Node.js Guide

From Beginner to Intermediate

Table of Contents

1 Introduction to Node.js

Node.js is a JavaScriptruntime built on Chrome's V8 JavaScriptengine. It allows you to run JavaScripton the server-side, enabling the development of scalable network applications.

- What is Node.js
- History and use cases
- Node.js vs traditional server-side languages

2. Installing Node and npm

Node.js can be installed on various operating systems. npm is the package manager for Node.js, and npx is a package runner.

- Installation steps (Windows, Mac, Linux)
- Understanding npm & npx

3. Core Concepts of Node.js

Node.js is known for its event-driven architecture and non-blocking I/O, making it efficient for I/O-heavy operations.

- Event-driven architecture
- Non-blocking I/O
- Single-threaded vs multi-threaded

4. Node.js Modules

Modules are a fundamental part of Node.js, allowing you to organize your code into reusable components.

- Built-in modules (fs, http, path, etc.)
- Creating custom modules
- Export/import syntax

5. npm (Node Package Manager)

npm is the default package manager for Node.js, allowing you to install and manage packages.

- Installing packages
- package.json explained
- Global vs local packages

6. Asynchronous Programming

Asynchronous programming is a key feature of Node.js, enabling efficient handling of I/O operations.

- Callbacks
- Promises
- Async/Await

7. File System Operations (fs module)

The fs module provides an API for interacting with the file system in a manner closely modeled around standard POSIXfunctions.

- Reading/writing files
- Creating/deleting folders
- Streams

8. Building HTTP Servers

Node.js provides the http module to create HTTPservers and handle requests and responses.

- Using http module
- Creating GET/POSTroutes
- Handling request & response

9. Expressjs Framework

Expressis a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

- What is Express
- Setting up routes
- Middlewares
- Error handling

10. Templating Engines (EJS/ Handlebars)

Templating engines allow you to create HTML pages with dynamic content using templates.

- Rendering dynamic content
- Passingdata to views

11 Working with Databases

Node.js can interact with various databases, both SQLand NoSQL, to perform CRUDoperations.

- MongoDB with Mongoose
- CRUDoperations

12. Authentication

Authentication is crucial for securing your applications. Node.js supports various authentication methods.

- JWTbasics
- Sessions& cookies

13. Real-time Apps with Socket.IO

Socket.IO enables real-time, bidirectional and event-based communication between the browser and the server.

- WebSocketintroduction
- Basicchat application

14. Environment Variables & Security

Managing environment variables and securing your Node.js applications is essential for production environments.

- dotenv usage
- CORS, Helmet, Rate Limiting

15. Deploying Node.js Apps

Deploying Node.js applications can be done on various platforms, each with its own set of tools and configurations.

- Deployment on Render, Vercel, Railway

16. Mini Projects

Building mini projects is a great way to apply your Node.js knowledge and gain practical experience.

- To-Do app (with Express& MongoDB)
- RESTAPI (with JWT)

17. Resources

Here are some resources to help you continue learning Node.js:

- MDN
- Node.js Docs
- YouTube
- GitHub repos