

Node.js - Beginner to Intermediate

FA-0498 · 3 Days

Software Development · DevOps, Cloud & Infrastructure · Cybersecurity · Data & Analytics

Introduction

Welcome to the Node.js Course! Node.js is a powerful runtime that allows you to build fast, scalable, and real-time applications using JavaScript on the server side. In this course, you'll embark on a journey from the fundamentals to advanced topics, learning how to create web servers, handle asynchronous operations, interact with databases, secure your applications, and deploy them for production. Whether you're new to Node.js or looking to deepen your expertise, we're here to guide you through the exciting world of server-side JavaScript development.

Course Duration: 3 Days

Prerequisites

- Basic JavaScript
- Basic Node.js Knowledge
- Basic Command Line
- Version Control (Optional)
- Database Basics (Optional)

Learning Outcome:

- Gain a solid understanding of Node.js and its evolution, use cases, and advantages.
- Master asynchronous programming techniques, including callbacks, promises, and async/await.
- Develop web servers and RESTful APIs using Node.js and Express.js.

- Work with databases (e.g., MongoDB, MySQL) and implement CRUD operations.
- Create real-time applications using WebSockets.
- Optimize the performance and scalability of Node.js applications.
- Implement security best practices and address common vulnerabilities.
- Prepare Node.js applications for production deployment.
- Explore advanced topics such as GraphQL, serverless computing, and the latest Node.js features and trends.

Training outline

1. Day 1: Getting Started with Node.js

Introduction to Node.js

2. What is Node.js?

3. History and evolution of Node.js

4. Use cases and advantages

Setting Up Your Development Environment

5. Installing Node.js and npm (Node Package Manager)

6. Choosing a code editor

7. Creating your first Node.js application

Node.js Basics

8. JavaScript fundamentals review (if needed)

9. Writing and running a basic Node.js script

10. Modules and the CommonJS pattern

Asynchronous Programming in Node.js

11. Understanding callbacks

12. Using promises and async/await for asynchronous code

13. Handling errors in asynchronous code

14. Day 2: Building Applications with Node.js

Working with the File System

15. Reading and writing files

16. Working with directories

17. Handling file system events

HTTP and Web Servers with Node.js

18. Creating a simple HTTP server

19. Routing requests

20. Handling POST requests

21. Building a RESTful API with Express.js (a popular Node.js framework)

Database Access with Node.js

22. Connecting to databases (e.g., MongoDB, MySQL)

23. Performing CRUD operations

24. Using an ORM (e.g., Mongoose for MongoDB)

Middleware and Routing

25. Understanding middleware functions

26. Implementing authentication and authorization

27. Handling routing in Express.js

28. Day 3: Advanced Node.js Topics

Real-time Applications with WebSockets

29. Introduction to WebSockets

30. Building a real-time chat application

Scaling and Performance Optimization

31. Strategies for scaling Node.js applications

32. Load balancing and clustering

33. Profiling and optimizing Node.js code

Security Best Practices

34. Common security vulnerabilities in Node.js

35. Implementing security measures

36. Using authentication and authorization libraries

Deployment and DevOps

37. Preparing your Node.js application for production

38. Deployment options (e.g., hosting on Heroku, AWS, or a VPS)

39. Continuous integration and continuous deployment (CI/CD)

Advanced Topics and Trends

40. Introduction to GraphQL

41. Serverless computing with Node.js

42. Exploring the latest Node.js features and trends

© 2026 Faysal Aziz · Kuala Lumpur

<https://faysalaziz.com/courses/node-js-beginner-to-intermediate/>