

Index

■ A

- addListener method, 219
- Amazon Web Services (AWS)
 - advantages, 248
 - Domain Name System, 251
 - EC2 instance, 250
 - EPEL, 248
 - HTTP traffic, 251
 - Node.js build, 250
 - NPM, 249–250
 - server.js, 250
 - yum command, 249
- Application programming interface (API), 17
 - array objects, 17
 - Math objects, 17
 - string objects, 17
- Authentication, Blog, 132
 - destroy(), 137
 - findOne function, 136
 - return keyword, 136
- Authorization, 129, 133

■ B

- Behavior-driven development (BDD), 57
- Blog, 132
 - authentication
 - destroy(), 137
 - findOne function, 136
 - return keyword, 136
 - authorization, 133
 - running the app, 137
 - session middleware, 132

■ C

- Content delivery networks (CDNs), 211

■ D

- DELETE method, 175, 183
- DerbyJS, 195
 - app
 - app.js code, 209
 - declaration, 208
 - DOM, 208
 - model.ref(), 209
 - process.nextTick function, 209
 - snippetId route, 208
 - editor tryout, 212
 - package.json, 205
 - server-side code, 206
 - structure, 205
 - usage, 204
 - views, 210
- Document Object Model (DOM), 208
- Domain Name System (DNS), 251

■ E

- Editor tryout, 212
- Environment variables, 215
- Error handling
 - addListener method, 219
 - Node.js domains, 221
 - onerror event listeners, 219
 - REST API, 220
- Event logging, 230
- Everyauth, 139

Express.js, 33

- Agile JavaScript Development, 34
 - app.configure() method, 216
 - Blog app, 45
 - admin page, 48
 - \$.ajax() method, 47
 - elements, 46
 - REST API approach, 47
 - traditional server-side approach, 46
 - hello world creation, 48
 - app.js file, 51, 54–55
 - app.set(), 52
 - folder setup, 49
 - framework, 52
 - Jade syntax, 55
 - \$ node app command, 56
 - node_modules folder, 51
 - NPM Init and package.json, 50
 - \$ npm install, 50
 - require() method, 51
 - res.render(), 54
 - VERB values, 53
 - HTTP methods and URLs, 35
 - if/else statements, 216
 - installation
 - Express.js Generator, 37
 - Express.js Version, 36
 - node_modules folder, 39
 - \$ npm init terminal command, 38
 - \$ npm ls command, 39–40
 - package.json file, 38–39
 - middleware, 33, 36, 217
 - mongoose, 162
 - MVC-like, 35
 - Node.js framework, 35
 - scaffolding
 - app.js middleware, 43
 - command-line interface, 40–41
 - creation, 40
 - Express.js App, configuration of, 44
 - Jade template engine, 44
 - routes, 43
 - terminal command, 40
 - session options, 217
 - tasks, 33
 - working principles, 36
- Express.js usage, 4, 91
- __express method, 91
 - Handlebars, 92
 - Jade, 91
 - res.render method, 91
- Express.js middleware, 129
- Extra Packages for Enterprise Linux (EPEL), 248

■ F

- Forever, 252

■ G

- GET method, 175, 186
- Git, 235
 - GitHub, 239
 - installation, 236
 - local repository creation, 239
 - PaaS, 235
 - SSH Keys, 236
- Grunt
 - default task, 235
 - grunt.loadNpmTasks() method, 231
 - HTTP requests, 232
 - jshint, 232–234
 - NPM installation, 231
 - uglify method, 232

■ H

- Handlebars
 - comments, 87
 - custom helpers, 87
 - each iteration, 84
 - if condition, 85
 - includes/partials templates, 88
 - unescaped output, 84
 - unless statement, 85
 - usages, 89
 - With statement, 86
- Handlebars expressions, 83
- Handlebars syntax, 83
- Hapi framework
 - database collection, 188
 - findOne method, 190
 - goals, 188
 - hapi-app.js coding, 191
 - hapi-app.js file creation, 188
 - loadCollection method, 189
 - server.route(), 189
 - server.start() method, 191
 - URL parameter, 190
- Heroku
 - configuration, 246
 - creation, 244
 - env file, 246–247
 - environment variables, 246
 - Git, 245, 248
 - \$ heroku login command, 244
 - MongoHQ URI, 248

Paas, 243
 transactional e-mails, 247
 working principle, 243

■ **I**

init.d script, 255

■ **J, K**

Jade, 71
 add templates, 93
 admin.jade, 101
 article.jade, 97
 index.jade, 95
 layout.jade, 93
 login.jade, 98
 post.jade, 99
 attributes, 73
 block blockname, 79
 case statement, 77
 comments, 75
 extend filename, 79
 filters, 77
 if statement, 76
 include, 78
 interpolation, 77
 iterations, 76
 JavaScript code, 75
 literals, 74
 mixins functions, 77
 script/style tags, 74
 tags, 71
 text, 74
 usages, 79
 variables/locals, 72
 JavaScript fundamentals. *See* Node.js
 JSON, 174

■ **L**

Locking versions, 266

■ **M**

map() function, 178
 Mocha, 57
 BDD test, 65
 boot method, 66
 \$ make test command, 68
 \$ mkdir tests, 65
 \$ mocha tests, 66
 package.json file, 65
 shutdown method, 66
 test folder, 67

behavior-driven development, 57
 Except.js, BDD, 63
 chai library, 64
 library, 64
 hooks, 59
 installation, 57
 BDD, 58
 node_modules, 58
 nyan reporter, 59
 optional parameters, 58
 TDD, 58
 TDD assert, 57, 60
 array-type test, 61
 chai assert, 62

Mocha hooks, 59
 Mocha tool, 57
 MongoDB, 103, 113
 bin folder, 105
 installation, 103
 data directory, 104
 HomeBrew, 103
 mongod service, 106
 Node.js native driver, 108
 library documentation, 111
 mongo-native-insert.js file, 109
 package.json file, 108
 running sequence, 108
 NoSQL database/non-relational databases, 103
 shell commands, 107
 storing blog, mongoskin, 113
 add persistence (*see* Persistence)
 add seed data, 113
 admin page, 127
 \$ make db /\$ make test, 126
 mocha tests, 114
 \$ mongod service, 126
 \$ node app, 126
 Mongoose, 149
 custom instance method, 154
 hooks, 154
 installation, 150
 instance methods, 156
 mongoose.model(name, schema), 155
 nested documents, 158
 population
 populate method, 158
 posts and user models, 157
 Schema behavior, 161
 standalone Mongoose script, 150
 static methods, 154
 static model methods, 155
 True MVC, 162
 virtual fields, 159
 Mongoskin methods, 111
 data validation, 113

- Mongoose methods (*cont.*)
 - model-view-controller-like (MVC-like), 112
 - native Node.js, 112
- Monitoring, 227
- Multithreading
 - cluster, 224
 - cluster2, 226–227

■ N

- Native WebSocket, 196
- Nginx, 257
 - HTTP header, 257
 - installation, CentOS system, 257
 - proxy-server configurations, 258
 - URL path, 258
 - VIM editor, 257
- Node.js, 1, 9
 - application programming interface, 17
 - array objects, 17
 - Math objects, 17
 - string objects, 17
 - arrays, 12
 - Buffer—Node.js, 10
 - callbacks, 21
 - code editor, 28
 - core modules, 18
 - fs module, 19
 - http module, 18
 - querystring module, 18
 - url module, 19
 - util module, 18
 - debugging, 22
 - alert() statements, 22
 - console.log(), 22
 - debug commands, 23
 - Node inspector, 23
 - __dirname *vs.* process.cwd, 16
 - file changes, 30
 - functions, 10
 - creation, 10
 - definition, 11
 - invocation *vs.* expression, 12
 - parameters, 11
 - global space, 14
 - global scope, 15
 - modules, 15
 - process.pid, 14
 - Hello World server, 21
 - installation, 1
 - cloned Git repo, 5
 - HomeBrew/Macports, 3
 - Nave setup, 5
 - Node Version Manager (NVM), 6
 - one-click installer, 2
 - read–eval–print–loop (REPL), 7

- Tar File, 4
 - without sudo, 4
 - language conventions, 13
 - camelcase, 13
 - commas, 13
 - indentation, 14
 - naming, 13
 - semicolons, 13
 - whitespaces, 14
 - loose typing, 9
 - NPM installation, 20
 - object notation, 10
 - prototypal inheritance, 12
 - read and write, 20
 - streaming data, 20
 - utilities, 19
 - web development, IDEs and
 - text editor, 29
 - webstorm IDE, 29–30
- Node.js apps
 - forever, 252
 - init.d script, 255
 - upstart scripts, 253
- Node.js OAuth, 138
 - Everyauth, 139
 - Twitter, 138
- NPM commands, 265

■ O

- OAuth 1.0, 140
 - Everyauth Twitter strategy
 - configuration, 141
 - with debug mode, 147
 - run the app, 146
 - sign-in with Twitter link, 141
- Object relational mapping (ORM), 103, 149
- Open-source factors, 261
 - locking versions, 266
 - NPM commands, 265–266
 - package.json file, 265
 - patterns
 - class, 263
 - exports.NAME pattern, 264
 - function, 263
 - object, 264
 - recommended folder structure, 262

■ P, Q

- Papertrail App, 230
- Patterns
 - class, 263
 - exports.NAME pattern, 264
 - function, 263
 - object, 264

Persistence, 115

- Express.js 4 modules, 115
- find method, 119
- GET admin page route, 120
- mongoskin, 115
- post method, 119
- remove and `_id`, 119
- req objects, 116
- updateById shorthand method, 119

Platform as a Service (PaaS), 235, 243

- POST method, 175, 187
- prototype keywords, 263
- PUT method, 175

R

Read-eval-print-loop (REPL), 7, 229

Recommended folder structure, 262

REST API servers, 173

- approach, 173
- characteristics of, 174
- CRUD structure, 174
- Hapi framework
 - database collection, 188
 - findOne method, 190
 - goals, 188
 - hapi-app.js coding, 191
 - hapi-app.js file creation, 188
 - loadCollection method, 189
 - server.route(), 189
 - server.start() method, 191
 - URL parameter, 190

HTTP methods, 175

implementation

- app.param() method, 181
- coding, 183
- DELETE method, 183
- express.logger(), 181
- find(), 182
- GET method, 186
- POST method, 187
- PUT request handler, 182
- TDD, 187
- terminal window, 185

Mocha and Superagent, test coverage with

- coding, 179
- done(), 178
- error checking, 178
- get(), 178
- HTTP request, 177
- map() function, 178
- test/index.js file, 177
- usage, 176

NPM, 175

\$ npm install command, 176

S

Scaffolding

- app.js middleware, 43
- command-line interface, 40–41
- creation, 40
- Express.js App, configuration of, 44
- Jade template engine, 44
- routes, 43
- terminal command, 40

Session-based authentication, 131

- express.cookieParser(), 131
- express.session(), 131

setMode() method, 211

setTheme() method, 211

Simple Object Access Protocol (SOAP), 174

Socket.IO, 195, 218

and Express.js

- catching and processing, 201
- HTTP requests, 202
- Jade template, 204
- messageChange event listener, 203
- package.json and \$ npm install, 202
- server logs, 200

in production, 218–219

T

Template engines, 71

Handlebars

- comments, 87
- custom helpers, 87
- each iteration, 84
- expressions, 83
- Express.js 4, 92
- if condition, 85
- includes/partials templates, 88
- syntax, 83
- unescaped output, 84
- unless statement, 86
- usages, 88
- With statement, 86

Jade, 71

- admin.jade, 101
- article.jade, 97
- attributes, 73
- block blockname, 79
- case statement, 77
- comments, 75
- Express.js 4, 91
- extend filename, 79
- filters, 77
- if statement, 76
- include, 78
- index.jade, 95

Template engines (*cont.*)
 interpolation, 77
 iterations, 76
 JavaScript code, 75
 layout.jade, 93
 literals, 74
 login.jade, 98
 mixins functions, 77
 post.jade, 99
 script/style tags, 74
 syntax and features, 71
 tags, 71
 text, 74
 usages, 79
 variables/locals, 72
Test-driven development (TDD), 57
Token-based authentication, 130
TravisCI
 configuration file, 240–241
 definition, 240
 \$ make test command, 241

■ **U**

Upstart scripts, 253

■ **V**

Varnish cache, 259
Views, 210
Virtual fields, 159

■ **W, X, Y, Z**

WebSocket, 195
 browser implementation
 HTML tags, 196–197
 script tag, 196
 definition, 195
 Node.js Server, 197
 polling, 195
Winston, 230