

Introduction to Node.js

Introduction to Node.js – Study Notes

1. What Is Node.js?

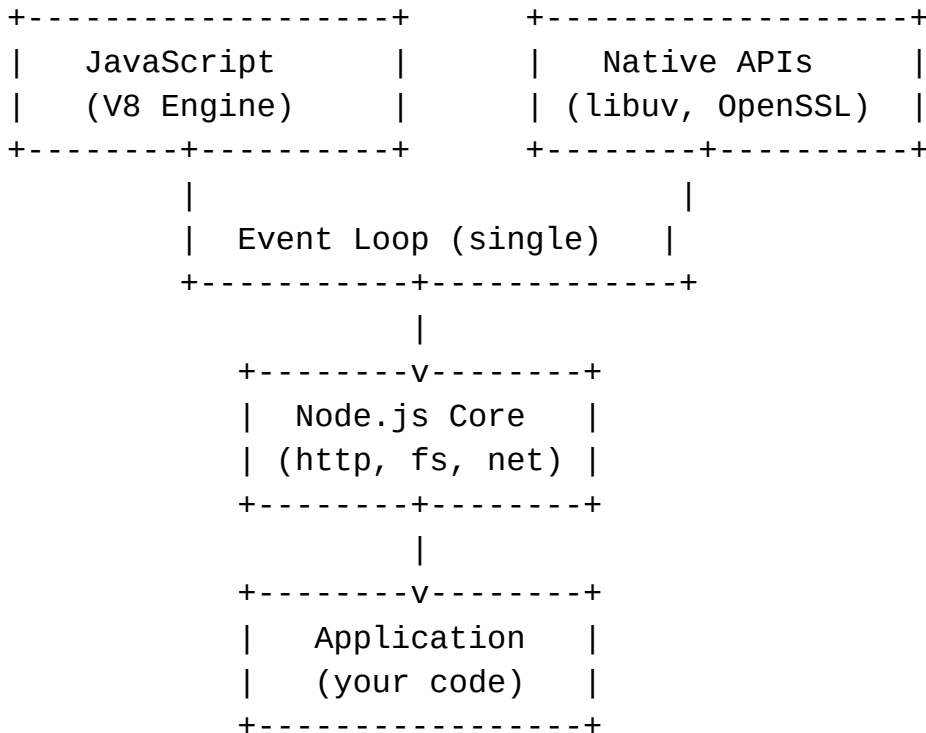
Aspect	Description
Definition	A JavaScript runtime built on Chrome's V8 engine that allows you to run JavaScript code outside a web browser.
Primary Goal	Enable server-side (back-end) development with JavaScript, providing a unified language for both client and server.
Release	First released in 2009 by Ryan Dahl; now maintained by the OpenJS Foundation.
Package Manager	npm (Node Package Manager) – the default repository for reusable modules.

2. Core Concepts

Concept	Key Points
Event-Driven Architecture	Uses an event loop to handle asynchronous operations without blocking the thread.
Non-Blocking I/O	I/O tasks (file system, network, DB) are delegated to the OS; Node continues processing other events.
Single-Threaded Model	JavaScript runs on one thread, but heavy work can be off-loaded to the libuv thread-pool or worker threads.
Modules	Built-in CommonJS (<code>require</code> , <code>module.exports</code>) and ES Modules (<code>import</code> , <code>export</code>).
Callback ? Promise ? async/await	Evolution of handling async code: callbacks ? Promise objects ? <code>async/await</code> syntax for readable flow.

3. Architecture Overview

Introduction to Node.js



- **V8** – Executes JavaScript.
- **libuv** – Provides the event loop, thread pool, and cross-platform abstractions for I/O.
- **Core modules** – Built-in libraries (http, fs, crypto, etc.).
- **User code** – Your application logic, built on top of the core.

4. Installing Node.js

Platform	Steps
Windows	Download the MSI installer from https://nodejs.org , run it, and verify with <code>node -v</code> .
macOS	Use Homebrew: <code>brew install node</code> or download the pkg installer.
Linux	Use the package manager (e.g., <code>apt-get install nodejs</code>), or install via nvm (Node Version Manager) for flexible version handling.
Version Managers	<code>nvm</code> , <code>nvs</code> , <code>fnm</code> – allow you to switch between multiple Node versions easily.

5. Creating a Simple Server

Introduction to Node.js

```
// server.js
const http = require('http');           // Core module

const PORT = 3000;

const requestHandler = (req, res) => {
  res.writeHead(200, { 'Content-Type': 'text/plain' });
  res.end('Hello from Node.js!\n');
};

const server = http.createServer(requestHandler);

server.listen(PORT, () => {
  console.log(`Server running at http://localhost:${PORT}`);
});
`;
```