

Node.js에서 Express.js 소개

Node.js는 JavaScript 런타임 환경으로, 서버 측 애플리케이션 개발에 널리 사용되는 기술입니다. 이 Node.js 기반에서 Express.js는 가장 유명하고 널리 사용되는 웹 애플리케이션 프레임워크입니다. Express.js는 간단하고 유연한 API를 제공하여 강력한 웹 애플리케이션을 빠르게 구축할 수 있도록 합니다.

 작성자: 우끼끼



Express.js의 장단점 살펴보기

장점

- 빠른 프로토타입 구축
- 유연한 라우팅 기능
- 미들웨어 지원으로 확장성 있는 애플리케이션 개발이 가능

단점

- 복잡한 애플리케이션에서 관리가 어려울 수 있음
- 대규모 프로젝트에서는 모놀리식 구조로 인한 한계가 있음

실무 적용

프로젝트의 요구사항과 개발 팀의 역량에 따라 Express.js가 적합할 수 있습니다.




Express.js의 주요 기능 및 특징

- 경량 프레임워크로 간단한 구조와 유연한 설계를 자랑합니다.
- 웹 애플리케이션 개발 시 필수적인 기능들을 제공하며, 미들웨어 시스템을 통해 개발자가 원하는 기능을 쉽게 확장할 수 있습니다.

Express.js 어플리케이션 구조 이해하기



Express.js 어플리케이션은 라우팅, 미들웨어, 템플릿 엔진 등으로 구성된 모듈화된 구조를 가집니다. 모듈화된 아키텍처를 통해 개발자는 유지 보수성이 높고 확장 가능한 웹 애플리케이션을 구축할 수 있습니다.



Express.js를 이용한 라우팅 처리 방법

1

URL 기반의 선언적 라우팅

Express.js의 라우팅 기능을 통해 클라이언트의 요청 URL에 따라 적절한 처리 로직을 적용할 수 있습니다.

2

쉬운 라우팅 설정

라우팅 테이블 기반의 선언적 라우팅 설정으로 URL 경로와 핸들러 함수를 쉽게 매핑할 수 있습니다.

Express.js와 middleware 활용하기



1

확장성 있는 기능 개발

Express.js의 미들웨어 기능을 통해 요청 전처리, 로깅, 에러 핸들링 등 다양한 기능을 쉽게 확장할 수 있습니다.

2

사용자 경험 제공

미들웨어 함수는 요청과 응답 객체에 접근할 수 있어 개발자가 원하는 사용자 경험을 제공하는 데 활용할 수 있습니다.

Express.js를 이용한 기업 웹사이트 구축

- 중소기업 X사는 Express.js를 활용하여 반응형 웹사이트를 빠르게 구축하였고, 손쉬운

