

# *Node.js in Practice*

ALEX YOUNG  
MARC HARTER



MANNING  
SHELTER ISLAND

For online information and ordering of this and other Manning books, please visit [www.manning.com](http://www.manning.com). The publisher offers discounts on this book when ordered in quantity. For more information, please contact


Special Sales Department  
Manning Publications Co.  
20 Baldwin Road  
PO Box 761  
Shelter Island, NY 11964  
Email: [orders@manning.com](mailto:orders@manning.com)

©2015 by Manning Publications Co. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by means electronic, mechanical, photocopying, or otherwise, without prior written permission of the publisher.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in the book, and Manning Publications was aware of a trademark claim, the designations have been printed in initial caps or all caps.

© Recognizing the importance of preserving what has been written, it is Manning's policy to have the books we publish printed on acid-free paper, and we exert our best efforts to that end. Recognizing also our responsibility to conserve the resources of our planet, Manning books are printed on paper that is at least 15 percent recycled and processed without the use of elemental chlorine.

 Manning Publications Co.  
20 Baldwin Road  
PO Box 761  
Shelter Island, NY 11964

Development editor: Cynthia Kane  
Technical development editor: Jose Maria Alvarez Rodriguez  
Copyeditor: Benjamin Berg  
Proofreader: Katie Tennant  
Typesetter: Gordan Salinovic  
Cover designer: Marija Tudor

ISBN 9781617290930

Printed in the United States of America

1 2 3 4 5 6 7 8 9 10 – EBM – 19 18 17 16 15 14

# *brief contents*

---

## **PART 1** NODE FUNDAMENTALS .....1

- 1 ■ Getting started 3
- 2 ■ Globals: Node’s environment 15
- 3 ■ Buffers: Working with bits, bytes, and encodings 39
- 4 ■ Events: Mastering EventEmitter and beyond 64
- 5 ■ Streams: Node’s most powerful and misunderstood feature 82
- 6 ■ File system: Synchronous and asynchronous approaches to files 114
- 7 ■ Networking: Node’s true “Hello, World” 136
- 8 ■ Child processes: Integrating external applications with Node 174

## **PART 2** REAL-WORLD RECIPES ..... 197

- 9 ■ The Web: Build leaner and meaner web applications 199
- 10 ■ Tests: The key to confident code 260

- 11 ■ Debugging: Designing for introspection and resolving issues 293
- 12 ■ Node in production: Deploying applications safely 326

**PART 3 WRITING MODULES .....359**

- 13 ■ Writing modules: Mastering what Node is all about 361

# contents

---

*foreword* xiii  
*preface* xv  
*acknowledgments* xvi  
*about this book* xviii  
*about the cover illustration* xx

## PART 1 NODE FUNDAMENTALS.....1

### 1 **Getting started** 3

#### 1.1 Getting to know Node 4

*Why Node?* 4 ▪ *Node's main features* 6

#### 1.2 Building a Node application 8

*Creating a new Node project* 9 ▪ *Making a stream class* 9

*Using a stream* 10 ▪ *Writing a test* 12

#### 1.3 Summary 13

### 2 **Globals: Node's environment** 15

#### 2.1 Modules 16

TECHNIQUE 1 Installing and loading modules 16

TECHNIQUE 2 Creating and managing modules 17

	TECHNIQUE 3	Loading a group of related modules	19
	TECHNIQUE 4	Working with paths	21
2.2		Standard I/O and the console object	22
	TECHNIQUE 5	Reading and writing to standard I/O	22
	TECHNIQUE 6	Logging messages	24
	TECHNIQUE 7	Benchmarking a program	25
2.3		Operating system and command-line integration	27
	TECHNIQUE 8	Getting platform information	27
	TECHNIQUE 9	Passing command-line arguments	28
	TECHNIQUE 10	Exiting a program	29
	TECHNIQUE 11	Responding to signals	31
2.4		Delaying execution with timers	32
	TECHNIQUE 12	Executing functions after a delay with setTimeout	32
	TECHNIQUE 13	Running callbacks periodically with timers	34
	TECHNIQUE 14	Safely managing asynchronous APIs	35
2.5		Summary	38

### **3 *Buffers: Working with bits, bytes, and encodings*** 39

3.1		Changing data encodings	40
	TECHNIQUE 15	Converting buffers into other formats	40
	TECHNIQUE 16	Changing string encodings using buffers	41
3.2		Converting binary files to JSON	44
	TECHNIQUE 17	Using buffers to convert raw data	44
3.3		Creating your own binary protocol	58
	TECHNIQUE 18	Creating your own network protocol	58
3.4		Summary	63

### **4 *Events: Mastering EventEmitter and beyond*** 64

4.1		Basic usage	65
	TECHNIQUE 19	Inheriting from EventEmitter	65
	TECHNIQUE 20	Mixing in EventEmitter	68
4.2		Error handling	69
	TECHNIQUE 21	Managing errors	69
	TECHNIQUE 22	Managing errors with domains	71
4.3		Advanced patterns	73
	TECHNIQUE 23	Reflection	73
	TECHNIQUE 24	Detecting and exploiting EventEmitter	75
	TECHNIQUE 25	Categorizing event names	77

- 4.4 Third-party modules and extensions 78
  - TECHNIQUE 26 Alternatives to EventEmitter 78
- 4.5 Summary 80

## 5 *Streams: Node's most powerful and misunderstood feature* 82

- 5.1 Introduction to streams 83
  - Types of streams* 83 ▪ *When to use streams* 84 ▪ *History* 85
  - Streams in third-party modules* 85 ▪ *Streams inherit from EventEmitter* 87
- 5.2 Built-in streams 88
  - TECHNIQUE 27 Using built-in streams to make a static web server 88
  - TECHNIQUE 28 Stream error handling 90
- 5.3 Third-party modules and streams 91
  - TECHNIQUE 29 Using streams from third-party modules 91
- 5.4 Using the stream base classes 94
  - TECHNIQUE 30 Correctly inheriting from the stream base classes 94
  - TECHNIQUE 31 Implementing a readable stream 96
  - TECHNIQUE 32 Implementing a writable stream 99
  - TECHNIQUE 33 Transmitting and receiving data with duplex streams 101
  - TECHNIQUE 34 Parsing data with transform streams 103
- 5.5 Advanced patterns and optimization 105
  - TECHNIQUE 35 Optimizing streams 105
  - TECHNIQUE 36 Using the old streams API 108
  - TECHNIQUE 37 Adapting streams based on their destination 109
  - TECHNIQUE 38 Testing streams 111
- 5.6 Summary 113

## 6 *File system: Synchronous and asynchronous approaches to files* 114

- 6.1 An overview of the fs module 115
  - POSIX file I/O wrappers* 115 ▪ *Streaming* 117 ▪ *Bulk file I/O* 117
  - File watching* 118 ▪ *Synchronous alternatives* 118
- TECHNIQUE 39 Loading configuration files 119
- TECHNIQUE 40 Using file descriptors 120
- TECHNIQUE 41 Working with file locking 121
- TECHNIQUE 42 Recursive file operations 125

- TECHNIQUE 43 Writing a file database 128
- TECHNIQUE 44 Watching files and directories 132
- 6.2 Summary 134

## 7 *Networking: Node's true "Hello, World" 136*

- 7.1 Networking in Node 137
  - Networking terminology 137* ▪ *Node's networking modules 141*
  - Non-blocking networking and thread pools 142*
- 7.2 TCP clients and servers 143
  - TECHNIQUE 45 Creating a TCP server and tracking clients 143
  - TECHNIQUE 46 Testing TCP servers with clients 145
  - TECHNIQUE 47 Improve low-latency applications 147
- 7.3 UDP clients and servers 149
  - TECHNIQUE 48 Transferring a file with UDP 149
  - TECHNIQUE 49 UDP client server applications 153
- 7.4 HTTP clients and servers 156
  - TECHNIQUE 50 HTTP servers 156
  - TECHNIQUE 51 Following redirects 158
  - TECHNIQUE 52 HTTP proxies 162
- 7.5 Making DNS requests 165
  - TECHNIQUE 53 Making a DNS request 165
- 7.6 Encryption 167
  - TECHNIQUE 54 A TCP server that uses encryption 167
  - TECHNIQUE 55 Encrypted web servers and clients 170
- 7.7 Summary 173

## 8 *Child processes: Integrating external applications with Node 174*

- 8.1 Executing external applications 175
  - TECHNIQUE 56 Executing external applications 176
    - Paths and the PATH environment variable 176* ▪ *Errors when executing external applications 177*
  - TECHNIQUE 57 Streaming and external applications 178
    - Stringing external applications together 179*
  - TECHNIQUE 58 Executing commands in a shell 180
    - Security and shell command execution 181*



- TECHNIQUE 59 Detaching a child process 182  
*Handing I/O between the child and parent processes* 183 ▪ *Reference counting and child processes* 184
- 8.2 Executing Node programs 185
  - TECHNIQUE 60 Executing Node programs 185
  - TECHNIQUE 61 Forking Node modules 186
  - TECHNIQUE 62 Running jobs 188  
*Job pooling* 190 ▪ *Using the pooler module* 191
- 8.3 Working synchronously 192
  - TECHNIQUE 63 Synchronous child processes 192
- 8.4 Summary 194

## PART 2 REAL-WORLD RECIPES.....197

### 9 *The Web: Build leaner and meaner web applications* 199

- 9.1 Front-end techniques 200
  - TECHNIQUE 64 Quick servers for static sites 200
  - TECHNIQUE 65 Using the DOM in Node 204
  - TECHNIQUE 66 Using Node modules in the browser 207
- 9.2 Server-side techniques 209
  - TECHNIQUE 67 Express route separation 209
  - TECHNIQUE 68 Automatically restarting the server 212
  - TECHNIQUE 69 Configuring web applications 215
  - TECHNIQUE 70 Elegant error handling 219
  - TECHNIQUE 71 RESTful web applications 222
  - TECHNIQUE 72 Using custom middleware 231
  - TECHNIQUE 73 Using events to decouple functionality 236
  - TECHNIQUE 74 Using sessions with WebSockets 238
  - TECHNIQUE 75 Migrating Express 3 applications to Express 4 242
- 9.3 Testing web applications 246
  - TECHNIQUE 76 Testing authenticated routes 246
  - TECHNIQUE 77 Creating seams for middleware injection 248
  - TECHNIQUE 78 Testing applications that depend on remote services 250
- 9.4 Full stack frameworks 256
- 9.5 Real-time services 257
- 9.6 Summary 258

<b>10</b>	<b><i>Tests: The key to confident code</i></b>	<b>260</b>
10.1	Introduction to testing with Node	261
10.2	Writing simple tests with assertions	262
	TECHNIQUE 79 Writing tests with built-in modules	263
	TECHNIQUE 80 Testing for errors	265
	TECHNIQUE 81 Creating custom assertions	268
10.3	Test harnesses	270
	TECHNIQUE 82 Organizing tests with a test harness	270
10.4	Test frameworks	273
	TECHNIQUE 83 Writing tests with Mocha	273
	TECHNIQUE 84 Testing web applications with Mocha	276
	TECHNIQUE 85 The Test Anything Protocol	280
10.5	Tools for tests	282
	TECHNIQUE 86 Continuous integration	283
	TECHNIQUE 87 Database fixtures	285
10.6	Further reading	291
10.7	Summary	292
<b>11</b>	<b><i>Debugging: Designing for introspection and resolving issues</i></b>	<b>293</b>
11.1	Designing for introspection	294
	<i>Explicit exceptions</i> 294 ▪ <i>Implicit exceptions</i> 295 ▪ <i>The error event</i> 295 ▪ <i>The error argument</i> 296	
	TECHNIQUE 88 Handling uncaught exceptions	296
	TECHNIQUE 89 Linting Node applications	299
11.2	Debugging issues	300
	TECHNIQUE 90 Using Node's built-in debugger	300
	TECHNIQUE 91 Using Node Inspector	306
	TECHNIQUE 92 Profiling Node applications	308
	TECHNIQUE 93 Debugging memory leaks	311
	TECHNIQUE 94 Inspecting a running program with a REPL	316
	TECHNIQUE 95 Tracing system calls	322
11.3	Summary	325
<b>12</b>	<b><i>Node in production: Deploying applications safely</i></b>	<b>326</b>
12.1	Deployment	327
	TECHNIQUE 96 Deploying Node applications to the cloud	327
	TECHNIQUE 97 Using Node with Apache and nginx	332

	TECHNIQUE 98	Safely running Node on port 80	335
	TECHNIQUE 99	Keeping Node processes running	336
	TECHNIQUE 100	Using WebSockets in production	338
12.2		Caching and scaling	342
	TECHNIQUE 101	HTTP caching	342
	TECHNIQUE 102	Using a Node proxy for routing and scaling	344
	TECHNIQUE 103	Scaling and resiliency with cluster	347
12.3		Maintenance	351
	TECHNIQUE 104	Package optimization	351
	TECHNIQUE 105	Logging and logging services	353
12.4		Further notes on scaling and resiliency	356
12.5		Summary	357

## PART 3 WRITING MODULES.....359

### **13** *Writing modules: Mastering what Node is all about* 361

13.1		Brainstorming	363
		<i>A faster Fibonacci module</i>	363
	TECHNIQUE 106	Planning for our module	363
	TECHNIQUE 107	Proving our module idea	366
13.2		Building out the package.json file	370
	TECHNIQUE 108	Setting up a package.json file	370
	TECHNIQUE 109	Working with dependencies	373
	TECHNIQUE 110	Semantic versioning	377
13.3		The end user experience	379
	TECHNIQUE 111	Adding executable scripts	379
	TECHNIQUE 112	Trying out a module	381
	TECHNIQUE 113	Testing across multiple Node versions	383
13.4		Publishing	385
	TECHNIQUE 114	Publishing modules	385
	TECHNIQUE 115	Keeping modules private	387
13.5		Summary	388
	<i>appendix</i>	<i>Community</i>	391
		<i>index</i>	395



# foreword

---

You have in your hands a book that will take you on an in-depth tour of Node.js. In the pages to come, Alex Young and Marc Harter will help you grasp Node's core in a deep way: from modules to real, networked applications.

Networked applications are, of course, an area where Node.js shines. You, dear reader, are likely well aware of that; I daresay it is your main reason for purchasing this tome! For the few of you who actually read the foreword, let me tell you the story of how it all began.

In the beginning, there was the C10K problem. And the C10K problem raised this question: if you want to handle 10,000 concurrent network connections on contemporary hardware, how do you go about that?

You see, for the longest time operating systems were terrible at dealing with large numbers of network connections. The hardware was terrible in many ways, the software was terrible in other ways, and when it came to the interaction between hardware and software ... linguists had a field day coming up with proper neologisms; plain *terrible* doesn't do it justice. Fortunately, technology is a story of progress; hardware gets better, software saner. Operating systems improved at managing large numbers of network connections, as did user software.

We conquered the C10K problem a long time ago, moved the goal posts, and now we've set our sights on the C100K, C500K, and C1M problems. Once we've comfortably crossed those frontiers, I fully expect that the C10M problem will be next.

Node.js is part of this story of ever-increasing concurrency, and its future is bright: we live in an increasingly connected world and that world needs a power tool to connect everything. I believe Node.js is that power tool, and I hope that, after reading this book, you will feel the same way.

BEN NOORDHUIS  
COFOUNDER, STRONGLOOP, INC.

# *preface*

---

When Node.js arrived in 2009, we knew something was different. JavaScript on the server wasn't anything new. In fact, server-side JavaScript has existed almost as long as client-side JavaScript. With Node, the speed of the JavaScript runtimes, coupled with the event-based parallelism that many JavaScript programmers were already familiar with, were indeed compelling. And not just for client-side JavaScript developers, which was our background—Node attracted developers from the systems level to various server-side backgrounds, PHP to Ruby to Java. We all found ourselves inside this movement.

At that time, Node was changing a lot, but we stuck with it and learned a whole lot in the process. From the start, Node focused on making a small, low-level core library that would provide enough functionality for a large, diverse user space to grow. Thankfully, this large and diverse user space exists today because of these design decisions early on. Node is a lot more stable now and used in production for numerous startups as well as established enterprises.

When Manning approached us about writing an intermediate-level book on Node, we looked at the lessons we had learned as well as common pitfalls and struggles we saw in the Node community. Although we loved the huge number of truly excellent third-party modules available to developers, we noticed many developers were getting less and less education on the core foundations of Node. So we set out to write *Node in Practice* to journey into the roots and foundations of Node in a deep and thorough manner, as well as tackle many issues we personally have faced and have seen others wrestle with.

# *acknowledgments*

---

We have many people to thank, without whose help and support this book would not have been possible.

Thanks to the Manning Early Access Program (MEAP) readers who posted comments and corrections in the Author Online forum.

Thanks to the technical reviewers who provided invaluable feedback on the manuscript at various stages of its development: 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, and Todd Williams.

Thanks to the entire Manning team for helping us every step of the way, especially our development editor Cynthia Kane, our copyeditor Benjamin Berg, our proofreader Katie Tennant, and everyone else who worked behind the scenes.

Special thanks to Ben Noordhuis for writing the foreword to our book, and to Valentin Crettaz and Michael Levin for their careful technical proofread of the book shortly before it went into production.

## **Alex Young**

I couldn't have written this book without the encouragement and support of the DailyJS community. Thanks to everyone who has shared modules and libraries with me over the last few years: keeping up to date with the Node.js community would have been impossible without you. Thank you also to my colleagues at Papers who have allowed me to



use my Node.js skills in production. Finally, thanks to Yuka for making me believe I can do crazy things like start companies and write books.

***Marc Harter***

I would like thank Ben Noordhuis, Isaac Schlueter, and Timothy Fontaine for all the IRC talks over Node; you know the underlying systems that support Node in such a deep way that learning from you makes Node even richer. Also, I want to thank my coauthor Alex; it seems rare to have such a similar approach to writing a book as I did with Alex, plus it was fun for a Midwestern US guy to talk shop with an English chap. Ultimately my heart goes out to my wife, who really made this whole thing possible, if I'm honest. Hannah, you are loved; thank you.

## *about this book*

---

*Node.js in Practice* exists to provide readers a deeper understanding of Node’s core modules and packaging system. We believe this is foundational to being a productive and confident Node developer. Unfortunately, this small core is easily missed for the huge and vibrant third-party ecosystem with modules prebuilt for almost any task. In this book we go beyond regurgitating the official Node documentation in order to get practical and thorough. We want the reader to be able to dissect the inner workings of the third-party modules they include as well as the projects they write.

This book is not an entry-level Node book. For that, we recommend reading Manning’s *Node.js In Action*. This book is targeted at readers who already have experience working with Node and are looking to take it up a notch. Intermediate knowledge of JavaScript is recommended. Familiarity with the Windows, OS X, or Linux command line is also recommended.

In addition, we’re aware that many Node developers have come from a client-side JavaScript background. For that reason, we spend some time explaining less-familiar concepts such as working with binary data, how underlying networking and file systems work, and interacting with the host operating system—all using Node as a teaching guide.

### **Chapter roadmap**

This book is organized into three parts.

Part 1 covers Node’s core fundamentals, where we focus our attention on what’s possible using only Node’s core modules (no third-party modules). Chapter 1 recaps

Node.js's purpose and function. Then chapters 2 through 8 each cover in depth a different core aspect of Node from buffers to streams, networking to child processes.

Part 2 focuses on real-world development recipes. Chapters 9 through 12 will help you master four highly applicable skills—testing, web development, debugging, and running Node in production. In addition to Node core modules, these sections include the use of various third-party modules.

Part 3 guides you through creating your own Node modules in a straightforward manner that ties in all kinds of ways to use npm commands for packaging, running, testing, benchmarking, and sharing modules. It also includes helpful tips on versioning projects effectively.

There are 115 techniques in the book, each module covering a specific Node.js topic or task, and each divided into practical Problem/Solution/Discussion sections.

### **Code conventions and downloads**

All source code in the book is in a fixed-width font like this, which sets it off from the surrounding text. In many listings, the code is annotated to point out the key concepts, and numbered bullets are sometimes used in the text to provide additional information about the code.

This book's coding style is based on the Google JavaScript Style Guide.<sup>1</sup> That means we've put var statements on their own lines, used camelCase to format function and variable names, and we always use semicolons. Our style is a composite of the various JavaScript styles used in the Node community.

Most of the code shown in the book can be found in various forms in the sample source code that accompanies it. The sample code can be downloaded free of charge from the Manning website at [www.manning.com/Node.jsinPractice](http://www.manning.com/Node.jsinPractice), as well as from GitHub at the following link: <https://github.com/alexyoung/nodeinpractice>.

### **Author Online forum**

Purchase of *Node.js in Practice* includes free access to a private web forum run by Manning Publications where you can make comments about the book, ask technical questions, and receive help from the authors and from other users. To access the forum and subscribe to it, point your web browser to [www.manning.com/Node.jsinPractice](http://www.manning.com/Node.jsinPractice). This page provides information on how to get on the forum once you're registered, what kind of help is available, and the rules of conduct on the forum.

The Author Online forum and the archives of previous discussions will be accessible from the publisher's website as long as the book is in print.

You can also contact the authors at the following Google Group URL: <https://groups.google.com/forum/#!forum/nodejsinpractice>.

---

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

## *about the cover illustration*

---

The caption for the illustration on the cover of *Node.js in Practice* is “Young Man from Ayvalik,” a town in Turkey on the Aegean Coast. The illustration is taken from a collection of costumes of the Ottoman Empire published on January 1, 1802, by William Miller of Old Bond Street, London. The title page is missing from the collection and we have been unable to track it down to date. The book’s table of contents identifies the figures in both English and French, and each illustration bears the names of two artists who worked on it, both of whom would no doubt be surprised to find their art gracing the front cover of a computer programming book ... two hundred years later.

The collection was purchased by a Manning editor at an antiquarian flea market in the “Garage” on West 26th Street in Manhattan. The seller was an American based in Ankara, Turkey, and the transaction took place just as he was packing up his stand for the day. The Manning editor didn’t have on his person the substantial amount of cash that was required for the purchase, and a credit card and check were both politely turned down. With the seller flying back to Ankara that evening, the situation was getting hopeless. What was the solution? It turned out to be nothing more than an old-fashioned verbal agreement sealed with a handshake. The seller simply proposed that the money be transferred to him by wire, and the editor walked out with the bank information on a piece of paper and the portfolio of images under his arm. Needless to say, we transferred the funds the next day, and we remain grateful and impressed by this unknown person’s trust in one of us. It recalls something that might have happened a long time ago.

We at Manning celebrate the inventiveness, the initiative, and, yes, the fun of the computer business with book covers based on the rich diversity of regional life of two centuries ago, brought back to life by the pictures from this collection.