,甚至已经用上了,你也有可能对它很好奇。现在Node还很年轻(它的首次亮相是在2009年),却非常流行。它在Github受关注项目排行榜上位列第二(<https://github.com/joyent/node>),在Google小组(<http://groups.google.com/group/nodejs>)和IRC频道(<http://webchat.freenode.net/?channels=node.js>)中都有很多追随者,并且社区同仁们在NPM包管理网站(<http://npmjs.org>)上发布的模块多达15 000多个。所有这些都足以表明这个平台的强大吸引力。

Node创始人Ryan Dahl 2009年柏林JSCONF的网站上有Node创始人Ryan Dahl第一次介绍Node的视频:[http://jsconf.eu/2009/video\\_nodejs\\_by\\_ryan\\_dahl.html](http://jsconf.eu/2009/video_nodejs_by_ryan_dahl.html)<sup>①</sup>。

官网上(<http://www.nodejs.org>)给Node下的定义是:“一个搭建在Chrome JavaScript运行时的平台,用于构建高速、可伸缩的网络程序。Node.js采用的事件驱动、非阻塞I/O模型,使它既轻量又高效,并成为构建运行在分布式设备上的数据密集型实时程序的完美选择。”

我们在本章中会看到下面这些概念:

- 为什么JavaScript对服务端开发很重要;
- 浏览器如何用JavaScript处理I/O;
- Node在服务端如何处理I/O;
- DIRT程序是什么意思,为什么适于用Node开发;
- 几个基础的Node程序示例。

<sup>①</sup> 已经看不到了,被拿掉了。——译者注