ELEVO LABS

LEARN. INNOVATE. ELEVATE.

Course Overview

Course Details

Course Name: Node.js - The Complete Backend Guide: Zero to Hero

Instructor: David Chen

Created On: 16 June, 2025

Updated On: 16 June, 2025

Price: 7000 INR

Duration: 6 weeks

Modules: 17

Language: English, Hindi

Level: Beginner, Expert

Certifications: Yes

Course Description

Unlock the power of server-side JavaScript with this ultimate guide to Node.js. This course is meticulously designed to take you from the fundamentals of Node to building and deploying enterprise-grade backend services. Whether you're a front-end developer looking to go full-stack or a newcomer to programming, this is your comprehensive roadmap to becoming a backend professional.

We start with the core concepts: understanding the Node.js runtime, the event loop, and its asynchronous nature. You'll master the module system, work with files, and build your first servers. The course then transitions to the Express.js framework, where you'll learn to build robust REST APIs, handle complex routing, and implement middleware. We cover everything from database integration (both SQL and NoSQL), secure user authentication with JWT, to handling file uploads and real-time communication with WebSockets.

Beyond the basics, you'll dive into advanced topics like structured application design, performance optimization, automated testing, API documentation, security best practices, and containerization with Docker. By the end, you won't just know how to build Node.js applications—you'll know how to build them the right way: scalable, secure, and ready for production.

What You'll Learn:

- Node.js architecture, the V8 engine, and the event loop.
- Asynchronous programming with callbacks, Promises, and Async/Await.
- Working with Node.js core modules and NPM.
- Building scalable REST APIs with Express.js.
- Database integration with both MongoDB (Mongoose) and PostgreSQL (Sequelize).
- Secure user authentication and authorization using JSON Web Tokens (JWT).
- Implementing API security best practices (Helmet.js, CORS, Rate Limiting).
- Real-time applications with WebSockets and Socket.IO.
- Handling file uploads and email sending.
- Structuring Node.js applications with best practices (e.g., MVC, services).
- Testing your backend with Jest and Supertest.
- Documenting your APIs with tools like Swagger/OpenAPI.
- Optimizing Node.js application performance and scalability (caching, clustering).
- Containerizing applications with Docker for easy deployment.
- Deploying Node.js applications to cloud platforms.
- An introduction to TypeScript for type-safe Node.js development.

Who Is This Course For?

This course is designed for JavaScript developers who want to master the backend. It's also perfect for students, career-changers, and anyone looking to build a strong foundation in server-side development. While basic JavaScript knowledge is beneficial, the course starts from the ground up to ensure no one is left behind.

Prerequisites:

- **Solid JavaScript Fundamentals:** A strong understanding of core JavaScript (including ES6+ features, basic object-oriented concepts, and a conceptual understanding of asynchronous programming).
- **Basic Understanding of HTTP & REST:** Familiarity with HTTP methods, status codes, and the concept of RESTful APIs.
- **A Computer/Laptop:** A working computer (Windows, macOS, or Linux) with Node.js and NPM installed.
- **Command Line Basics:** Comfort with basic terminal commands (e.g., `cd`, `ls`, `npm install`).
- **A Willingness to Learn:** Enthusiasm for building robust backend systems.

Features multiple in-depth projects, including a full REST API and a real-time chat app, plus a certificate of completion. Available in English and Hindi.

Course Curriculum Details

Module 1: Module 1: Introduction to Node.js & Environment Setup

5 Lessons

1. 1. What is Node.js? (Key Features and Use Cases) **15min**
2. 2. How Node.js Works: The V8 Engine, Libuv, and the Event Loop (Fundamentals) **30min**
3. 3. Setting Up Your Node.js Development Environment **20min**
4. 4. Your First Node.js Application: 'Hello World' Server **15min**
5. 5. Recap & Practical Task: Run a Basic Node.js Server **30min**

Module 2: Module 2: Node.js Fundamentals & Core Modules

5 Lessons

1. 6. The Module System: CommonJS (`require`/`module.exports`) and ES Modules (`import`/`export`) **25min**

2. 7. The Node Package Manager (NPM) and `package.json` **20min**

3. 8. Core Modules: File System (`fs`) for File Operations **35min**

4. 9. Core Modules: Path, OS, and Events (`EventEmitter`) **30min**

5. 10. Recap & Practical Task: Build a Simple File Reader/Writer **45min**

Module 3: Module 3: Asynchronous Programming in Node.js

5 Lessons

1. 11. Understanding Asynchronicity: Callbacks and Callback Hell **20min**

2. 12. The Modern Approach: Promises for Cleaner Async Code **30min**

3. 13. Simplifying Async Code with Async/Await **30min**

4. 14. Error Handling in Asynchronous Code (Callbacks, Promises, Async/Await) **25min**

5. 15. Recap & Practical Task: Refactor an Async Operation Using Promises and Async/Await **50min**

Module 4: Module 4: Building Web Servers with Express.js

6 Lessons

1. 16. Introduction to Express.js: The Fast, Unopinionated, Minimalist Web Framework **20min**

2. 17. Basic Routing and Handling HTTP Methods (GET, POST, etc.) **30min**

3. 18. Understanding and Writing Middleware (`req`, `res`, `next`) **35min**

4. 19. Using Built-in and Third-Party Middleware (e.g., `express.json()`, Morgan) **25min**

5. 20. Error Handling with Express Middleware **30min**

6. 21. Recap & Practical Task: Create a Simple Express Web Server with Custom Middleware **1hr**

Module 5: Module 5: Advanced Express.js & Application Structuring

7 Lessons

1. 22. Serving Static Files (CSS, Images, HTML) **15min**

2. 23. Organizing Routes with Express Router **25min**

3. 24. Route Parameters and Query Strings **20min**

4. 25. Structuring Express Applications: MVC Pattern for APIs **30min**

5. 26. Separating Business Logic into Services/Managers **25min**

6. 27. Managing Configuration and Environment Variables (`dotenv`) **25min**

7. 28. Recap & Practical Task: Refactor an Express App into a Modular Structure **1hr**

Module 6: Module 6: Building RESTful APIs

7 Lessons

1. 29. Principles of RESTful API Design **20min**

2. 30. Implementing GET Endpoints (Read All & Read One) **35min**

3. 31. Implementing POST Endpoints (Create Resources) **30min**

4. 32. Implementing PUT/PATCH Endpoints (Update Resources) **30min**

5. 33. Implementing DELETE Endpoints (Remove Resources) **20min**

6. 34. Sending Proper JSON Responses and HTTP Status Codes **25min**

7. 35. Recap & Practical Task: Build a Complete CRUD API for a Resource **1hr 15min**

Module 7: Module 7: Data Persistence with NoSQL Databases (MongoDB)

6 Lessons

1. 36. Introduction to MongoDB and Document Databases **20min**

2. 37. Setting up MongoDB (Local or Atlas) and Connecting with Mongoose (ODM) **30min**

3. 38. Defining Schemas and Models with Mongoose **30min**

4. 39. Performing CRUD Operations on MongoDB with Mongoose **40min**

5. 40. Advanced Mongoose Queries and Data Validation **35min**

6. 41. Recap & Practical Task: Integrate MongoDB into Your API **1hr**

Module 8: Module 8: Data Persistence with SQL Databases (PostgreSQL/Sequelize)

6 Lessons

1. 42. Introduction to Relational Databases and SQL Concepts **20min**

2. 43. Setting up PostgreSQL and Connecting with Sequelize (ORM) **30min**

3. 44. Defining Models and Migrations with Sequelize **35min**

4. 45. Performing CRUD Operations on SQL Database with Sequelize **35min**

5. 46. Defining Associations (Relationships) in Sequelize **30min**

6. 47. Recap & Practical Task: Integrate PostgreSQL into Your API **1hr**

Module 9: Module 9: Authentication & Authorization

8 Lessons

1. 48. Understanding User Authentication vs. Authorization **20min**

2. 49. Password Hashing with `bcrypt.js` **25min**

3. 50. Introduction to JWT (JSON Web Tokens): Structure and Purpose **30min**

4. 51. Creating and Signing JSON Web Tokens **35min**

5. 52. Creating Middleware to Verify and Protect Routes with JWT **30min**

6. 53. Implementing Refresh Tokens (Conceptual Overview) **20min**

7. 54. Role-Based Access Control (RBAC) in APIs **25min**

8. 55. Recap & Practical Task: Implement User Registration, Login, and JWT Protected Routes **1hr 30min**

Module 10: Module 10: API Security Best Practices

7 Lessons

1. 56. Securing HTTP Headers with Helmet.js **20min**

2. 57. Enabling CORS (Cross-Origin Resource Sharing) for API Access **25min**

3. 58. Implementing Rate Limiting to Prevent Abuse **25min**

4. 59. Data Validation and Sanitization (e.g., with Joi or Express-Validator) **30min**

5. 60. Preventing SQL Injection (with ORMs) and XSS (Output Encoding) **25min**

6. 61. Handling Sensitive Data (Environment Variables, Secret Management) **20min**

7. 62. Recap & Practical Task: Secure Your API with Helmet.js, CORS, and Rate Limiting **1hr**

Module 11: Module 11: File Uploads & Email Sending

5 Lessons

1. 63. Handling File Uploads with Multer (`multipart/form-data`) **35min**

2. 64. Validating and Processing Uploaded Files **25min**

3. 65. Storing Files Locally vs. Cloud Storage (e.g., AWS S3, Cloudinary - Conceptual) **30min**

4. 66. Sending Automated Emails with Nodemailer **30min**

5. 67. Recap & Practical Task: Implement a User Profile Picture Upload and a Welcome Email **1hr 15min**

Module 12: Module 12: Real-time Communication with WebSockets

5 Lessons

1. 68. Introduction to Real-time Applications and WebSockets **20min**

2. 69. Integrating Socket.IO with an Express Server **35min**

3. 70. Emitting Events and Broadcasting Messages to Clients **30min**

4. 71. Handling Socket.IO Events (Connection, Disconnect, Custom Events) **25min**

5. 72. Recap & Practical Task: Build a Simple Chat Application Backend **1hr**

Module 13: Module 13: Testing Node.js Applications

6 Lessons

1. 73. Testing Fundamentals & Setup (Jest for Unit/Integration Testing) **25min**

2. 74. Writing Unit Tests for Controllers and Services **35min**

3. 75. Writing Integration Tests for API Endpoints with Supertest **40min**

4. 76. Mocking Services and Database Interactions for Tests **30min**

5. 77. Test Driven Development (TDD) Approach (Conceptual) **20min**

6. 78. Recap & Practical Task: Write Tests for Authentication and CRUD Endpoints **1hr 30min**

Module 14: Module 14: API Documentation & Performance Optimization

6 Lessons

1. 79. Importance of API Documentation **15min**

2. 80. Automating Documentation with Swagger/OpenAPI (`swagger-ui-express`) **35min**

3. 81. Caching Strategies with Redis for Performance **30min**

4. 82. Scaling on Multi-Core Machines with the Cluster Module **25min**

5. 83. Using a Process Manager like PM2 (Clustering, Monitoring) **20min**

6. 84. Recap & Practical Task: Add Swagger Docs and Implement Caching for an Endpoint **1hr 15min**

Module 15: Module 15: Containerization with Docker & Deployment

9 Lessons

1. 85. Introduction to Docker and Containers **25min**

2. 86. Creating a Dockerfile for a Node.js App **30min**

3. 87. Managing Multi-Service Applications with Docker Compose **30min**

4. 88. Preparing for Production: Environment Variables and Security Checklist **25min**

5. 89. Deploying to PaaS (Heroku, Render - Conceptual Overview) **30min**

6. 90. Deploying to IaaS (AWS EC2, DigitalOcean - Conceptual Overview) **40min**

7. 91. Setting up Nginx as a Reverse Proxy (Conceptual) **20min**

8. 92. Continuous Integration/Continuous Deployment (CI/CD) Overview **20min**

9. 93. Recap & Practical Task: Dockerize Your API and Prepare for Deployment **1hr 30min**

Module 16: Module 16: Advanced Topics & Ecosystem

4 Lessons

1. 94. An Introduction to TypeScript with Node.js **30min**

2. 95. Building a Basic GraphQL API (Overview) **45min**

3. 96. Working with Microservices (Conceptual Overview) **30min**

4. 97. Recap & Practical Task: Explore a GraphQL Schema **45min**

Module 17: Module 17: Final Project: Building a Production-Ready API

5 Lessons

1. 98. Project Planning, Data Modeling, and API Design (for Final Project) **45min**

2. 99. Implementing All API Endpoints (CRUD, Authentication, Authorization, File Uploads) **3hr**

3. 100. Integrating Database, Error Handling, and Security Best Practices **1hr 30min**

4. 101. Final Testing, Documentation, and Deployment Preparation **1hr 30min**

5. 102. Recap & Final Project Submission **1hr 30min**

This curriculum is subject to minor adjustments to ensure the most up-to-date and effective learning experience.