

Table of Contents

Preface

xvii

Part 1: Node.js Overview and JavaScript Language

1

Introduction to Node.js

3

Technical requirements	3	I/O needs to be done differently	6
Why is Node.js so popular?	3	Non-blocking I/O	6
Lightweight and fast	4	Going deeper into the event loop	8
Cross-platform and multi-purpose	4	Node.js versions	8
Easy learning curve	4	Semantic versioning (SemVer)	8
Ecosystem	5	Release details	9
Community-powered	5	Release schedule	10
The Node.js single-thread architecture	6	Summary	12
		Further reading	12

2

Setting Up the Development Environment

13

Technical requirements	13	Other environments	15
Installing Node.js on macOS, Windows, and Linux	14	Verifying the installation	16
macOS	14	Managing Node.js versions	16
Windows	15	Managing Node.js by using nvm	17
Linux	15	Installing and using versions	18
		Using a .nvmrc file	18

Node.js REPL	19	The Network panel	21
Interacting with JavaScript using Chrome DevTools	20	Using Chrome DevTools	21
The Console panel	21	Summary	22
		Further reading	23

3

JavaScript Fundamentals 25

Technical requirements	25	Exploring numbers	33
JavaScript is a powerful language	26	Arithmetic operators	34
Understanding versioning – TC39	26	Assignment operators	34
Understanding versioning – ECMAScript	27	Useful methods	35
What is included in the next version of JavaScript?	27	The Math object	35
What is not included in the JavaScript specification?	27	Other numbers	35
Exploring JavaScript documentation	28	Exploring Dates object	36
Linting JavaScript code	28	Conditional statements	37
Commenting JavaScript code	29	Math comparison operators	37
Using JSDoc	29	Equality operators	37
Printing values and debugging	30	Logical operators	38
Variables and constants	31	The NOT operator (!)	38
Naming conventions	31	Equality in JavaScript	39
let versus const	31	The nullish coalescing operator (??)	40
Understanding data types	32	The if statement	40
undefined	32	The switch statement	40
object	32	Ternary operator	41
boolean	33	Understanding loops	42
number	33	while	42
string	33	do...while	42
function	33	for	42
bigint	33	Using strings in JavaScript	43
symbol	33	Important methods	44
		Exploring arrays	45
		Basic operations	45
		Iterating over an array	47
		Destructuring	50

Sets	50	Arrow functions	59
Using objects in JavaScript	51	Closures	60
Basic operations	51	Creating and managing classes	62
Iteration	52	Creating a class	62
Shallow copy versus deep copy	53	Class methods	62
Merging objects	55	Extending classes	63
Destructuring	56	Static methods	63
Optional chaining (?.)	56	Getters and setters	64
Exploring functions	57	Summary	64
The basics	57	Further reading	65

4

Asynchronous Programming 67

Technical requirements	67	Callback hell with promises	76
Asynchronous programming in JavaScript	67	Parallel promises	77
The asynchronous mindset	68	Error handling	79
Understanding callbacks	69	Chaining promises	80
Timers and intervals	70	Using async and await to handle asynchronous code	80
Error first callbacks	72	async	80
Callback hell	73	await	81
Mastering promises	74	IIFEs	82
Using promises	75	Summary	83
Creating promises	75	Further reading	83

Part 2: Node.js Ecosystem and Architecture

5

Node.js Core Libraries 87

Technical requirements	88	CommonJS (CJS)	89
Modularizing your code (ESM versus CJS)	88	ECMAScript Modules (ESM)	91
		Understanding how interoperability works	93

Structuring core libraries	95	Command-line options	99
The library structure	95	Summary	100
Stability index	97	Further reading	100
Other core libraries	98		

6

External Modules and npm 103

Technical requirements	104	Using npm scripts	117
Managing the application with package.json	104	Executing packages directly with NPX	117
Choosing the correct dependencies for your project	105	npm alternatives	119
Risks	106	Yarn	119
Good criteria	107	PNPM	120
Installing dependencies	111	Publishing your first package	120
Locally or globally	111	Registries	120
Dependency or development dependency	111	npm account	121
Adding new dependency	112	Prepare the package	121
Installing all the dependencies	113	Reviewing the package	122
Removing dependencies	114	Publish the package	123
Understanding the package-lock.json	114	Avoid the scoped packages	124
Managing dependencies versions	115	Release new versions	124
Outdated dependencies	115	Prevent accidental publishing	126
		Best practices	126
Building Isomorphic JavaScript	116	Summary	127
		Further reading	127

7

Event-Driven Architecture 129

Technical requirements	130	The Node.js event emitter library	131
Introducing events	130	Preventing chaos by organizing the listeners	133
Watching for file changes	130		

Removing listeners when they are not needed	133	Adding an event layer to your modules	134
Your first HTTP server	133	Summary	135
		Further reading	136

8

Testing in Node.js 137

Technical requirements	137	Writing our first test suite	144
Why is testing important?	138	Utils module	144
Testing is a complex culture	138	Test core library	144
Testing has to be a team activity	138	Using the Jest library	146
Take advantage now	138	Mastering code coverage tooling	149
Testing approach and philosophy	139	Configuration	149
Testing is wildly used in the world	139	Running the tests	150
The test pyramid	141	Node.js report	150
How should I test my code?	142	TDD in action	153
Frameworks and libraries	142	Summary	155
Our first test	142	Further reading	156
Testing principles and goals	143		

Part 3: Web Application Fundamentals

9

Handling HTTP and REST APIs 159

Technical requirements	159	Request and response	164
How the internet works under the hood	160	Server-side rendering	166
History of the internet	160	Single-Page Applications (SPAs)	166
Internet Infrastructure	161	Mastering HTTP	167
Request for Comments (RFC)	161	HTTP headers	167
HTTP – server and client relationship	164	Status codes	169
		Request methods	171
		HTTP payloads	173

HTTP versions	173	Exploring the JSON specification	177
Using URLs in web applications	174	Debugging HTTP requests	178
REST APIs in a nutshell	175	Summary	179
		Further reading	179

10

Building Web Applications with Express 181

Technical requirements	181	Mastering responses	194
Getting familiar with the Express library	182	Header management	194
Installing Express	182	Status codes	195
Hello World	182	Redirects	196
Using the generator	184	Sending data	197
Debugging	185	Sending files	198
Understanding template engines	186	Using the middleware pattern	199
Choosing a template engine	187	Understanding the scope	201
Rendering a template	187	Middleware anatomy	201
Understanding the process	189	Adding middleware to the application	203
Mastering requests	189	Adding middleware to a route	203
HTTP methods	190	Common middleware used in Express	204
Routes	190	Using third-party middleware	205
Handlers	193	Summary	206
		Further reading	206

Part 4: Building Solid Web Applications with Node.js

11

Building a Web Application Project from Scratch 209

Technical requirements	209	Building a REST API	212
The project Kickoff	210	Adding the routes	212
The project goal	210	Adding the stores	214
Starting with the first step	211	Adding the static files	219

Adding the templates	219
Testing with supertest	221
Adding stores to the server	221
Creating test utils	222
Adding server tests	224

Reviewing the final result of the project	230
The about page	231
Let's celebrate it!	234
Next steps	234
Summary	234

12

Data Persistence with MongoDB 235

Technical requirements	235	dotenv	243
What is MongoDB?	236	Using an ORM – Mongoose	243
Versions	236	Migrating a web application to MongoDB	244
Setting up MongoDB	237	Installing dependencies	244
Installing Docker	237	Managing the secrets	244
Running MongoDB with a container	237	Migrating the data layer	246
Running MongoDB with Docker Compose	239	Testing our MongoDB integration layer	249
Adding Docker commands to package.json	240	Update the utilities	249
Connecting to MongoDB	240	Refactoring the test suite	250
Other ways to install MongoDB	241	Some tests must change	251
How to load secrets in Node.js	241	Checking the coverage	252
Environment variables	242	Summary	255
The .env file	242	Further reading	255

13

User Authentication and Authorization with Passport.js 257

Technical requirements	258	JWT in a nutshell	260
Understanding modern authentication and authorization	258	The process	260
Authentication	258	The theory	260
Authorization	259	Cryptography 101	261
		JWT structure	262
		JWT.io	263

Understanding Passport.js fundamentals	264	Changes in the frontend	270
Adding authentication and authorization to our web application	264	Added a new store for users	272
Clone the base project	264	JWT utilities	273
Setup	265	Adding new routes	275
Run the tests	265	Improved testing utilities	276
The UI changes	266	Test case changes	277
Added dependencies	270	Test coverage	277
		Summary	278
		Further reading	278

14

Error Handling in Node.js 279

Technical requirements	280	Custom errors	284
Exploring the types of errors	280	Gracefully shutting down the application	284
Syntax errors	280	Events	284
Runtime errors	280	Exit codes	286
Throwing meaningful errors	281	Avoiding zombie processes	286
The error object	281	Summary	287
Custom errors	282	Further reading	287
Managing errors in Express	283		
Error-handling middleware	283		

15

Securing Web Applications 289

Technical requirements	289	CWE	294
The importance of security	290	CVE	294
Social engineering	290	Node.js threat model	296
Risks in the supply chain	291	The Node.js official recommendations	297
The business impact	292	Improving the security of our applications	297
Where to start with security	292	Put it into practice	300
OWASP Top 10 overview	292		

Becoming an ethical hacker	300	Getting the skills	302
Coordinated vulnerability disclosure (CVD)	301	Summary	303
Bounty programs	301	Further reading	303

Part 5: Mastering Node.js Deployment and Portability

16

Deploying Node.js Applications 307

Technical requirements	307	Creating a DigitalOcean Droplet	318
Defining the requirements	308	Connecting to the machine	319
Using a GitHub repository	309	Preparing the machine	319
Creating a GitHub repository	310	Cloning the repository	320
Pushing the code to the repository	310	Installing the dependencies	321
Continuous integration with GitHub Actions	312	Preparing the environment	321
Using MongoDB Atlas	316	Managing the application with PM2	321
Deploying Node.js applications with PM2	317	Accessing the application	322
		Summary	323
		Further reading	323

17

Dockerizing a Node.js Application 325

Technical requirements	325	Running the containers	332
Containers and cloud-native solutions with Docker	326	Using Cloudflare	334
Docker lifecycle	326	Next level – Twelve-Factor App principles	335
Dockerizing the application	327	Cleaning up	336
Managing the application with Docker	328	Summary	337
Adding docker build to the CI	329	Further reading	337
Pushing the image to Docker Hub	329		
Publishing the image with GitHub Actions	331		

Index 339

Other Books You May Enjoy 354