Node. js for Beginners : A Comprehensive Guide to Building Efficient, Full-Featured Web Applications with Node. js /

Gascón, Ulises,
author

Monografía

Build your first production-grade web application from scratch using Node.js, Express, MongoDB, authentication, and testing with the help of expert guidance and step-by-step explanations Key Features Learn JavaScript, async programming, and event-driven architectures in Node.js Discover best practices for secure deployment and testing of Node.js apps Build a feature-rich web app from scratch using MongoDB, Express.js, and passport.js, including testing Purchase of the print or Kindle book includes a free PDF eBook Book Description Node.js is a runtime environment that revolutionizes server-side JavaScript development and combines high performance, a unified language stack, a rich ecosystem of packages, and scalability, empowering developers to build efficient, scalable, and versatile applications across a range of use cases. Node. js for Beginners takes you on a progressive learning path, starting with the basics and gradually advancing to complex topics that will give you the skills needed to leverage Node.js. First, you'll delve into Node.js fundamentals and refresh your knowledge of JavaScript before transitioning to the development of modern web applications using Node.js, Express.js, and MongoDB. You'll explore and get hands-on with complex topics such as asynchronous programming, npm libraries, event-driven patterns for HTTP servers, RESTful API, JSON Web Tokens, and more. The final chapters will guide you through the deployment process, offering insights into various strategies, including bare metal setups, virtual machines, and containerization with Docker. You'll also be guided through the use of Process Manager 2 (PM2) for efficient process management. By the end of this Node.js book, you'll have the skills to craft, test, and deploy Node.js web applications confidently and use this powerful stack in your day-to-day projects. What you will learn Build solid and secure Node.js applications from scratch Discover how to consume and publish npm packages effectively Master patterns for refactoring and evolving your applications over time Gain a deep understanding of essential web development principles, including HTTP, RESTful API design, JWT, authentication, authorization, and error handling Implement robust testing strategies to enhance the quality and reliability of your applications Deploy your Node.js applications to production environments using Docker and PM2 Who this book is for This book is for developers eager to learn Node.js swiftly and utilize it for web app development. Front-end developers looking to delve into back-end development or become full-stack developers using Node.js will also benefit from this book's comprehensive coverage of topics. Before you get started with this book, you'll need to be familiar with the basics of coding (JavaScript programming or another language), Git and/or GitHub, and front-end development, as these topics aren't covered in the book

Título: Node. js for Beginners A Comprehensive Guide to Building Efficient, Full-Featured Web Applications with Node. js Ulises Gascón

Edición: First edition

Editorial: Birmingham Packt Publishing [2024] 2024

Descripción física: 1 online resource (383 pages)

Contenido: Cover -- Title Page -- Copyright -- Dedication -- Contributors -- Table of Contents -- Preface -- Part 1: Node.js Overview and JavaScript Language -- Chapter 1: Introduction to Node.js -- Technical requirements -- Why is Node.js so popular? -- Lightweight and fast -- Cross-platform and multi-purpose -- Easy learning curve -- Ecosystem -- Community-powered -- The Node.js single-thread architecture -- I/O needs to be done differently -- Non-blocking I/O -- Going deeper into the event loop -- Node.js versions -- Semantic versioning (SemVer) -- Release details -- Release schedule -- Summary -- Further reading -- Chapter 2: Setting Up the Development Environment -- Technical requirements -- Installing Node.js on macOS, Windows, and Linux -- macOS -- Windows -- Linux -- Other environments -- Verifying the installation -- Managing Node.js versions -- Managing Node.js by using nvm -- Installing and using versions -- Using a .nvmrc file -- Node.js REPL -- Interacting with JavaScript using Chrome DevTools -- The Console panel -- The Network panel -- Using Chrome DevTools -- Summary -- Further reading -- Chapter 3: JavaScript Fundamentals -- Technical requirements -- JavaScript is a powerful language -- Understanding versioning - TC39 -- Understanding versioning - ECMAScript -- What is included in the next version of JavaScript? -- What is not included in the JavaScript specification? -- Exploring JavaScript documentation -- Linting JavaScript code -- Commenting JavaScript code -- Using JSDoc -- Printing values and debugging -- Variables and constants -- Naming conventions -- let versus const -- Understanding data types -- undefined -- object -- boolean -- number -- string -- function -- bigint -- symbol -- Exploring numbers -- Arithmetic operators -- Assignment operators -- Useful methods -- The Math object -- Other numbers -- Exploring Dates object Conditional statements -- Math comparison operators -- Equality operators -- Logical operators -- The NOT operator (!) -- Equality in JavaScript -- The nullish coalescing operator (??) -- The if statement -- The switch statement -- Ternary operator -- Understanding loops -- while -- do...while -- for -- Using strings in JavaScript -- Important methods -- Exploring arrays -- Basic operations -- Iterating over an array -- Destructuring -- Sets -- Using objects in JavaScript -- Basic operations -- Iteration -- Shallow copy versus deep copy -- Merging objects -- Destructuring -- Optional chaining (?.) -- Exploring functions -- The basics -- Arrow functions -- Closures -- Creating and managing classes -- Creating a class -- Class methods -- Extending classes -- Static methods -- Getters and setters -- Summary -- Further reading -- Chapter 4: Asynchronous Programming -- Technical requirements -- Asynchronous programming in JavaScript -- The asynchronous mindset -- Understanding callbacks -- Timers and intervals -- Error first callbacks -- Callback hell -- Mastering promises -- Using promises -- Creating promises -- Callback hell with promises -- Parallel promises -- Error handling -- Chaining promises -- Using async and await to handle asynchronous code -- async -- await -- IIFEs -- Summary -- Further reading -- Part 2: Node.js Ecosystem and Architecture -- Chapter 5: Node.js Core Libraries -- Technical requirements -- Modularizing your code (ESM versus CJS) -- CommonJS (CJS) -- ECMAScript Modules (ESM) -- Understanding how interoperability works -- Structuring core libraries -- The library structure -- Stability index -- Other core libraries -- Command-line options -- Summary -- Further reading -- Chapter 6: External Modules and npm -- Technical requirements -- Managing the application with package.json Choosing the correct dependencies for your project -- Risks -- Good criteria -- Installing dependencies -- Locally or globally -- Dependency or development dependency -- Adding new dependency -- Installing all the dependencies -- Removing dependencies -- Understanding the package-lock.json -- Managing dependencies versions -- Outdated dependencies -- Building Isomorphic JavaScript -- Using npm scripts -- Executing packages directly with NPX -- npm alternatives -- Yarn -- PNPM -- Verdaccio -- Publishing your first package -- Registries -- npm account -- Prepare the package -- Reviewing the package -- Publish the package -- Avoid the scoped packages -- Release new versions -- Prevent accidental publishing -- Best practices -- Summary -- Further reading -- Chapter 7: Event-Driven Architecture -- Technical requirements -- Introducing events -- Watching for file changes -- The Node.js event emitter library -- Preventing chaos by organizing the listeners -- Removing listeners when they are not needed -- Your first HTTP server -- Adding an event layer to your modules

-- Summary -- Further reading -- Chapter 8: Testing in Node.js -- Technical requirements -- Why is testing important? -- Testing is a complex culture -- Testing has to be a team activity -- Take advantage now -- Testing approach and philosophy -- Testing is wildly used in the world -- The test pyramid -- How should I test my code? -- Frameworks and libraries -- Our first test -- Testing principles and goals -- Writing our first test suite -- Utils module -- Test core library -- Using the Jest library -- Mastering code coverage tooling -- Configuration -- Running the tests -- Node.js report -- TDD in action -- Summary -- Further reading -- Part 3: Web Application Fundamentals -- Chapter 9: Handling HTTP and REST APIs -- Technical requirements -- How the internet works under the hood History of the internet -- Internet Infrastructure -- Request for Comments (RFC) -- HTTP - server and client relationship -- Request and response -- Server-side rendering -- Single-Page Applications (SPAs) -- Mastering HTTP -- HTTP headers -- Status codes -- Request methods -- HTTP payloads -- HTTP versions -- Using URLs in web applications -- REST APIs in a nutshell -- Exploring the JSON specification -- Debugging HTTP requests -- Summary -- Further reading -- Chapter 10: Building Web Applications with Express -- Technical requirements -- Getting familiar with the Express library -- Installing Express -- Hello World -- Using the generator -- Debugging -- Understanding template engines -- Choosing a template engine -- Rendering a template -- Understanding the process -- Mastering requests -- HTTP methods -- Routes -- Handlers -- Mastering responses -- Header management -- Status codes -- Redirects -- Sending data -- Sending files -- Using the middleware pattern -- Understanding the scope -- Middleware anatomy -- Adding middleware to the application -- Adding middleware to a route -- Common middleware used in Express -- Using third-party middleware -- Summary -- Further reading -- Part 4: Building Solid Web Applications with Node.js -- Chapter 11: Building a Web Application Project from Scratch -- Technical requirements -- The project Kickoff -- The project goal -- Starting with the first step -- Building a REST API -- Adding the routes -- Adding the stores -- Adding the static files -- Adding the templates -- Testing with supertest -- Adding stores to the server -- Creating test utils -- Adding server tests -- Reviewing the final result of the project -- The about page -- Let's celebrate it! -- Next steps -- Summary -- Chapter 12: Data Persistence with MongoDB -- Technical requirements -- What is MongoDB? -- Versions -- Setting up MongoDB Installing Docker -- Running MongoDB with a container -- Running MongoDB with Docker Compose -- Adding Docker commands to package.json -- Connecting to MongoDB -- Other ways to install MongoDB -- How to load secrets in Node.js -- Environment variables -- The .env file -- dotenv -- Using an ORM - Mongoose -- Migrating a web application to MongoDB -- Installing dependencies -- Managing the secrets -- Migrating the data layer -- Testing our MongoDB integration layer -- Update the utilities -- Refactoring the test suite -- Some tests must change -- Checking the coverage -- Summary -- Further reading -- Chapter 13: User Authentication and Authorization with Passport.js -- Technical requirements -- Understanding modern authentication and authorization -- Authentication -- Authorization -- JWT in a nutshell -- The process -- The theory -- Cryptography 101 -- JWT structure -- JWT.io -- Understanding Passport.js fundamentals -- Adding authentication and authorization to our web application -- Clone the base project -- Setup -- Run the tests -- The UI changes -- Added dependencies -- Changes in the frontend -- Added a new store for users -- JWT utilities -- Adding new routes -- Improved testing utilities -- Test case changes -- Test coverage -- Summary -- Further reading -- Chapter 14: Error Handling in Node.js -- Technical requirements -- Exploring the types of errors -- Syntax errors -- Runtime errors -- Throwing meaningful errors -- The error object -- Custom errors -- Managing errors in Express -- Error-handling middleware -- Custom errors -- Gracefully shutting down the application -- Events -- Exit codes -- Avoiding zombie processes -- Summary -- Further reading -- Chapter 15: Securing Web Applications -- Technical requirements -- The importance of security -- Social engineering -- Risks in the supply chain -- The business impact Where to start with security

ISBN: 1-80323-814-3

Materia Título preferido: Node.js

Materia: Application software- Development JavaScript (Computer program language)

Enlace a formato físico adicional: 1-80324-517-4

Baratz Innovación Documental

- Gran Vía, 59 28013 Madrid

- (+34) 91 456 03 60
- informa@baratz.es