

Node.js

Description

In this 5-days course, you will learn how to design, build, test, and deploy a RESTful API using Node.js and Mongo. Topics such as working with middleware, testing in Node.js, application organization, data modeling, querying data with Mongoose, using JSON Web Token, securing routes, and deployment.

Course Content

Module 1: Getting Setup

- Lesson 1: Section Intro
- Lesson 2: Installing Node
- Lesson 3: What Is Node?
- Lesson 4: Why Should I Use Node?

Module 2: Node.js Fundamentals

- Lesson 1: Using Require
- Lesson 2: Requiring Your Own Files
- Lesson 3: Using 3rd Party Modules
- Lesson 4: Restarting App with Nodemon
- Lesson 5: Getting Input From User
- Lesson 6: Simplified Input With Yargs
- Lesson 7: Working With JSON
- Lesson 8: Adding and Saving Notes
- Lesson 9: Refactoring For Reusability
- Lesson 10: Removing a Note
- Lesson 11: Debugging Node.js Applications
- Lesson 12: Debugging via Chrome Dev Tools
- Lesson 13: Requiring Arguments and Advanced Yargs
- Lesson 14: Arrow Functions

Module 3: Asynchronous Node.js

- Lesson 1: Async Basics
- Lesson 2: Call Stack & Event Loop
- Lesson 3: Callback Functions & APIs
- Lesson 4: Pretty Printing Objects
- Lesson 5: What's Makes up an HTTP Request?
- Lesson 6: Encoding User Input
- Lesson 7: Callback Errors
- Lesson 8: Abstracting Callbacks
- Lesson 9: Wiring Up Weather Search
- Lesson 10: Chaining Callbacks Together
- Lesson 11: Intro to ES6 Promises
- Lesson 12: Advanced Promises
- Lesson 13: Weather App With Promises
- Lesson 14: Extra Features

Module 4: Web Servers and Application Deployment

- Lesson 1: Hello Express
- Lesson 2: Creating a Web Server
- Lesson 3: Rendering Templates with Data
- Lesson 4: Advanced Templating
- Lesson 5: Express Middleware
- Lesson 6: Adding Version Control (Git)
- Lesson 7: Setting Up GitHub & SSH Keys
- Lesson 8: Deploying Your Apps
- Lesson 9: Adding a New Feature and Deploying

Module 5: Testing Your Applications

- Lesson 1: Mocha and Basic Testing
- Lesson 2: Watch and Auto Restart Tests
- Lesson 3: Using an Assertion Library
- Lesson 4: Testing Asynchronous Code
- Lesson 5: Testing Express Applications: Part I
- Lesson 6: Testing Express Applications: Part II
- Lesson 7: Organizing Test With describe()
- Lesson 8: Test Spies

Module 6: MongoDB, Mongoose, and REST APIs (Todo API)

- Lesson 1: Installing MongoDB and Robomongo (Mac, Linux)
- Lesson 2: Installing MongoDB and Robomongo (Windows)
- Lesson 3: Building a NoSQL Vocabulary
- Lesson 4: Connecting to Mongo and Writing Data
- Lesson 5: The ObjectId
- Lesson 6: Fetching Data
- Lesson 7: Setting Up The Repo
- Lesson 8: Deleting Documents
- Lesson 9: Updating Data
- Lesson 10: The Mongoose ORM
- Lesson 11: Setting Up Mongoose

- Lesson 12: Validators, Types, and Defaults
- Lesson 13: Mongoose Queries and ID Validation
- Lesson 14: Getting an Individual Resource - GET /todos/:id
- Lesson 15: Deploy API to Heroku

Module 7: Security and Authentication

- Lesson 1: Setting Up The User Model
- Lesson 2: JWTs and Hashing
- Lesson 3: Generating Auth Tokens and Setting Headers
- Lesson 4: Private Routes and Auth Middleware
- Lesson 5: Hashing Passwords
- Lesson 6: Seeding Test Database With Users
- Lesson 7: Improving App Configuration

Module 8: Real-Time Web Apps with Socket.io

- Lesson 1: Creating a New Project
- Lesson 2: Adding Socket.io to an App
- Lesson 3: Emitting and Listening to Events
- Lesson 4: Emitting and Listening to Events (Solution)
- Lesson 5: Broadcasting Events
- Lesson 6: Message Generator & Tests
- Lesson 7: Event Acknowledgements
- Lesson 8: Message Form & jQuery
- Lesson 9: Geolocation Part I
- Lesson 10: Geolocation Part II
- Lesson 11: Styling The Chat Page
- Lesson 12: Timestamps and Formatting with Moment
- Lesson 13: Printing Message Timestamps
- Lesson 14: Moustache.js
- Lesson 15: Autoscrolling
- Lesson 16: Adding a Join Page
- Lesson 17: Passing Room Data
- Lesson 18: Socket.io Rooms
- Lesson 19: Storing Users With ES6 Classes: Part I
- Lesson 20: Storing Users With ES6 Classes: Part II
- Lesson 21: Wiring Up User List
- Lesson 22: Sending Messages To Room Only
- Lesson 23: New Feature Ideas

Module 9: Async/Await

- Lesson 1: Async/Await Project Setup
- Lesson 2: Async/Await Basics
- Lesson 3: A Real-World Example
- Lesson 4: Handling Errors and Awaiting Async Function
- Lesson 5: Using Async/Await in the Todo API

Lab / Exercises

- During the course participants are encouraged to actively participate in the learning experience by running example files during lectures and performing coding challenges during labs. Each lab session allows you

to compare your solution to the instructor's

Documentation

- Digital courseware included

Participant profiles

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

Prerequisites

- Basic understanding of vanilla JavaScript (variables, if statements, basic functions, basic objects)

Objectives

- Built and deployed a fully functional API for an app with authentication
- Build their own APIs
- Turn into a professional Node developer capable of developing, testing, and deploying real-world production applications

Niveau

Fondamental

Classroom Registration Price (CHF)

3800

Virtual Classroom Registration Price (CHF)

3550

Duration (in Days)

5

Reference

NODEJS