

Node.js에 대해서

Node.js 소개

- 정의 및 역사 — Node.js는 서버 사이드 애플리케이션을 개발하기 위해 Google의 V8 JavaScript 엔진을 기반으로 만든 JavaScript 런타임입니다. 2009년에 Ryan Dahl에 의해 처음 출시되었으며, 서버와 클라이언트 간의 통신을 비동기적으로 처리할 수 있는 기능 덕분에 빠르게 성장하였습니다.
- 주요 특징 — 단일 스레드 이벤트 루프 기반으로 동작하여 높은 동시성을 지원하며, 비동기 I/O 모델 덕분에 효율적인 데이터 처리를 가능하게 합니다. 이를 통해 개발자는 복잡한 멀티스레딩에 대한 걱정 없이 높은 성능의 애플리케이션을 작성할 수 있습니다.

주요 기능

- 비동기 I/O — Node.js는 비동기 프로그래밍 모델을 채택하여 입력 및 출력 작업을 처리합니다. 이 덕분에 blocking 없이 대규모 네트워크 애플리케이션을 효율적으로 구축할 수 있으며, 많은 클라이언트 요청을 동시 처리하는 데 유리합니다.
- 이벤트 기반 아키텍처 — 모든 I/O 작업이 이벤트 중심으로 작동하며, 특정 이벤트가 발생했을 때 콜백 함수가 호출됩니다. 이를 통해 이벤트 주도형 프로그래밍이 가능해지고, 기능 확장이 용이해집니다.
- NPM (Node Package Manager) — NPM은 Node.js의 공식 패키지 관리자이며, 다양한 외부 라이브러리와 패키지를 설치하고 관리할 수 있는 기능을 제공합니다. 이를 통해 개발자는 필요에 따라 다양한 모듈을 쉽게 추가하여 애플리케이션을 확장할 수 있습니다.

활용 분야

- 웹 서버 개발 — Node.js는 고성능 웹 서버를 쉽게 구축할 수 있도록 해줍니다. Express.js와 같은 프레임워크와 함께 사용하면 RESTful API와 같은 웹 서비스도 효율적으로 개발할 수 있습니다.
- 실시간 애플리케이션 — 채팅 애플리케이션, 실시간 데이터 스트리밍 등에 적합하며, WebSocket을 활용하여 클라이언트와 서버 간의 실시간 양방향 통신이 가능합니다.
- 마이크로서비스 아키텍처 — Node.js는 경량화된 애플리케이션을 구축하는 데 적합하며, 마이크로서비스 아키텍처를 통해 각 서비스가 독립적으로 배포되고 확장될 수 있도록 지원합니다.

Node.js 설치 및 환경 설정

- 설치 방법 — Node.js 공식 웹사이트에서 제공하는 설치 프로그램을 다운로드하여 설치할 수 있습니다. 또한, 패키지 관리자 등을 활용하여 간편하게 설치할 수도 있습니다.
- 환경 변수 설정 — 시스템 환경 변수에 Node.js의 설치 경로를 추가하여, 명령 프롬프트 또는 터미널에서 Node.js를 쉽게 실행할 수 있도록 설정해야 합니다.

주요 모듈

- HTTP 모듈 — Node.js의 기본 HTTP 모듈을 사용하여 간단한 웹 서버를 구현할 수 있으며, 클라이언트 요청을 처리하고 응답을 전송하는 기능을 제공합니다.
- File System 모듈 — 파일 시스템에 대한 접근 및 조작을 지원하는 모듈로, 파일 읽기, 쓰기, 삭제 등 다양한 파일 관련 작업을 수행할 수 있습니다.
- Express 프레임워크 — Express는 Node.js를 위한 경량화된 웹 애플리케이션 프레임워크로, 라우팅 기능과 미들웨어를 통해 유연하고 강력한 웹 애플리케이션을 구축할 수 있습니다.

장단점

- 장점
 - 높은 성능 및 확장성 — 비동기 I/O 처리 덕분에 고성능의 네트워크 애플리케이션을 개발할 수 있으며, 대규모 트래픽을 효과적으로 처리할 수 있습니다.
 - 활발한 커뮤니티 지원 — Node.js는 활발한 개발자 커뮤니티와 다양한 패키지를 제공하며, 문제 해결이나 기능 추가에 있어 많은 지원을 받을 수 있습니다.
- 단점
 - CPU 집약적인 작업에는 부적합 — Node.js는 단일 스레드로 동작하기 때문에 CPU 집약적인 작업을 처리할 때 성능 저하가 발생할 수 있습니다.
 - 콜백 지옥 문제 — 비동기 처리를 위해 콜백 함수를 많이 사용하게 되면, 코드가 복잡해지고 가독성이 떨어지는 '콜백 지옥' 현상이 발생할 수 있습니다.