ScholarHat

Swipe

Node.js ROADMAP FOR BEGINNERS

2024 EDITION

LEVEL UP YOUR CAREER

Who is a Node.js Developer?

- Builds **applications** using Node.js, JavaScript's runtime environment.
- Crafts **servers**, handles **data requests**, and interacts with databases.
- JavaScript pro with an understanding of **event-driven** architecture.
- Leverages **npm** for code management.
- Builds **real-time features**, **web applications**, **APIs**, and more.

1 Introduction to Node.js

- Node.js is a **JavaScript runtime environment.**
- Operates on the **V8 JavaScript engine**.
- Runs on **Windows**, **Linux**, **Unix**, and **macOS**.
- Executes JavaScript code outside of browsers.
- Allows construction of command-line utilities.
- Enables **server-side scripting**.
- Promotes JavaScript for various applications.
- **Install Node.js:** Go to the **official Node.js website** and download the latest LTS version for your operating system. Follow the installation instructions.
- **Hello World:** Write a simple "Hello World" program in Node.js to get started.

2 Why Learn Node.js?

- **JavaScript Everywhere:** Use JavaScript for both front-end and back-end development.
- **Fast & Scalable:** Node.js excels at building real-time applications with its event-driven architecture.
- **Rich Ecosystem:** npm offers a vast library of pre-built modules for various functionalities.
- **In-Demand Skill:** Node.js is a sought-after skill in the web development industry.
- **Full-Stack Potential:** Learning Node.js complements front-end skills as well.
- **Following Organizations uses Angular:**

PayPal	LinkedIn	tcs
Infosys	IBM	Walmart
ebay	wipro	Uber

Swipe

3 JavaScript Fundamentals

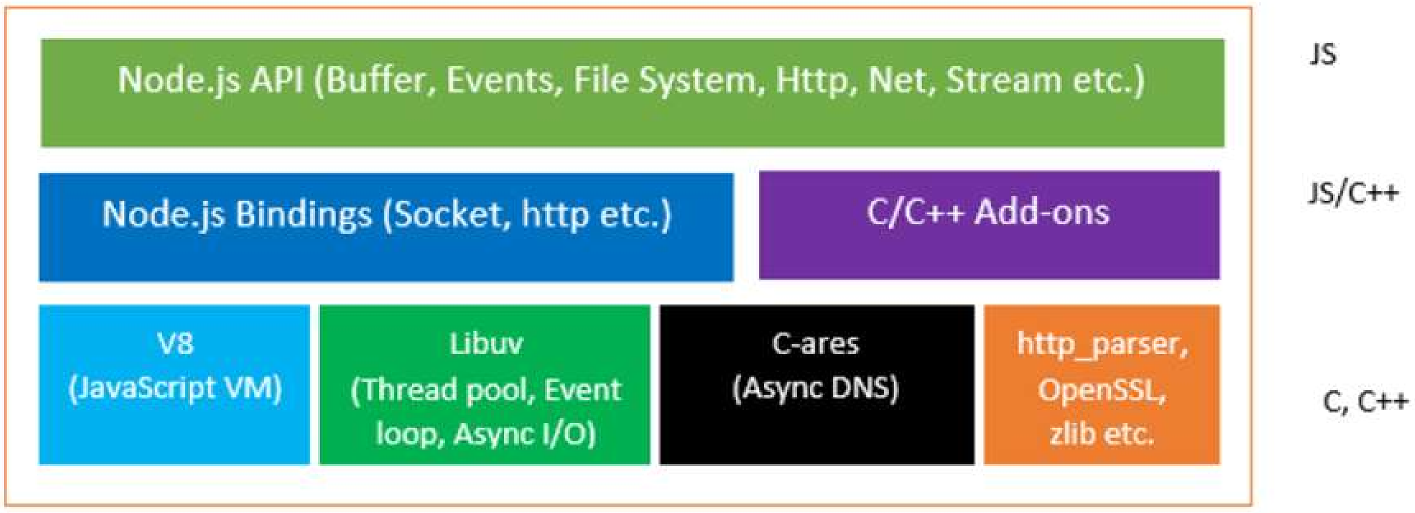
Before diving into Node.js, ensure you have a solid understanding of JavaScript, as Node.js is a JavaScript runtime.

- Understand how to declare **variables** using var, let, or const.
- Learn about **data types**, including strings, numbers, booleans, arrays, and objects.
- Study **Operators** and **Arrays Manipulations** as well.
- Learn **control flow statements** including if, else, switch, for, while, and do-while loops.
- Master the **JavaScript functions** of declaration, expression, and invocation.
- Learn about **object-oriented programming**.
- Master **Functions**, **Hoisting** and **Prototypes**.

4 Node.js Architecture

Node.js has mainly two types of components – **core components** and **node.js API (modules).**

- **Node.js API**
- **Node.js Binding**
- **C/C++ Add-ons**
- **V8**
- **Libuv**
- **C-ares**
- **http_parser**
- **OpenSSL**
- **Zlib**

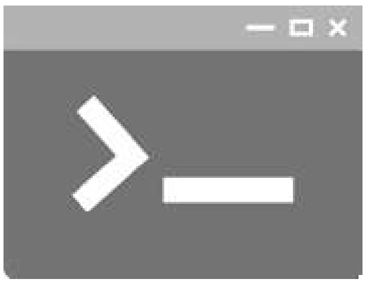

Node.js API (Buffer, Events, File System, Http, Net, Stream etc.)
JS
Node.js Bindings (Socket, http etc.)
C/C++ Add-ons
JS/C++
V8 (JavaScript VM)
Libuv (Thread pool, Event loop, Async I/O)
C-ares (Async DNS)
http_parser, OpenSSL, zlib etc.
C, C++

5 Node.js Development Tools

- Learn how to install **Visual Studio Code (VS Code)** from the website.
- Try installing the **Node.js plugin** in VS Code.
- Master managing **Node.js projects** with Visual Studio Code.
- Learn how **Nodemon** automatically restarts servers when files change.
- Try ***npm install -g nodemon*** for global installation or ***npm install --save-dev nodemon*** for local installation.
- Know setup with nodemon.JSON or command-line parameters.
- Connect Nodemon with **Gulp** or **npm scripts**.

6 Node.js CLI

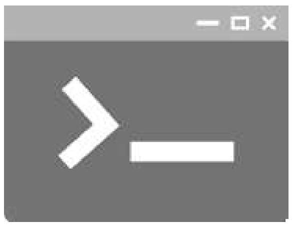
- Learn the **basic npm** and **node commands** for managing packages and scripts.
- Understand how to use **npm init** to configure projects and dependencies.
- Use **npm install** or **npm uninstall** to add or remove dependencies.
- Use npm to differentiate between **global** and **local** package installations.
- Execute project tasks by running scripts from the package.json with npm run.
- Manage **package versions** and dependencies easily with npm obsolete and npm update.



Swipe

7 npm CLI (Node Package Manager)

- Learn about **npm package management**, including installation, updates, and removal.
- Check out **Add Project Dependencies** with Options in npm install.
- To set up the project, create package.json using master ***npm init.***
- Understand the concept of **Semantic Versioning** (SemVer) about package versions.
- Use **npm list** and **npm obsolete** to manage dependencies.
- For script execution in package.json, use master **npm run.**
- Learn how to use **npm publish** to publish packages to the npm registry.

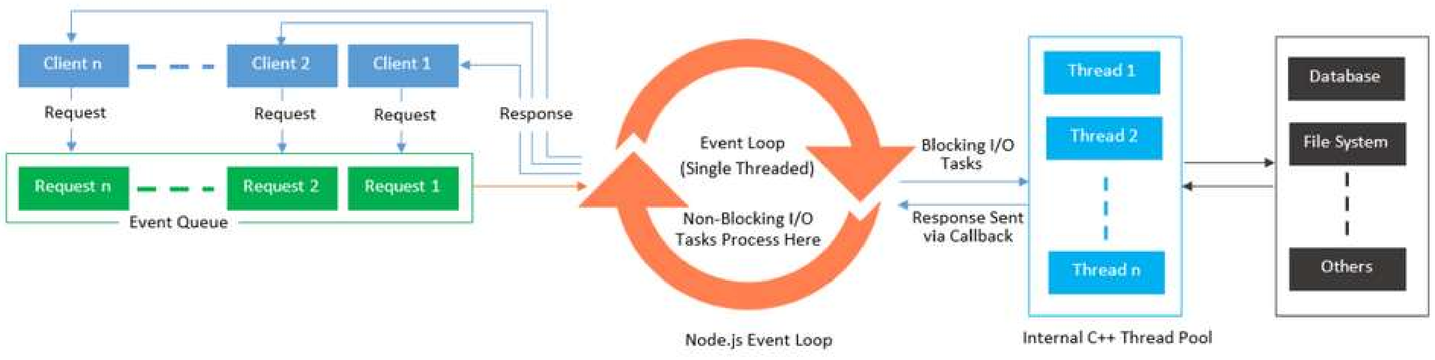


Swipe

8 Node.js Code Execution Process/Event Loop

The Event Loop acts like a traffic controller, managing a queue of incoming events (client requests). It processes events one by one from the queue.

- Learn about **Node.js' event-driven design** and how to handle **asynchronous events**.
- Understand the Node.js' **single-threaded**, **non-blocking I/O** for concurrent tasks.
- Discover Node.js' **modular structure** for code organization and reuse.
- Learn about **error handling** best practices, such as error-first callbacks and try-catch blocks.

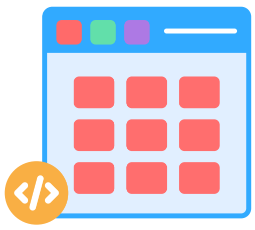


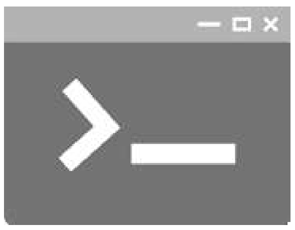
Swipe

9 Node.js Built-in Modules

A collection of JavaScript code which encapsulate related code into single unit of code.

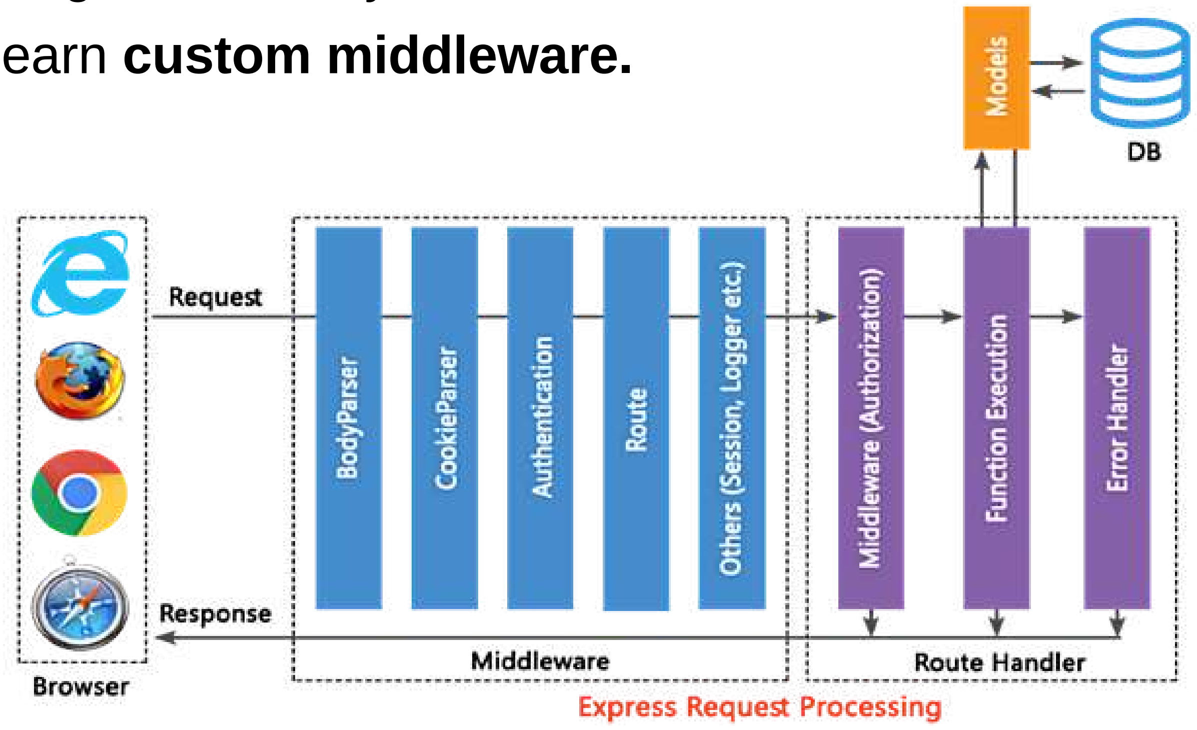
- Discover the concept of **Node.js modules.**
- Learn to create **HTTP** servers and handle HTTP requests and responses.
- Explore **file system** operations such as reading and writing files, creating and deleting directories.
- Understand **path manipulation** operations.
- In **os module**, gain insights into system-related information like CPU architecture.
- Learn about **crypto module** and **cryptographic operations** such as encryption, and decryption.
- Explore utilities for parsing and formatting **URLs**.
- Understand data compression and decompression using the **zlib module**.





10 Web Development with Express.js

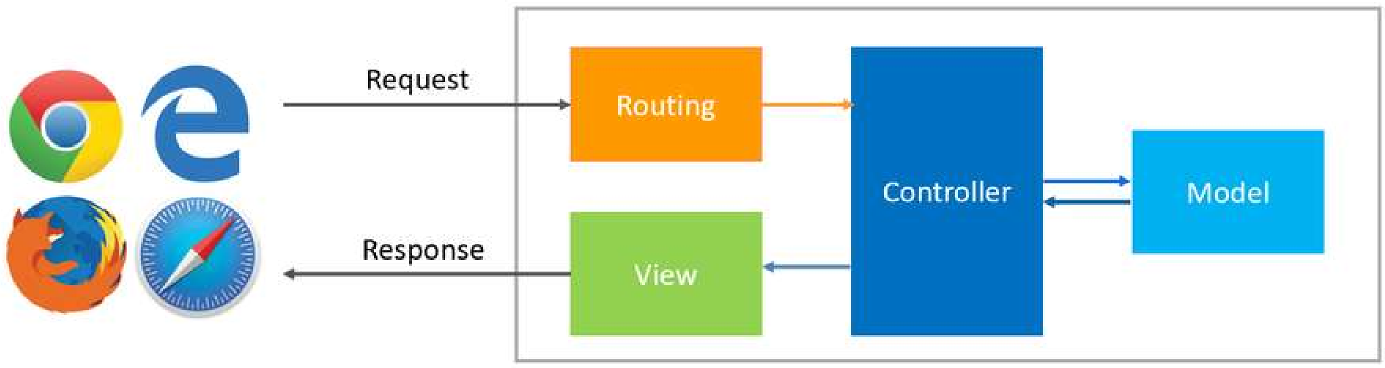
- Explore **features** such as routing, middleware, templates, and error handling.
- Discover how Express.js helps you create **RESTful APIs** and **web apps**.
- Understand the **Node.js middleware** for HTTP request handling.
- Learn to use **middleware for** logging and authentication.
- Discover popular **middleware packages** such as Morgan and Body-Parser.
- Learn **custom middleware.**



Swipe

11 Express Routing and View Engines

- Express.js uses **routing** to handle URL requests efficiently.
- Understand **route** and **query parameters**, as well as **route chaining**.
- Consider **route middleware** for authentication and authorization.
- Practice **organising routes** to improve code maintainability.
- Explore popular **view engines** such as Handlebars, EJS, and Pug.
- Use **template inheritance** and **partials** to optimise view rendering.



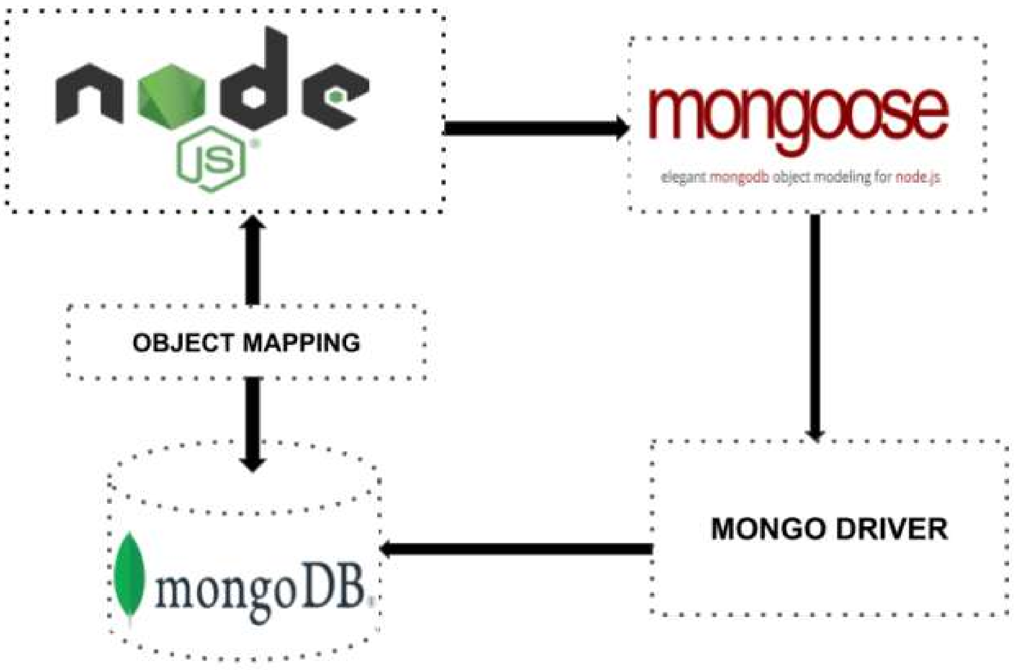
12 MongoDB Database

- Understand the **fundamentals of MongoDB**, including documents, collections, fields, searches, indexes, and aggregation pipelines.
- Know **MongoDB's schema** flexibility, scalability, and the MongoDB query language.
- Differentiate MongoDB from **relational databases**.
- Explore the **principles of NoSQL** databases.
- Learn about MongoDB's **unique features** for efficient data management.

Swipe

13 Mongoose: Models & Relationships Setup

- Get started with **Mongoose**, MongoDB's Node.js ODM.
- Create **MongoDB schemas** using Mongoose definitions.
- Study MongoDB **relationship types**.
- Use Mongoose **features** such as validation, middleware, virtuals, and plugins.
- Learn how to use Mongoose to link a **Node.js program** to a MongoDB database.



14 Node.js App Deployment

- Understand the **Build Automation Tools** for Node.js deployment.
- Learn how to set up **Gulp** or **Webpack** for fast deployment operations.
- Study **Task Automation** using Gulp to streamline deployment procedures.
- Analyse **Module Bundling with Webpack** for improved code delivery.
- Understand deployment concepts and select appropriate platforms such as **Heroku**, **AWS**, or **Microsoft IIS**.

15 Real-time Applications

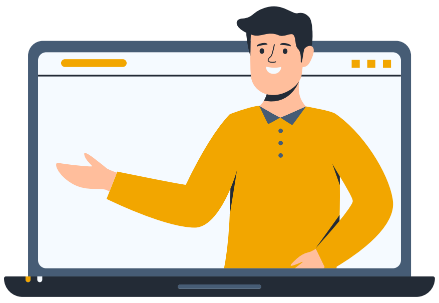
- **Chat Applications:** Real-time chat apps powered by Node.js use Socket.IO for WebSocket connections, allowing clients and servers to communicate instantly.
- **Online Gaming:** With its event-driven architecture, Node.js supports real-time multiplayer games while effectively managing concurrent connections.
- **Live Auctions & Bidding Systems:** Node.js supports real-time bidding systems, which enable users to put bids and receive fast updates during auctions.
- **Streaming Applications:** Node.js is best suited for creating live video and audio streaming platforms, as well as real-time data processing pipelines.

Swipe

Node.js Tutorial For Beginners

ScholarHat offers concise, insightful Node.js articles. Dive into Angular with clear explanations and practical examples, perfect for enhancing your skills.

- What is Node.js and Why to use it?
- Exploring Node.js Architecture
- Node.js vs. Other Server-Side Frameworks
- Exploring Node.js Code Execution Process
- Node.js Core Modules
- Getting Started with Express.js
- Express.js Routing
- A Guide to build Real Time Application in Node.js
- Top 50 Node.js Interview Questions and Answers



How to follow this roadmap?

At ScholarHat, we believe **mastering a technology** is a **three-step process** as mentioned below:

- **Step1 - Learn Skills:** You can learn Azure Developer skills by using **Microsoft official docs** on Node.js, or **through Videos** on YouTube or **Videos based courses**. For topic revision and recalling make **short notes**. You can also learn **Live from Microsoft MVP** at ScholarHat.
- **Step2 - Build Experience:** You can build hands-on experience by **building end-to-end real world applications** like Chat Application, Stock Price Monitoring, Fraud Detection etc.
- **Step3 - Empower Yourself:** Build your **strong profile** by mentioning all the above skills with **hands-on experience** on projects. Prepare yourself with interview **Q&A about Azure Developer** to crack your next job interview.

Did You Find it Useful?

Alamin CodePapa

@CodePapa360

Follow for more

Like Comment Repost