

## Contents

Foreword	xv
Preface	xvii
Acknowledgments	xix
About this book	xxi
About the author	xxix

### Unit 0

---

#### GETTING SET UP 1

<b>Lesson 0</b>	Setting up Node.js and the JavaScript engine	3
	What you're going to learn	3
	Understanding Node.js	5
	Why learn to develop in Node.js?	9
	Preparing yourself for this book	10
	Summary	11
<b>Lesson 1</b>	Configuring your environment	12
	Installing Node.js	12
	Installing a text editor	16
	Setting up SCM and deployment tools	17
	Working with the Node.js REPL in terminal	19
	Summary	21
<b>Lesson 2</b>	Running a Node.js application	22
	Creating a JavaScript file	23
	Running your JavaScript file with Node.js	24
	Running individual JavaScript commands	25
	Summary	27

**Unit 1****GETTING STARTED WITH NODE.JS 29**

- Lesson 3** Creating a Node.js module 31  
 Running npm commands 33  
 Initializing a Node.js application 35  
 Summary 39
- Lesson 4** Building a simple web server in Node.js 40  
 Understanding web servers 41  
 Initializing the application with npm 43  
 Coding the application 43  
 Running the application 47  
 Summary 48
- Lesson 5** Handling incoming data 49  
 Reworking your server code 50  
 Analyzing request data 51  
 Adding routes to a web application 55  
 Summary 58
- Lesson 6** Writing better routes and serving external files 59  
 Serving static files with the fs module 60  
 Serving assets 64  
 Moving your routes to another file 67  
 Summary 72
- Lesson 7** Capstone: Creating your first web application 73  
 Initializing the application 74  
 Understanding application directory structure 75  
 Creating main.js and router.js 76  
 Creating views 79  
 Adding assets 80  
 Creating routes 81  
 Summary 83

**Unit 2****EASIER WEB DEVELOPMENT WITH EXPRESS.JS 85**

- Lesson 8** Setting up an app with Express.js 87  
 Installing the Express.js package 88  
 Building your first Express.js application 90

- Working your way around a web framework 92  
 Summary 94
- Lesson 9** Routing in Express.js 95  
 Building routes with Express.js 96  
 Analyzing request data 98  
 Using MVC 101  
 Summary 105
- Lesson 10** Connecting views with templates 106  
 Connecting a templating engine 107  
 Passing data from your controllers 110  
 Setting up partials and layouts 111  
 Summary 113
- Lesson 11** Configurations and error handling 114  
 Modifying your start script 115  
 Handling errors with Express.js 116  
 Serving static files 119  
 Summary 119
- Lesson 12** Capstone: Enhancing the Confetti Cuisine site with Express.js 121  
 Initializing the application 121  
 Building the application 123  
 Adding more routes 124  
 Routing to views 125  
 Serving static views 127  
 Passing content to the views 128  
 Handling the errors 129  
 Summary 131

**Unit 3****CONNECTING TO A DATABASE 133**

- Lesson 13** Setting up a MongoDB database 135  
 Setting up MongoDB 136  
 Running commands in the MongoDB shell 140  
 Connecting MongoDB to your application 144  
 Summary 146
- Lesson 14** Building models with Mongoose 147  
 Setting up Mongoose with your Node.js application 148  
 Creating a schema 149  
 Organizing your models 151  
 Summary 153

- Lesson 15** Connecting controllers and models 155  
     Creating a controller for subscribers 156  
     Saving posted data to a model 159  
     Using promises with Mongoose 162  
     Summary 166
- Lesson 16** Capstone: Saving user subscriptions 167  
     Setting up the database 168  
     Modeling data 168  
     Adding subscriber views and routes 171  
     Summary 173
- Unit 4**
- 
- BUILDING A USER MODEL 175**
- Lesson 17** Improving your data models 177  
     Adding validations on the model 178  
     Testing models in REPL 182  
     Creating model associations 184  
     Populating data from associated models 188  
     Summary 192
- Lesson 18** Building the user model 193  
     Building the user model 194  
     Adding CRUD methods to your models 199  
     Building the index page 202  
     Cleaning up your actions 205  
     Summary 206
- Lesson 19** Creating and reading your models 208  
     Building the new user form 209  
     Creating new users from a view 211  
     Reading user data with show 214  
     Summary 218
- Lesson 20** Updating and deleting your models 219  
     Building the edit user form 220  
     Updating users from a view 223  
     Deleting users with the delete action 226  
     Summary 229
- Lesson 21** Capstone: Adding CRUD models to Confetti Cuisine 230  
     Getting set up 231  
     Building the models 231

- Creating the views 237  
 Structuring routes 242  
 Creating controllers 243  
 Summary 247

## Unit 5

### AUTHENTICATING USER ACCOUNTS 249

- Lesson 22** Adding sessions and flash messages 251  
     Setting up flash message modules 252  
     Adding flash messages to controller actions 254  
     Summary 258
- Lesson 23** Building a user login and hashing passwords 260  
     Implementing the user login form 261  
     Hashing passwords 265  
     Adding validation middleware with express-validator 269  
     Summary 272
- Lesson 24** Adding user authentication 274  
     Implementing Passport.js 275  
     Modifying the create action to use passport registration 279  
     Authenticating users at login 280  
     Summary 284
- Lesson 25** Capstone: Adding user authentication to Confetti Cuisine 285  
     Getting set up 286  
     Creating a login form 286  
     Adding encryption with Passport.js 288  
     Adding flash messaging 289  
     Adding validation middleware with express-validator 291  
     Adding authentication with Passport.js 293  
     Logging in and out 294  
     Summary 296

## Unit 6

### BUILDING AN API 297

- Lesson 26** Adding an API to your application 299  
     Organizing your routes 300  
     Creating an API 304

- Calling your API from the client 306
  - Summary 310
  - Lesson 27** Accessing your API from your application 311
    - Applying an API namespace 312
    - Joining courses via modal 315
    - Creating an API endpoint to connect models 318
    - Summary 321
  - Lesson 28** Adding API security 323
    - Implementing simple security 324
    - Adding API tokens 325
    - Using JSON web tokens 328
    - Summary 333
  - Lesson 29** Capstone: Implementing an API 334
    - Restructuring routes 335
    - Adding the courses partial 338
    - Creating the AJAX function 339
    - Adding an API endpoint 341
    - Creating an action to enroll users 344
    - Summary 347
- Unit 7**
- 
- ADDING CHAT FUNCTIONALITY 349**
- Lesson 30** Working with Socket.io 351
    - Using socket.io 352
    - Creating a chat box 355
    - Connecting the server and client 357
    - Summary 360
  - Lesson 31** Saving chat messages 361
    - Connecting messages to users 362
    - Displaying user names in chat 366
    - Creating a message model 369
    - Summary 372
  - Lesson 32** Adding a chat notification indicator 373
    - Broadcasting to all other sockets 374
    - Creating a chat indicator in navigation 376
    - Summary 378
  - Lesson 33** Capstone: Adding a chat feature to Confetti Cuisine 379
    - Installing socket.io 380
    - Setting up socket.io on the server 380

- Setting up socket.io on the client 381
- Creating a Message model 384
- Loading messages on connection 386
- Setting up the chat icon 387
- Summary 388

## Unit 8

### DEPLOYING AND MANAGING CODE IN PRODUCTION 389

- Lesson 34** Deploying your application 391
  - Preparing for deployment 392
  - Deploying your application 394
  - Setting up your database in production 396
  - Summary 397
- Lesson 35** Managing in production 398
  - Loading seed data 399
  - Linting 401
  - Debugging your application 404
  - Summary 407
- Lesson 36** Testing your application 408
  - Basic testing with core modules 409
  - Testing with mocha and chai 411
  - Testing with a database and server 415
  - Summary 419
- Lesson 37** Capstone: Deploying Confetti Cuisine 420
  - Linting and logging 421
  - Preparing for production 421
  - Deploying to Heroku 422
  - Setting up the database 424
  - Debugging in production 427
  - Summary 427
- Appendix A** JavaScript syntax introduced in ES6 429
- Appendix B** Logging and using Node.js global objects 436
- Index 439