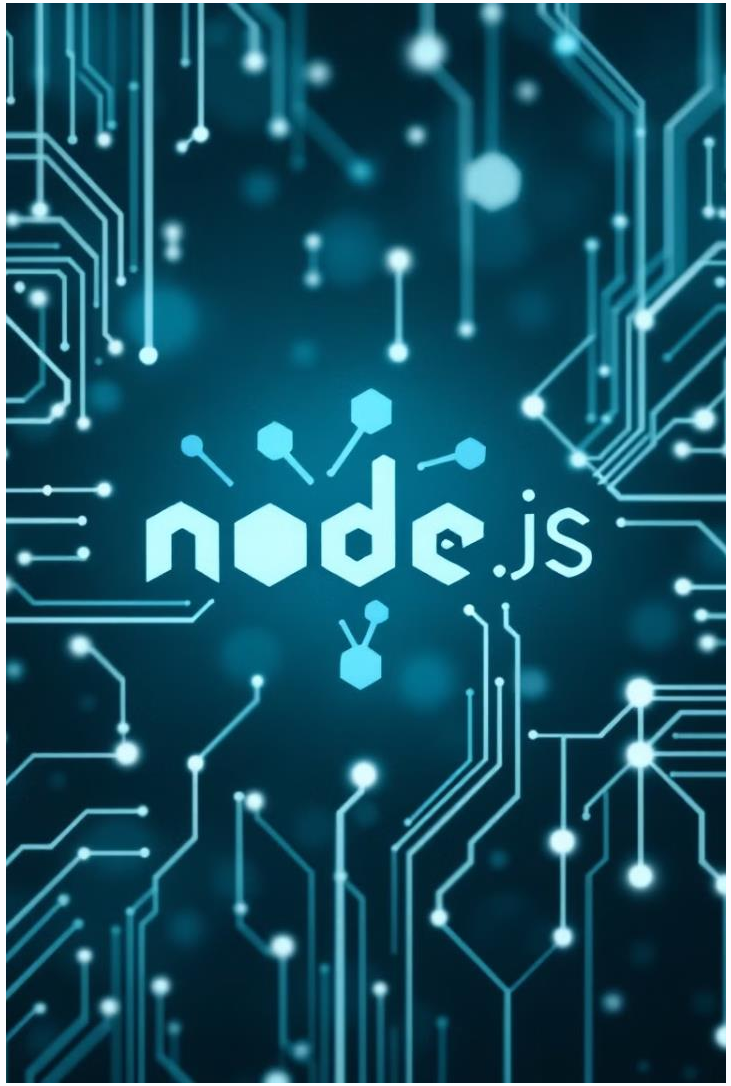


Node.js: A Powerful JavaScript Runtime

Node.js is a powerful JavaScript runtime that allows developers to build scalable network applications using an event-driven, non-blocking I/O model. It enables developers to create server-side applications using JavaScript, a language traditionally used for client-side scripting.

경한 작성자: 경한 김

Unique Features of Node.js

Asynchronous Model

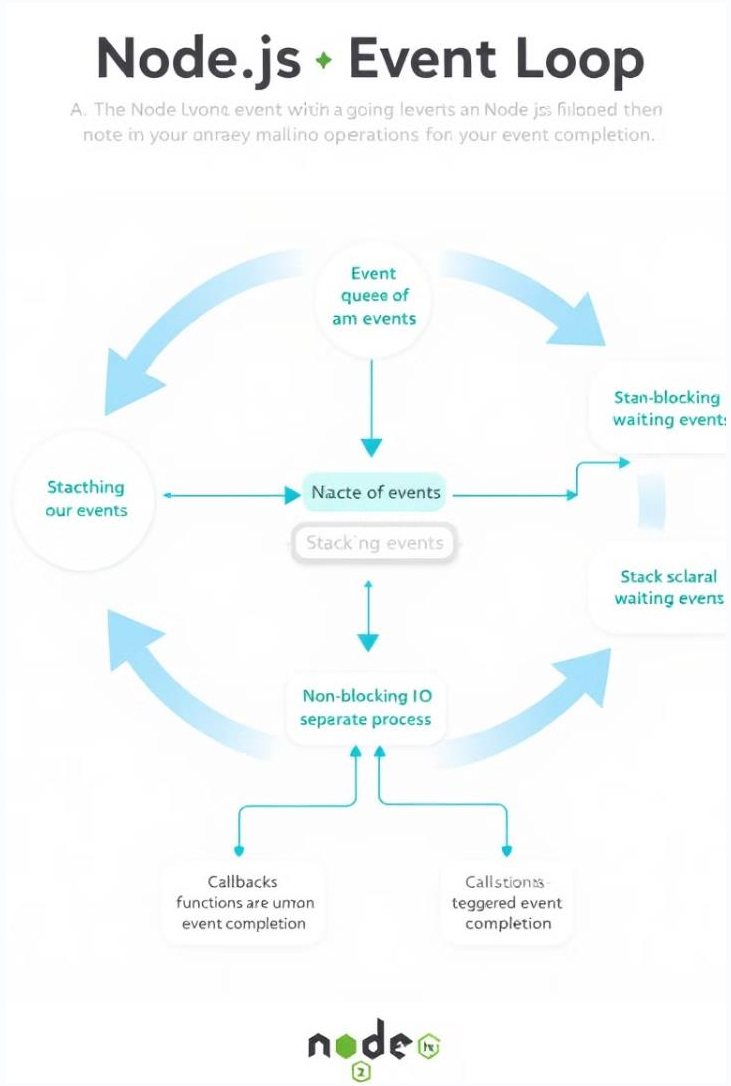
Node.js uses an asynchronous, event-driven model, which allows it to handle a large number of concurrent connections without incurring the overhead of creating new threads.

Single-Threaded

Node.js runs in a single process, using non-blocking I/O calls to support concurrency, making it lightweight and efficient.

Cross-Platform

Node.js is cross-platform, allowing developers to build applications that run on Windows, macOS, and Linux.



Node.js Asynchronous Programming Model

1

Event Loop

Node.js uses an event loop to manage asynchronous tasks, continuously checking for new events and executing their associated callbacks.

2

Non-Blocking I/O

Node.js uses a non-blocking I/O model, allowing it to handle multiple concurrent connections without the overhead of creating new threads.

3

Callbacks and Promises

Node.js encourages the use of callbacks and promises to handle asynchronous operations, making it easier to manage complex, nested asynchronous code.

Event-Driven Architecture in Node.js

Event-Driven

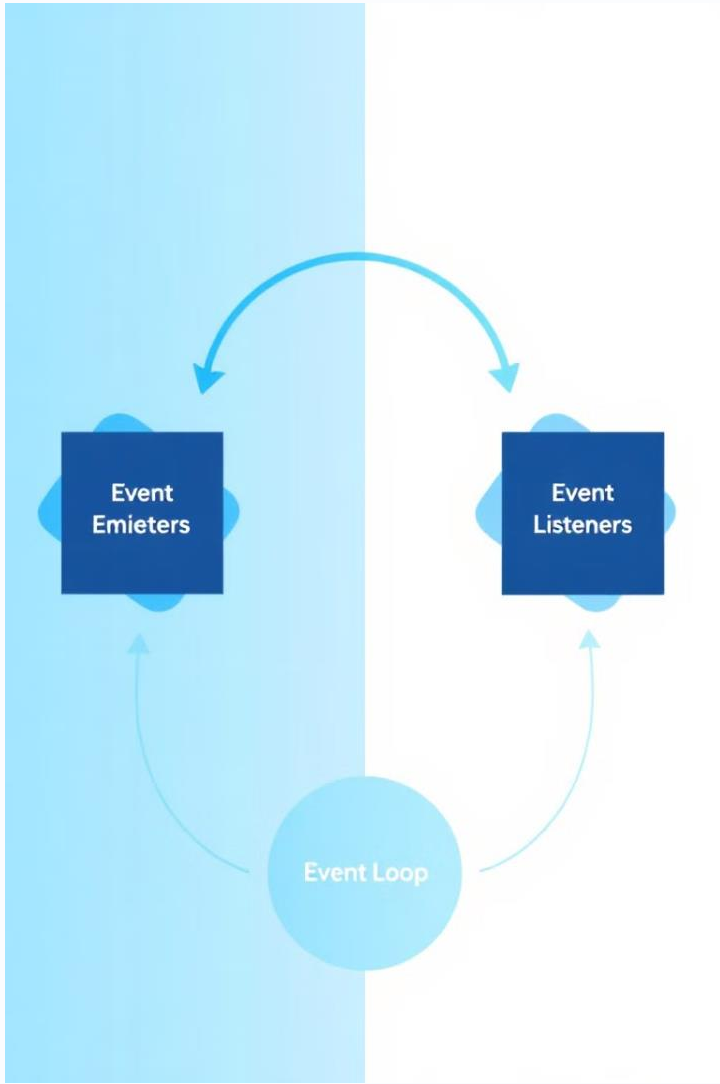
Node.js is built on an event-driven architecture, allowing it to respond to specific events and triggers efficiently.

Scalability

The event-driven model enables Node.js to scale well, handling a large number of concurrent conne ctions without performance degradation.

Flexibility

The event-driven nature of Node.js makes it highly flexible, allowing developers to build a wide range of applications.



Core Modules in Node.js

File System (fs)

The fs module provides a way to interact with the file system, allowing developers to read, write, and manage files and directories.

HTTP/HTTPS

The http and https modules enable developers to create HTTP and HTTPS servers, as well as make requests to other servers.

Path

The path module provides utilities for working with file paths, including joining, normalizing, and resolving paths.

Crypto

The crypto module provides cryptographic functionality, including hash, HMAC, cipher, decipher, and random number generation.

Node.js Package Management with npm

1

npm Registry

The npm registry is the world's largest software registry, hosting ov er 1 million open-source packages.

2

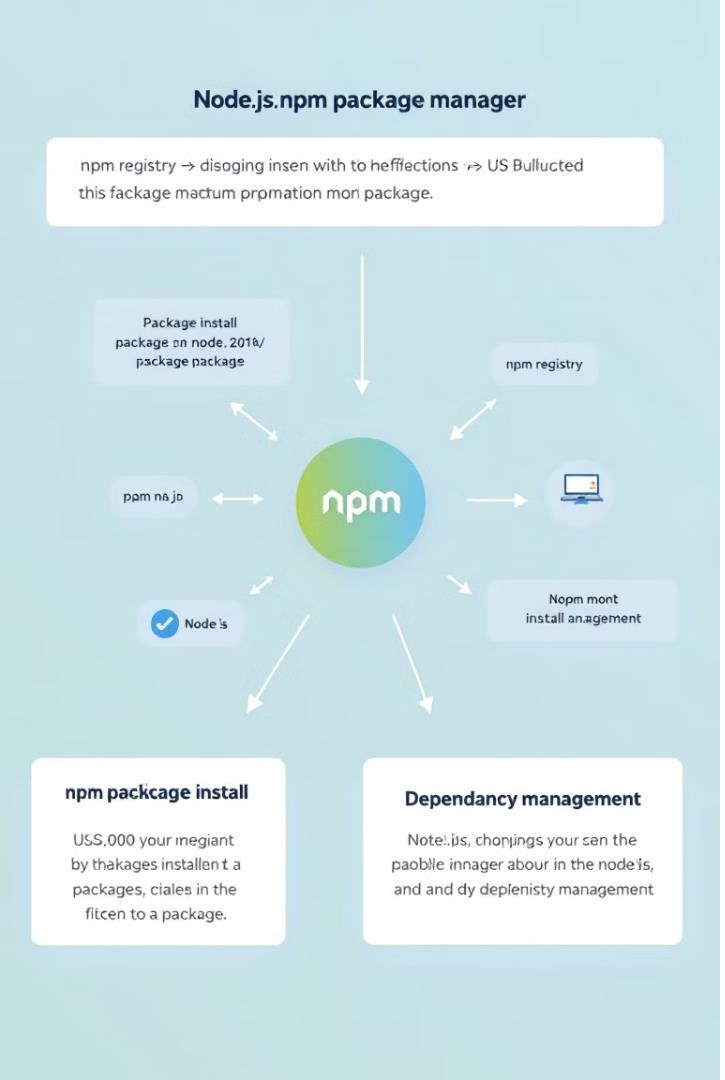
Package Installation

Developers can easily install and manage dependencies using the npm command-line tool.

3

Dependency Management

The npm package manager helps developers track and manage th eir project's dependencies, ensuring consistent and reliable builds.



Web Frameworks in Node.js

Express.js

Express.js is a popular and widely used web application framework for Node.js, providing a robust set of features for we b and mobile applications.

Koa.js

Koa.js is a lightweight, expressive, and middleware-based web framework that builds on top of Node.js, allowing develo pers to create efficient and scalable we b applications.

Hapi.js

Hapi.js is a powerful and flexible server framework for building scalable web ap plications and services with Node.js, foc used on developer experience and best practices.

Node.js: Real-World Applications

1 Web Servers and APIs

Node.js is widely used for building high-performance web servers and APIs, taking advantage of its asynchronous and event-driven architecture.

2 Real-Time Applications

Node.js is well-suited for building real-time applications, such as chat servers, online games, and real-time collaboration tools.

3 Microservices and Serverless

Node.js is a popular choice for building microservices and serverless architectures, leveraging its lightweight and scalable nature.

4 IoT and Edge Computing

Node.js is increasingly used in the Internet of Things (IoT) and edge computing, where its small footprint and event-driven model are beneficial.