

Practical Node.js

Building Real-World Scalable Web Apps

Azat Mardan

Apress®

Contents

About the Author	xv
About the Technical Reviewer	xvii
Acknowledgments	xix
Introduction	xxi
■Chapter 1: Setting up Node.js and Other Essentials	1
Installing Node.js and NPM	1
One-Click Installers	2
Installing with HomeBrew or MacPorts	3
Installing from a Tar File	4
Installing Without sudo	4
Installing from a Git Repo	5
Multiversion Setup with Nave	5
Multiversion Setup with NVM	6
Alternative Multiversion Systems	6
Checking the Installation	7
Node.js Console (REPL)	7
Launching Node.js Scripts	8
Node.js Basics and Syntax	9
Loose Typing	9
Buffer—Node.js Super Data Type	10
Object Literal Notation	10
Functions	10
Arrays	12

Prototypal Nature.....	12
Conventions.....	13
Node.js Globals and Reserved Keywords	14
__dirname vs. process.cwd	16
Browser Application Programming Interface Helpers	17
Node.js Core Modules.....	18
Handy Node.js Utilities.....	19
Reading to and Writing from the File System in Node.js	20
Streaming Data in Node.js	20
Installing Node.js Modules with NPM	20
Taming Callbacks in Node.js.....	21
Hello World Server with HTTP Node.js Module	21
Debugging Node.js Programs.....	22
Core Node.js Debugger	22
Debugging with Node Inspector	23
Node.js IDEs and Code Editors	28
Watching for File Changes	30
Summary.....	31
■ Chapter 2: Using Express.js 4 to Create Node.js Web Apps.....	33
What Is Express.js?	33
How Express.js Works	36
Express.js Installation	36
Express.js Version.....	36
Express.js Generator.....	37
Local Express.js.....	37
Express.js Scaffolding.....	40
Express.js Command-Line Interface.....	41
Routes in Express.js	43
Middleware as the Backbone of Express.js.....	43
Configuration of an Express.js App.....	44

Jade Is Haml for Express.js/Node.js	44
Conclusion About Scaffolding	45
The Blog Project Overview	45
Submitting the Data	46
Express.js 4 Hello World Example	48
Setting up Folders	49
NPM Init and package.json	49
Dependency Declaration: npm install	50
The App.js File	51
Meet Jade: One Template to Rule Them All	55
Running the Hello World App	55
Summary	56
■Chapter 3: TDD and BDD for Node.js with Mocha	57
Installing and Understanding Mocha	57
Understanding Mocha Hooks	59
TDD with the Assert	60
Chai Assert	62
BDD with Expect.js	63
Expect.js Syntax	64
Project: Writing the First BDD Test for Blog	65
Putting Configs into a Makefile	67
Summary	69
■Chapter 4: Template Engines: Jade and Handlebars	71
Jade Syntax and Features	71
Tags	71
Variables/Locals	72
Attributes	73
Literals	74
Text	74
Script and Style Blocks	74

JavaScript Code.....	75
Comments	75
Conditions (if)	76
Iterations (each loops)	76
Filters.....	77
Interpolation	77
Case.....	77
Mixins	77
Include.....	78
Extend.....	79
Standalone Jade Usage.....	79
Handlebars Syntax	83
Variables.....	83
Iteration (each)	84
Unescaped Output	84
Conditions (if)	85
Unless.....	85
With	86
Comments	87
Custom Helpers	87
Includes (Partials).....	88
Standalone Handlebars Usage	88
Jade and Handlebars Usage in Express.js 4	91
Jade and Express.js.....	91
Handlebars and Express.js	92
Project: Adding Jade Templates to Blog	93
layout.jade	93
index.jade	95
article.jade.....	97
login.jade.....	98

post.jade.....	99
admin.jade.....	101
Summary.....	102
■Chapter 5: Persistence with MongoDB and Mongoskin.....	103
Easy and Proper Installation of MongoDB	103
How to Run the Mongo Server	105
Data Manipulation from the Mongo Console	106
MongoDB Shell in Detail.....	107
Minimalistic Native MongoDB Driver for Node.js Example.....	108
Main Mongoskin Methods	111
Project: Storing Blog Data in MongoDB with Mongoskin	113
Project: Adding MongoDB Seed Data.....	113
Project: Writing Mocha Tests	114
Project: Adding Persistence.....	115
Running the App	126
Summary.....	127
■Chapter 6: Using Sessions and OAuth to Authorize and Authenticate Users in Node.js Apps.....	129
Authorization with Express.js Middleware	129
Token-Based Authentication.....	130
Session-Based Authentication	131
Project: Adding E-mail and Password Login to Blog	132
Session Middleware	132
Authorization in Blog	133
Authentication in Blog	136
Running the App	137
Node.js OAuth	138
Twitter OAuth 2.0 Example with Node.js OAuth	138
Everyauth.....	139

Project: Adding Twitter OAuth 1.0 Sign-in to Blog with Everyauth	140
Adding a Sign-in with a Twitter Link.....	141
Configuring the Everyauth Twitter Strategy.....	141
Summary.....	147
■Chapter 7: Boosting Your Node.js Data with the Mongoose ORM Library	149
Mongoose Installation	150
Connection Establishment in a Standalone Mongoose Script.....	150
Mongoose Schemas.....	152
Hooks for Keeping Code Organized.....	154
Custom Static and Instance Methods.....	154
Mongoose Models	155
Relationships and Joins with Population	157
Nested Documents.....	158
Virtual Fields	159
Schema Type Behavior Amendment.....	160
Express.js + Mongoose = True MVC.....	162
Summary.....	172
■Chapter 8: Building Node.js REST API Servers with Express.js and Hapi.....	173
RESTful API Basics	174
Project Dependencies	175
Test Coverage with Mocha and Superagent.....	176
REST API Server Implementation with Express and Mongoskin	181
Refactoring: Hapi RESP API Server.....	187
Summary.....	194
■Chapter 9: Real-Time Apps with WebSocket, Socket.IO, and DerbyJS	195
What Is WebSocket?.....	195
Native WebSocket and Node.js with the ws Module Example.....	196
Browser WebSocket Implementation.....	196
Node.js Server with ws Module Implementation.....	197

Socket.IO and Express.js Example	199
Collaborative Online Code Editor Example with DerbyJS, Express.js, and MongoDB	204
Project Dependencies and package.json	205
Server-side Code	206
DerbyJS App	208
DerbyJS View	210
Editor Tryout	212
Summary	213
■ Chapter 10: Getting Node.js Apps Production Ready	215
Environment Variables	215
Express.js in Production	216
Socket.IO in Production	218
Error Handling	219
Node.js Domains for Error Handling	221
Multithreading with Cluster	224
Multithreading with Cluster2	226
Event Logging and Monitoring	227
Monitoring	227
REPL in Production	229
Winston	230
Papertrail App for Logging	230
Building Tasks with Grunt	231
Git for Version Control and Deployments	235
Installing Git	235
Generating SSH Keys	236
Creating a Local Git Repository	239
Pushing the Local Repository to GitHub	239
Running Tests in Cloud with TravisCI	240
TravisCI Configuration	241
Summary	241

■Chapter 11: Deploying Node.js Apps	243
Deploying to Heroku	243
Deploying to Amazon Web Services	248
Keeping Node.js Apps Alive with forever, Upstart, and init.d	252
forever	252
Upstart Scripts	253
init.d	255
Serving Static Resources Properly with Nginx	257
Caching with Varnish	259
Summary	260
■Chapter 12: Publishing Node.js Modules and Contributing to Open Source.....	261
Recommended Folder Structure	262
Required Patterns.....	262
package.json	265
Publishing to NPM	265
Locking Versions	266
Summary	266
Practical Node.js Conclusion	267
Further Reading.....	267
Errata and Contacts.....	267
Index.....	269