Node.js

Description

During this 5-day course, participants will learn how to design, test, and deploy a RESTful API with Node.js and MongoDB. The training takes a hands-on approach, focusing on building robust and secure applications tailored to professional environments. Each step is guided to help learners create a complete project, from initial setup to final deployment.

Why take a Node.js training course

The Node.js course covers the core modern technologies used by developers. Participants will explore the role of middleware, how to structure an application, and how to model data with Mongoose. They will also learn how to integrate testing, secure routes using JSON Web Tokens, and manage authentication. Finally, the course concludes with concrete deployment steps to make the project fully operational in a production environment.

Course Content
Module 1: Setup program

- Introduction
- Installing Node
- What is Node?
- Why should I use Node?

Module 2: Node.js basics

- Using Require
- Requiring your own files
- Using third-party modules
- Restarting the app with Nodemon
- Getting user input
- Simplified input with Yargs
- Working with JSON
- Refactoring for reusability
- Reading notes and reusability
- Debugging Node.js apps
- Debugging with Chrome Dev Tools
- Identifying notes
- Requiring arguments and advanced arguments
- Arrow functions

Module 3: Asynchronous Node.js

- Asynchronous basics
- Call stack and event loop
- Callback functions and APIs
- Printing objects
- What makes up an HTTP request?

- Encoding user input
- Callback errors
- Callback summary
- Wiring up the weather
- Chaining callbacks together
- Intro to ES6 Promises
- Advanced promises
- Additional features

Module 4: Web servers and app deployment

- Creating a web server
- Rendering templates with data
- Advanced templates
- Express middleware
- Adding version control (Git)
- Setting up GitHub and SSH keys
- Deploying your apps
- Adding a new feature and deploying

Module 5: Testing your applications

- Mocha and basic tests
- Watch mode and auto-restart testing
- Using an assertion library
- Testing asynchronous code
- Testing Express apps: Part I
- Testing Express apps: Part II
- Organizing tests with describe()

Module 6: MongoDB, Mongoose and REST APIs

- 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
- Mongoose queries and ID validation
- Deploying the API to Heroku

Module 7: Security and authentication

- Setting up the user model
- JWT and hashing

- Generating auth tokens and setting headers
- Private routes and auth middleware
- Password hashing
- Seeding the test database with users
- Improving app configuration

Module 8: Real-time web apps with Socket.io

- Creating a new project
- Adding Socket.io to an app
- Emitting and listening to events
- Broadcasting events
- Message generator and tests
- Event acknowledgments
- Message form and jQuery
- Geolocation Part I
- Geolocation Part II
- Timestamps and formatting with Moment
- Printing message timestamps
- Mustache.js
- Autoscrolling
- Adding a page
- Passing data
- Socket.io Rooms
- Storing users with ES6 classes: Part I
- Storing users with ES6 classes: Part II
- User list
- Sending messages only to a Room
- New feature ideas

Module 9: Async / Await

- Async / Await project setup
- Async / Await basics
- A real-world example
- Error handling and awaiting async functions
- 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

- Web developers looking to expand their skills
- Aspiring freelancers in Node.js development
- System administrators wanting to automate services

- Students or career changers in web programming

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

Description
Node.js Training
Niveau
Fondamental
Classroom Registration Price (CHF)
3800
Virtual Classroom Registration Price (CHF)
3550
Duration (in Days)
5
Reference
NODEJS