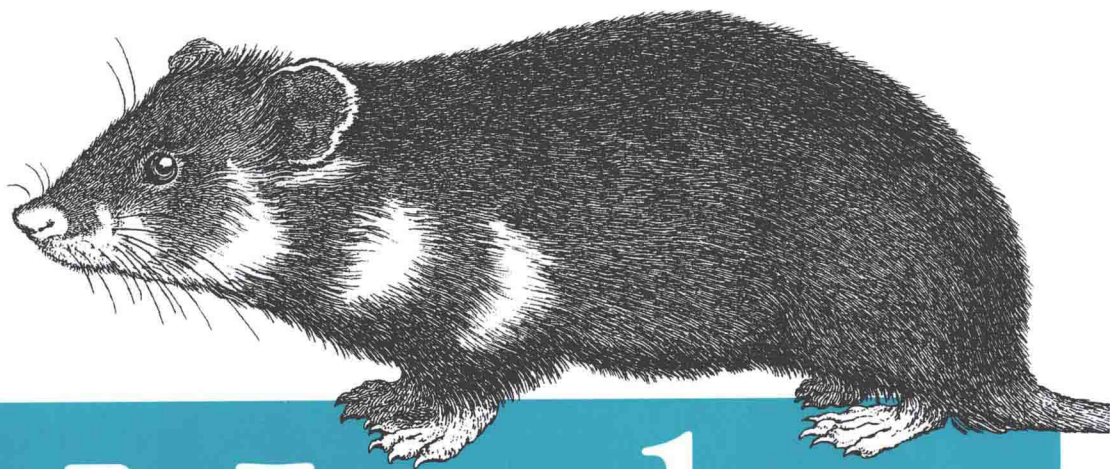


*Learning Node*



Node

学习指南

O'REILLY®

[美] *Shelley Powers* 著  
夏思雨 高亮 译

 人民邮电出版社  
POSTS & TELECOM PRESS

# Node 学习指南

[美] Shelley Powers 著

夏思雨 高亮 译

人民邮电出版社

北京

## 图书在版编目 (C I P) 数据

Node学习指南 / (美) 鲍尔丝 (Powers, S.) 著 ; 夏思雨, 高亮译. -- 北京 : 人民邮电出版社, 2014. 3  
ISBN 978-7-115-33796-2

I. ①N… II. ①鲍… ②夏… ③高… III. ①  
JAVA语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2013)第277709号

## 版权声明

Copyright© 2012 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2014. Authorized translation of the English edition, 2014 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

本书中文简体版由 O'Reilly Media, Inc. 授权人民邮电出版社出版。未经出版者书面许可, 对本书的任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

- 
- ◆ 著 [美] Shelley Powers
  - 译 夏思雨 高亮
  - 责任编辑 陈冀康
  - 责任印制 程彦红 杨林杰
  - ◆ 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: 787×1000 1/16  
印张: 23.75  
字数: 451千字  
印数: 1-3 000册
- 2014年3月第1版  
2014年3月河北第1次印刷



---

著作权合同登记号 图字: 01-2013-3678 号

定价: 69.00 元

读者服务热线: (010)81055410 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

# 内 容 提 要

Node.js 是一套用来编写高性能网络服务器的 JavaScript 工具包。它可以让 JavaScript 在服务器端运行，因此，可用来快速构建网络服务及应用的平台。

本书是学习 Node 编程的入门指南。全书共 16 章。前 4 章主要介绍 Node 基本知识，包管理工具（npm）的安装和使用等。第 5 章介绍了 Node 处理异步开发的独特的实现方式等。第 6~8 章，讲解了路由、代理、Web 服务器、中间件等基本概念，包括 Express。第 9 章到第 11 章分别介绍了基于 Redis、MongoDB 以及关系型数据库的 Node 应用开发。第 12 章到第 14 章分别介绍了图形和媒体、Sockets.io 模块、调试和测试等主题。第 15 章介绍了安全和权限的问题，第 16 章介绍了 Node 应用的扩展和部署。

本书适合有一定基础的 JavaScript 程序员阅读，也适合对学习 Node 应用开发感兴趣的读者学习参考。

## 非同寻常的 JavaScript

目前正是学习 Node 的好时机。

Node 相关的技术依然年轻充满生机，经常出现有趣的变化和改动。同时，这项技术也达到了一定的成熟度，可以确保你在学习 Node 上花费的时间是值得的：即使在 Windows 上安装也非常简单；从成百上千的可用模块中涌现出了最佳组合模块；对于产品环境来说这种结构足够健壮。

当使用 Node 时需要记得两个要点。第一，Node 是基于 JavaScript 的，与你之前用于客户端开发的 JavaScript 多少有些类似。当然，你也可以使用另一种变形的语言，如 CoffeeScript，但是 JavaScript 是通用的语言。

第二个需要注意的要点是，Node 并不是常规的 JavaScript。它是一门服务器端的技术，这意味着很多你在浏览器环境中认为应该有的功能——如保护措施——都不会出现在这里，但也会有很多其他新的不熟悉的功能。

当然，如果 Node 和浏览器端的 JavaScript 一样的话，那有什么乐趣呢？

## 为什么是 Node

如果你想要看 Node 的源码，你可以找一下 Google V8 的源代码。Google V8 是 JavaScript 引擎（从技术角度来讲，是 ECMAScript），也是 Google Chrome 浏览器的核心。那么，Node.js 的一个优点是你可以为一种 JavaScript 实现开发 Node 程序，而不是一大堆不同版本的不同浏览器。

Node 被设计用于那些需要频繁 I/O 操作，但计算量不大的程序。更重要的是，它提供的这个功能是直接可用的。在等待一个文件加载完成或者数据库更新的过程中，不需要担心程序会阻塞其他进程，因为 Node 中大部分功能默认都是 I/O 异步的；也不需要担心线程的工作，因为 Node 的实现是单线程的。



异步 I/O 意味着程序并不会等待输入/输出操作处理完成之后才处理代码中的下一个步骤。第 1 章会介绍更多 Node 异步特性的细节。

更重要的一点是，Node 是由很多传统 Web 开发人员都熟悉的语言 JavaScript 编写的。你会学习到如何使用新的技术，如 WebSocket 或者基于 Express 这种框架进行开发，但是至少你不需要在学习新概念的同时学习一门新的语言。对语言的熟悉使你可以只专注新的特性。

## 本书的目标读者

使用 Node 的一个挑战就是假设学习 Node 的部分人有 Ruby 或者 Python 背景，或者使用过 Redis。我没有假定这一点，所以在解释 Node 组件时我不会说这就“像 Sinatra 一样”。

这本书唯一的假设就是读者使用过 JavaScript 并且喜欢它。你并不需要是个专家，但是你需要在我提到“闭包”的时候知道我在说什么，并且使用过 Ajax 以及对客户端环境的事件处理比较熟悉。如果你做过传统的 Web 开发，熟悉一些概念，如 HTTP 方法（GET、POST）、Web session、cookie 等，你会从本书中获益良多。除了这些，你需要熟悉 Windows 控制台，或者 Unix、Linux、MAC OSX 的命令行。

如果你对一些新技术感兴趣，诸如 WebSocket 或者使用架构创建程序，就会喜欢这本书的。我通过这些方面向你介绍如何在现实世界使用 Node。

最重要的一点，在你阅读本书时要保持思维开放，要有思想准备、你可能碰上版本不成熟的问题，也可能会撞上这种处在发展中的技术陷阱。无论如何，带着期待开始你的学习旅程吧，因为这是一个很有趣的过程。



如果你不确定你达到了“熟悉”JavaScript 的标准，可以查看一下我对 JavaScript 的介绍：*Learning JavaScript*，第二版（O’Reilly）。

## 怎么更好地使用本书

如果你不想按顺序阅读本书，有一些途径可供选择，这取决于你想要知道些什么以及你有多少 Node 经验。

如果你从来没有用过 Node，建议你从第 1 章开始阅读至少到第 5 章。这几章主要讲了 Node 基本知识、包管理工具（npm）安装、如何使用、创建你的第一个程序

以及可用的模块。第 5 章还涉及了一些 Node 中的样式问题，包括 Node 处理异步开发的独特实现方式。

如果你有一点 Node 经验，使用过内建和第三方的一些模块，也熟悉 REPL (read-eval-print loop, 交互控制台)，可以跳过第 1 章到第 4 章，但是我建议第 5 章还是必读的。

在本书中，我使用了 Express 架构，Express 又使用了 Connect 中间件。如果你没用过 Express，请阅读第 6~第 8 章，这几章讲解了路由、代理、Web 服务器、中间件等基本概念，包括 Express。尤其是如果你好奇在 MVC (Model-View-Controller) 框架中如何使用 Express，一定要阅读第 7 章和第 8 章。

在这些基础章节之后，你可以有选择地进行阅读。比如，如果你主要使用 key/value (键值对)，你应该阅读第 9 章关于 Redis 的讨论。如果你对基于文档的数据感兴趣，查阅第 10 章，这一章介绍了如何在 Node 中使用 MongoDB。当然，如果你只需要使用关系型数据库，可以直接跳过 Redis 和 MongoDB 到第 11 章，记得经常检查数据库的更新——它们可能提供了一种处理数据的新视角。

在这三章关于数据的讲解之后，我们会介绍具体程序的使用。第 12 章主要关注于图形和媒体访问，包括如何提供 HTML5 视频媒体，以及使用 Canvas 和 PDF 文件。第 13 章主要讲了非常流行的 Sockets.io 模块，特别是如何使用新的 web socket 功能。

在第 12 章和第 13 章两章关于 Node 两种不同的具体用法介绍之后，本书末尾再次回到同一点上。当你在其他章节花费了一些时间尝试示例之后，需要读一下第 14 章，深入了解如何调试和测试 Node 程序。

第 15 章可能是最难的一章，也更重要。这一章主要介绍了安全和权限的问题。我不建议把这章作为你阅读的第一章，但是在你发布一个 Node 应用之前需要花点时间读一下这一章节。

第 16 章是最后一章，不管你兴趣是什么、经验多少，你都可以很放心地把它留到最后。这章主要介绍如何使你的产品上线，包括如何在流行的云服务上或者你自己的系统中部署 Node 程序，如何确保你的程序可以与其他 Web 服务器兼容，比如 Apache，如何确保你的程序在崩溃或者系统重启之后自动重启。

Node 与 Git 版本控制联系紧密，并且绝大部分（如果不是全部的话）的 Node 模块都在 GitHub 上。附录里为那些不了解 Git/GitHub 的人提供了教程。

虽然我之前说不需要按章节进行阅读，但是我建议你还是按顺序阅读。很多章节的

工作都是在前一章的基础上完成，如果你跳跃阅读的话可能会错过一些重点。尽管本书有很多独立的程序示例，但是我主要使用了一个简单的 Express 程序——叫做 Widget Factory，从第 7 章开始，在剩下的章节中都有涉及。我相信如果你从开始阅读，跳过那些你了解的小节会比跳过整个章节好很多。

就像《爱丽丝漫游奇境记》里的国王说的一样：从“开头开始，一直念到末尾，然后停止。”

## 技术

本书中的示例是在 Node 0.6.x 不同的发布版本中创建的，大部分都在 Linux 环境中测试过，应该在各种 Node 环境中都可以正常工作。

Node 0.8.x 发布的时候本书刚刚上市。大部分示例都兼容 Node 0.8.x。有一些例子需要做一些修改以兼容新版 Node，我在书中也有指出。

## 示例

本书中的所有示例在 O'Reilly 官网本书主页上 ([http://oreil.ly/Learning\\_node](http://oreil.ly/Learning_node)) 有压缩文件可以下载。下载解压，安装好 Node 之后，可以切换 examples 目录，输入以下命令来安装示例需要的所有依赖：

```
npm install-d
```

在第 4 章中我会讲到更多关于 npm 的用法。

## 本书中的体例

以下是本书中使用的印刷体例：

文本

菜单标题，菜单选项，菜单按钮，键盘快捷键（比如 Alt 和 Ctrl）。

等宽字体

命令，选项，switch，变量，属性，key，函数，类型，类，命名空间，方法，模块，参数，值，对象，事件，事件处理，XML 标记，HTML 标记，宏，文件内容或者命令行输出。

等宽粗体

显示命令或者其他应该由用户输入的文本。



等宽斜体

显示应该由用户提供的內容替换的文本。



这个图标表示提示、建议或者一般的记录。



这个图标表示一个警告或者提醒。

## 使用代码示例

本书有助于你完成工作。一般来说，可以在程序或者文档中使用本书的代码。关于权限的问题你并不需要联系我们，除非大量复制代码。比如，使用本书中的几段代码编写一段程序并不需要我们的允许，但是如果使用 O’ Reilly 书中的代码买卖或者制成 CD，则需要我们允许才可以。将本书中的大量代码作为你的产品文档也需要权限。

我们非常感激你在引用时标明出处，但是并不强制。引用包括标题、作者、出版商、ISBN。比如：“Learning Node by Shelley Powers (O’Reilly). Copyright 2012 Shelley Powers, 978-1-449-32307-3”。

如果你觉得你的使用不在上述范围之内，请联系我们 [permissions@oreilly.com](mailto:permissions@oreilly.com)。

## Safari Bookd Online

Safari Books Online ([www.safaribooksonline.com](http://www.safaribooksonline.com)) 是一个即时的电子图书馆，传递全世界在技术和商业领域著名作者的书籍和视频。

技术专家、软件开发人员、网站设计、商业和创意专家都使用 Safari Books Online 作为他们搜索、解决问题，学习和培训的主要资源。

Safari Books Online 为组织、政府代理和个人提供了不同的产品搭配组合和定价。购买者可以访问数以千计的书籍、培训视频，以及从出版商的数据库访问还未发布的手稿，比如 O’ Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、

FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology，等等。更多信息请访问我们的网站。

## 联系我们

有关本书的提问和评价请联系出版商：

O'Reilly Media, Inc.

1005 Gravenstein Highway North

Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室 (100035)

奥莱利技术咨询（北京）有限公司

我们为本书提供了网页列出勘误表、示例，以及其他附加信息。可以访问：  
[http://oreil.ly/Learning\\_node](http://oreil.ly/Learning_node)。

关于本书的评价或者任何技术问题，请发送邮件到 [bookquestions@oreily.com](mailto:bookquestions@oreily.com)。

更多有关书籍、课程、会议和最新信息，请访问我们的网站：<http://www.oreily.com>。

Facebook: <http://facebook.com/oreilly>

Twitter: <http://twitter.com/oreillymedia>

YouTube: <http://www.youtube.com/oreillymedia>

## 感谢

感谢所有的朋友和家人在我写书的时候帮助我保持清醒。特别感谢我的编辑 Simon St. Laurent，经常听我发牢骚。

还要感谢整个产品组帮助这本书从一个想法变成一个实物：Rachel Steely、Rachel Monaghan、Kiel Van Horn、Aaron Hazelton 和 Rebecca Demarest。

当你使用 Node 时，你会接收很多的帮助来自于 Node.js 创始人 Ryan Dahl，npm 创造者 Issac Schlueter，也是现在的 Node.js 掌门人。

其他为本书提供了有用的代码和模块的是：Bert Belder、TJ Holowaychuk、Jeremy

Ashkenas、Mikeal Rogers、Guillermo Rauch、Jared Hanson、Felix Geisendörfer、Steve Sanderson、Matt Ranney、Caolan McMahon、Remy Sharp、Chris O' Hara、Mariano Iglesias、Marco Aurelio、Damian Suarez、Jeremy Ashkenas、Nathan Rajlich、Christian Amor Kvalheim 和 Gianni Chiappetta。如果有任何漏掉没有提到的模块开发人员，我在此表示歉意。

如果没有那些好心人提供教程、做法以及有用的指导，本书还有些什么呢？感谢 Tim Caswell、Felix Geisendorfer、Mikato Takada、Geo Paul、Manuel Kiessling、Scott Hanselman、Peter Krumins、Tom Hughes-Croucher、Ben Nadel，以及 Nodejitsu 和 Joyent 的全体人员。

# 目录

第 1 章 Node.js: 启动与运行	1
1.1 搭建 Node 开发环境	2
1.1.1 Linux (Ubuntu) 下安装 Node	2
1.1.2 Windows 7 平台下 Node+WebMatrix	4
1.1.3 升级 Node	8
1.2 开始 Node 开发	9
1.2.1 Hello, World in Node	9
1.2.2 分析 “Hello,World”	11
1.3 异步函数及 Node 事件循环	13
1.3.1 使用异步方式读取文件	14
1.3.2 观察异步程序流程	15
1.4 Node 的优势	19
第 2 章 Node 与 REPL	20
2.1 REPL: 先睹为快和未定义的表达式	20
2.2 REPL 的优势: 更好地理解表层之下的 JavaScript	22
2.3 多行以及更复杂的 JavaScript	23
2.3.1 REPL 命令	26
2.3.2 REPL 和 rlwrap	27
2.3.3 定制 REPL	28
2.4 不可预计的意外——记得经常保存	32
第 3 章 Node 核心库	33
3.1 全局对象: global、process 和 Buffer	34
3.1.1 global	34
3.1.2 process	36
3.1.3 Buffer	38
3.2 定时器: setTimeout、clearTimeout、setInterval 和 clearInterval	39
3.3 Servers、Streams 和 Sockets	40
3.3.1 TCP Sockets 和 Servers	41
3.3.2 HTTP	43
3.3.3 UDP 数据报套接字	45

3.3.4	流、管道和 Readline	47
3.4	子进程	49
3.4.1	child_process.spawn	50
3.4.2	child_process.exec 和 child_process.execFile	52
3.4.3	child_process.fork	52
3.4.4	在 Windows 系统中使用子进程	53
3.5	域名解析和 URL 处理	54
3.6	Utilities 模块和对象继承	55
3.7	Events 和 EventEmitter	59
<b>第 4 章</b>	<b>Node 模块系统</b>	<b>63</b>
4.1	使用 require 和默认路径加载模块	63
4.2	外部模块和 Node 包管理工具	65
4.3	如何找到你需要的模块	69
4.3.1	Colors: 简单至上	71
4.3.2	Optimist: 另一个简单的小模块	73
4.3.3	Underscore	74
4.4	创建自定义模块	75
4.4.1	打包整个目录	76
4.4.2	为你的模块发布做准备	76
4.4.3	发布模块	80
<b>第 5 章</b>	<b>控制流、异步模式和异常处理</b>	<b>82</b>
5.1	使用 Callback 而不使用 Promises	82
5.2	顺序调用、嵌套回调、异常捕获	85
5.3	异步模式和控制流模块	92
5.3.1	Step	93
5.3.2	Async	96
5.4	Node 编码风格	101
<b>第 6 章</b>	<b>路由寻址、服务文件和中间件</b>	<b>103</b>
6.1	从头开始: 创建一个简单的静态文件服务器	103
6.2	中间件	110
6.2.1	Connect 基本知识	111
6.2.2	Connect 中间件	113
6.2.3	定制 Connect 中间件	118
6.3	Routers	121
6.4	Proxies	124

第 7 章	Express 框架	128
7.1	Express: 启动和运行	129
7.2	app.js 文件	130
7.3	错误处理	133
7.4	Express 与 Connect 的关系	134
7.5	路由	135
7.5.1	路由路径	137
7.5.2	路由和 HTTP 动词	140
7.6	关于 MVC	147
7.7	使用 cURL 测试 Express 应用程序	152
第 8 章	Express、模板系统和 CSS	154
8.1	EJS 模板系统 (Embedded JavaScript Template System)	154
8.1.1	基本语法	155
8.1.2	Node 与 EJS	156
8.1.3	EJS 与 Node Filters	158
8.2	在 Express 中使用 EJS	159
8.2.1	多对象环境的改造	161
8.2.2	静态文件路由	162
8.2.3	处理一个新对象的 Post 请求	164
8.2.4	Widget 索引和生成 picklist	166
8.2.5	显示单个对象并确认对象的删除操作	168
8.2.6	提供更新信息的表达以及处理 PUT 请求	170
8.3	Jade 模板系统	173
8.3.1	Jade 语法简介	173
8.3.2	使用 block 和 extends 模块化视图模板	176
8.3.3	Widget View 转换为 Jade 模板	178
8.3.4	转换 edit 和 delete 表单	179
8.4	使用 Stylus 完成简单的 CSS 样式	182
第 9 章	结构化数据、Noe 和 Redis	187
9.1	Node 和 Redis	188
9.2	构建游戏得分排行榜	190
9.3	创建消息队列	197
9.4	为 Express 应用程序添加统计中间件	201
第 10 章	Node 和 MongoDB: 文档中心数据	206
10.1	MongoDB Native Node.js Driver (MongoDB 原生 Node.js 驱动)	207

10.1.1	MongoDB 入门	207
10.1.2	定义、创建以及销毁 MongoDB Collection	208
10.1.3	为 Collection 添加数据	209
10.1.4	查询数据	212
10.1.5	使用 Updates、Upserts、Find 和 Remove	216
10.2	使用 Mongoose 实现 Widget 模块	221
10.3	重构 Widget 工厂	222
10.4	添加 MongoDB 后台	223
<b>第 11 章</b>	<b>Node 与关系型数据库</b>	<b>228</b>
11.1	db-mysql 入门	229
11.1.1	查询字符串和方法链	229
11.1.2	使用查询字符串更新数据库	233
11.1.3	使用方法链更新数据库	235
11.2	使用 node-mysql 实现本地 MySQL 访问	237
11.2.1	使用 node-mysql 做基本的 CRUD 操作	237
11.2.2	MySQL 事务与 mysql-queues	239
11.3	ORM 与 Sequelize	241
11.3.1	定义模型	241
11.3.2	ORM 风格的 CRUD 实现	243
11.3.3	添加多个对象	246
11.3.4	从关系型到 ORM	247
<b>第 12 章</b>	<b>图形和 HTML5 Video</b>	<b>248</b>
12.1	创建和使用 PDF	248
12.1.1	使用子进程访问 PDF 工具	249
12.1.2	使用 PDFKit 创建 PDF	257
12.2	从子进程访问 ImageMagick	258
12.3	通过 HTTP 提供 HTML5 Video 服务	263
12.4	创建和流化画布内容 (Canvas Content)	267
<b>第 13 章</b>	<b>WebSockets 和 Socket.IO</b>	<b>271</b>
13.1	WebSockets	271
13.2	Socket.IO 简介	272
13.2.1	一个简单的通信范例	273
13.2.2	异步世界里的 WebSockets	276
13.2.3	关于客户端代码	277
13.3	配置 Socket.IO	278

13.4	Chat: WebSockets 版本的 “Hello, World”	279
13.5	在 Express 中使用 Socket.IO	282
<b>第 14 章</b>	<b>Node 应用程序的测试和调试</b>	<b>284</b>
14.1	调试	284
14.1.1	Node.js Debugger	284
14.1.2	使用 Node Inspector 的客户端调试	287
14.2	单元测试 (Unit Testing)	289
14.2.1	Assert 与单元测试	289
14.2.2	Nodeunit 与单元测试	293
14.2.3	其他测试框架	295
14.3	验收测试	299
14.3.1	Soda 和 Selenium 测试	299
14.3.2	通过 Tobi 和 Zombie 模拟浏览器	303
14.4	性能测试: 基准问题和负载测试	304
14.4.1	ApacheBench 基准测试	305
14.4.2	Nodeload 与负载测试	309
14.5	Nodemon 更新代码	312
<b>第 15 章</b>	<b>安全及防护</b>	<b>313</b>
15.1	数据加密	314
15.1.1	TSL / SSL 配置	314
15.1.2	使用 HTTPS	315
15.1.3	如何安全的保存密码	317
15.2	认证/授权及 Passport	320
15.2.1	授权/认证策略: Oauth、OpenID、用户名/密码验证	321
15.2.2	Local Passport Strategy	323
15.2.3	Twitter Passport Strategy ( OAuth )	330
15.3	保护应用程序, 防止攻击	336
15.3.1	不要使用 eval	336
15.3.2	尽量使用复选框、单选按钮和下拉式选项	337
15.3.3	使用 node-validator	337
15.4	在沙箱中执行代码	339
<b>第 16 章</b>	<b>扩展和部署 Node 应用</b>	<b>343</b>
16.1	把你的节点部署到服务器上	343
16.1.1	编写 package.json 文件	344
16.1.2	使用 Forever 让你的应用 “永不掉线”	347



16.1.3	使用 Node 和 Apache .....	350
16.1.4	改善性能 .....	352
16.2	部署到云服务 .....	352
16.2.1	通过 Cloud9 IDE 部署到 Windows Azure .....	353
16.2.2	Joyent Development SmartMachine .....	355
16.2.3	Heroku .....	355
16.2.4	Amazon EC2 .....	356
16.2.5	Nodejitsu .....	356
附录	Node、Git 和 GitHub .....	357