NODE JS

Requirements

- A computer on which you can install software (Windows, MacOS, or Linux) is required
- A basic understanding of vanilla JavaScript (variables, if statements, basic functions, basic objects)

During eight chapters you'll learn:

1. **Node**
2. **Npm**
3. **Asynchronous programming**
4. **MongoDB**
5. **Express**
6. **Socket.IO**
7. **Authentication with JWTs**
8. **Mongoose**
9. **Deploying apps with Heroku**
10. **Version control with Git**
11. **GitHub**
12. **Code testing**
13. **Application debugging**
14. **Awesome new ES6 features**
15. **Mocha**
16. **And more!**

Who is the target audience?

- Anyone looking to launch their own Node applications, switch careers, or freelance as a Node developer

Welcome

Welcome & Asking Awesome Questions

Getting Setup

Section Intro

Installing Node

What Is Node?

Why Should I Use Node?

Installing Atom

Hello World!

Node.js Fundamentals (Note App)

Section Intro

Using Require

Requiring Your Own Files

Using 3rd Party Modules

Restarting App with Nodemon

Getting Input From User

Simplified Input With Yargs

Working With JSON

Adding and Saving Notes

Refactoring For Reusability

Removing a Note

Reading Notes and Reusability

Debugging Node.js Applications

Debugging via Chrome Dev Tools

Listing Notes

Requiring Arguments and Advanced Yargs

Arrow Functions

Asynchronous Node.js (Weather App)

Section Intro

Async Basics

Call Stack & Event Loop

Callback Functions & APIs

Pretty Printing Objects

What's Makes up an HTTP Request?

Encoding User Input

Callback Errors

Abstracting Callbacks

Wiring Up Weather Search

Chaining Callbacks Together

Intro to ES6 Promises

Advanced Promises

Promises Quiz

6 questions

Weather App With Promises

Extra Features

Web Servers and Application Deployment

Section Intro

Hello Express

Creating a Web Server

Rendering Templates with Data

Advanced Templating

Express Middleware

Adding Version Control (Git)

Setting Up GitHub & SSH Keys

Deploying Your Apps

Adding a New Feature and Deploying

Testing Your Applications

Section Intro

Mocha and Basic Testing

Watch and Auto Restart Tests

Using an Assertion Library

Testing Asynchronous Code

Testing Express Applications: Part I

- **Project Setup**
- **Database Setup**
- **Test Structure**

Testing Express Applications: Part II

- **Unit Tests**
- **Test Fixtures**
- **Validation**
- **Refactor**

Organizing Test With describe()

Test Spies

MongoDB, Mongoose, and REST APIs (Todo API)

Section Intro

Installing MongoDB and Robomongo (Mac, Linux)

Installing MongoDB and Robomongo (Windows)

Building a NoSQL Vocabulary

Connecting to Mongo and Writing Data

The ObjectId

Fetching Data

Setting Up The Repo

Deleting Documents

Updating Data

The Mongoose ORM

Setting Up Mongoose

Validators, Types, and Defaults

Installing Postman

Resource Creation Endpoint - POST /todos

Testing POST /todos

List Resources - GET /todos

Testing GET /todos

Mongoose Queries and ID Validation

Getting an Individual Resource - GET /todos/:id

Testing GET /todos/:id

Deploy API to Heroku

Postman Environments

Delete a Resource - DELETE /todos/:id

Testing DELETE /todos/:id

Updating a Resource - PATCH /todos/:id

Testing PATCH /todos/:id

Creating a Test Database

Security and Authentication

Section Intro

Setting Up The User Model

JWTs and Hashing

Generating Auth Tokens and Setting Headers

Private Routes and Auth Middleware

Hashing Passwords

Seeding Test Database With Users

Testing POST /users and GET /users/me

Logging In - POST /users/login

Testing POST /users/login

Logging Out - DELETE /users/me/token

Testing DELETE /users/me/token

Making Todo Routes Private (Part 1)

Making Todo Routes Private (Part 2)

Improving App Configuration

Deploying To Heroku

Bonus: Advanced Postman

Real-Time Web Apps with Socket.io

Section Intro

Creating a New Project

Adding Socket.io to an App

Emitting and Listening to Events

Emitting and Listening to Events (Solution)

Broadcasting Events

Message Generator & Tests

Event Acknowledgements

Message Form & jQuery

Geolocation Part I

Geolocation Part II

Styling The Chat Page

Timestamps and Formatting with Moment

Printing Message Timestamps

Moustache.js

Autoscrolling

Adding a Join Page

Passing Room Data

Socket.io Rooms

Storing Users With ES6 Classes: Part I

Storing Users With ES6 Classes: Part II

Wiring Up User List

Sending Messages To Room Only

New Feature Ideas

Async/Await - Just Added!

Async/Await Project Setup

Async/Await Basics

A Real-World Example

Handling Errors and Awaiting Async Function

Using Async/Await in the Todo API