NODE js

Audience

The audience for Node.js is mainly developers (JavaScript, backend, full-stack) working on real-time, scalable web applications and startups or small businesses looking to build fast, lightweight, and scalable services.

Pre-requisites

To get started with Node.js, you should have:

1. A solid understanding of JavaScript.
2. Basic web development knowledge (HTML, CSS, HTTP).
3. Familiarity with CLI commands and npm.
4. Understanding of asynchronous programming.
5. Optionally, knowledge of Git, databases, and frameworks like Express.js.

About Node.js

Node.js is a JavaScript runtime built on Chrome's V8 engine that allows you to run JavaScript code outside the browser

Executing Node.js

```
node <filename.js>
```

Features of Node.js

Asynchronous and Event-Driven: Handles multiple operations concurrently without blocking the execution.

Single-Threaded: Uses a single thread to handle requests through event looping .

Fast and Scalable: Built on Chrome's V8 engine, it provides high performance and scalability.

Cross-Platform: Works across various operating systems like Windows, Linux, and macOS.

NPM (Node Package Manager): Access to a large library of open-source packages for rapid development.

Built-in Libraries: Provides essential modules like http, fs, and path for server-side development.

Environment Setup

Node.js Environment Setup

Follow these steps to set up Node.js on your system:

1. Download & Install Node.js

- Go to Node.js Official Website
- Download LTS (Long-Term Support) version
- Install it (includes npm automatically)

2. Verify Installation

Check if Node.js and npm are installed:

sh
CopyEdit
node -v   # Check Node.js version
npm -v   # Check npm version

Package Manager (NPM)

NPM is the default package manager for Node.js. It helps in installing, managing, and sharing JavaScript packages (libraries).

Global vs Local Installation (NPM)

- Local Installation (npm install package-name) → Installs the package inside the project folder (node_modules), accessible only within that project.
- Global Installation (npm install -g package-name) → Installs the package system-wide, making it available for all projects and command-line use.

Using package.json

package.json is a configuration file in a Node.js project that stores important project details such as metadata, dependencies, scripts, and configurations. It helps in managing and automating tasks efficiently.

Attributes of package.json

Understanding Attributes of `package.json` (With One-Line Description)

1. `name` – The project name (must be lowercase).
2. `version` – The project version (follows semantic versioning).
3. `description` – A short description of the project.
4. `main` – The entry point file (default: `index.js`).
5. `scripts` – Custom commands to run the project (e.g., `npm start`).
6. `dependencies` – Packages required for production.
7. `devDependencies` – Packages needed only for development.
8. `keywords` – Keywords related to the project for searchability.
9. `author` – The creator of the project.
10. `license` – Specifies the licensing type (e.g., MIT).
11. `engines` – Specifies required Node.js version.

📌 **Purpose:** `package.json` helps manage dependencies, scripts, and project metadata in a structured way! 🚀

Uninstalling Modules

```
npm uninstall  < package-name >
```

Updating Modules in Node.js (NPM)

npm update package-name