

Node.js로 구현하는 실시간 분산 채팅서비스

개발환경 구축 및 명령어 실습

강창훈

지니공공아카데미

Contents

01

Node.js

백엔드 개발환경 구축

1. Node Framework 설치
2. 실습 작업 폴더 만들기
3. VSCode 개발 툴 설치

02

NPM,NODE

주요 명령어 실습하기

1. Node,NPM 소개 및 설치버전 확인
2. NodePackage 와 package.json 이해
3. NPM 주요 명령어 실습

03

NVM

설치 및 명령어 실습

1. NVM 설치하기
2. NVM 주요 명령어 실습

01

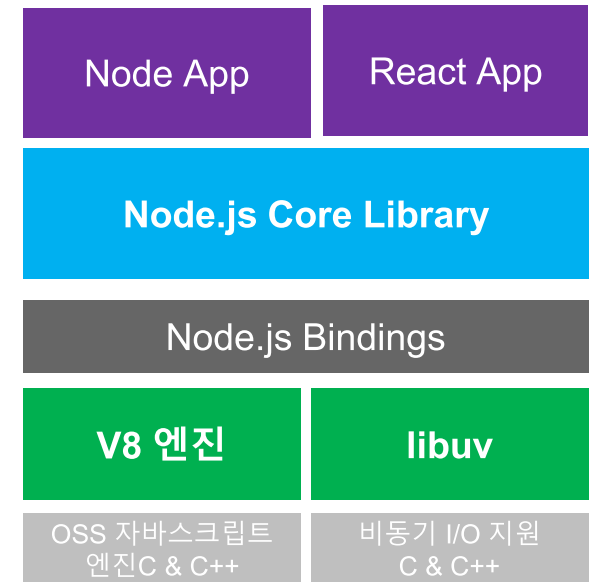
Node.js

백엔드 개발환경 구축

1. Node Framework 설치
2. 실습 작업 폴더 만들기
3. VSCode 개발 툴 설치

1. Node Framework 설치

- <https://nodejs.org/ko/>
- 20.15.1 LTS (2024.07기준) 다운로드 설치 진행
- 오픈소스는 되도록 최신버전의
LTS(Long Term Support) 버전 설치 필수 3년간 안정적 사용환경지원



Node.js Framework

2. 실습 작업 폴더 만들기

- 실습 작업 폴더 만들기
- C:₩Backend₩ or D:₩Backend₩
- macOS는 사용자 폴더내 실습 폴더명만 동일하게 지정해줌

3. VSCode 개발툴 설치

- <https://code.visualstudio.com/>
- **Windows x64 or macOS Stable** 다운로드 설치
- 설치 이후 **한국어 언어 팩** 설치 권장 시 설치
- 최초 설치 후 각종 설정 화면 **나타나면 하단에 Skip** 진행 건너뛰



02

NPM,NODE

주요 명령어 실습하기

1. Node,NPM 소개 및 설치버전 확인
2. NodePackage 와 package.json 이해
3. NPM 주요 명령어 실습

1. Node,NPM 소개 및 설치버전 확인



- Node Framework이 설치되면 **node** 와 **npm** 두개의 SW가 설치됩니다.
- 설치된 node,npm 버전확인하기
- 윈도우 command Prompt 나 VSCode 터미널에서 하기 명령어 실행-문제시 관리자 권한으로 실행할 것

node -v

npm -version or npm -v

1) Node.js

노드는 크롬 v8 자바스크립트 엔진으로 빌드된 자바스크립트 런타임

런타임이란? 특정 개발언어로 만든 프로그램을 실행할 수 있는 환경

노드는 자바스크립트로 개발된 프로그램을 각종 컴퓨터(서버포함)에서 실행할 수 있게 런타임 환경지원

2008년 구글이 V8 자바스크립트 엔진을 이용 크롬 브라우저를 출시,오픈소스화

2009년 V8엔진 기반 노드 프로젝트 시작

2) 노드 주요 특징

이벤트 기반 개발 모델

논 블로킹 I/O 모델 제공

가볍고 효율적인 백엔드 자바스크립트 기반 개발 프레임워크

자바스크립트 오픈소스 **라이브러리 저장소(노드팩키지저장소)** 생태계 제공

NPM(Node Package Manager)을 통한 노드 패키지 저장소 오픈소스 로컬관리

2. Node Package 와 package.json 이해

1) NPM(Node Package Manager)= SW

- Node Package : 특정 기능을 제공하는 자바스크립트/노드 모듈들의 집합체
- Node Package 저장소 : 세계 최대 재사용 가능한 노드(패키지들)와 자바스크립트 오픈소스 저장소
- Npm(Node Package Manager): 각종 Node Package의 설치 및 설치된 패키지들을 관리해주는 SW
- 개발자는 노드 패키지 저장소에서 **NPM명령어**를 이용해 각종 패키지를 개발 또는 서비스 환경으로 다운로드 설치하여 빠르게 오픈소스 패키지 기반 프로젝트 개발 및 서비스가 가능하다.

2) package.json

- 해당 패키지의 정보와 버전 및 설치된 각종 패키지 이력정보를 관리하는 파일
- 노드 프로젝트는 하나의 패키지.json 파일 생성이 필요하다.
- 프로젝트 시작 전 packange.json 파일부터 생성 시작

npm init 실행 후 순차적 정보 입력 후 최종 **yes** 하면 폴더 또는 프로젝트에 **package.json** 파일 생성

3. NPM 주요 명령어 실습

1) 패키지 설치와 일괄 재설치 그리고 패키지 삭제하기

- 신규 패키지 설치하기 : **npm install 패키지명** 또는 **npm i 패키지명** 또는
- 여러 패키지를 동시 설치 : **npm install [패키지명] [패키지명] [패키지명]**
- **package.json** 파일 : dependencies 영역 아래 설치된 패키지 이력(패키지명 과 버전)이 기록저장됨
- **node-modules** 폴더 : 명령어에 의해 설치된 실제 노드 패키지 소스들의 물리적 파일 위치
설치된 패키지가 의존하는 다른 패키지들도 함께 설치됨.
- **노드 패키지 전체 재설치** : **package.json** 파일만 있으면 **npm install** 또는 **npm i** 명령어를 통해
프로젝트에서 사용한 **모든 패키지들을 자동 복원 재설치**가 가능함.
- **package-lock.json** : 설치된 패키지들 간의 의존/종속 관계 정보를 관리해줌
- 설치된 패키지 삭제하기 : **npm uninstall 패키지명**

3. NPM 주요 명령어 실습

2)개발시에만 사용하는 개발환경 지원 패키지 설치

- npm install 패키지명 --save-dev

실제 프로덕션 배포시에는 사용하지 않고 개발시에만 사용하는 개발용 패키지 설치 시 사용

- package.json 파일내 devDependencies 영역내 설치된 개발용 패키지 정보 제공

ex) npm install nodemon --save-dev

3. NPM 주요 명령어 실습

3)개발 컴퓨터 전역공간에 전역 패키지 설치하기

- **npm install -global [패키지명]** or **npm i -g 패키지명**

현재 개발중인 프로젝트 폴더에 패키지들을 설치하지 않고 컴퓨터 전역 공간에 패키지를 설치하여

컴퓨터내 모든 노드 프로젝트에서 사용 및 실행이 가능 함 **c:\users\사용자계정\appdata\roamng\npm** 경로내 설치

전역으로 설치한 패키지는 package.json파일에 기록되지 않음

03

NVM

설치 및 명령어 실습

1. NVM 설치하기
2. NVM 주요 명령어 실습

1. NVM 설치하기

1) NVM(Node Version Manager) 설치

- <https://github.com/coreybutler/nvm-windows/releases>
- nvm-setup.zip 다운로드 압축해제 설치
- 용도는 동일한 컴퓨터에 여러 버전의 Node Framework을 설치하고 관리하고자 할때 사용

2. NVM 주요 명령어 실습

2) NVM 주요 명령어

- NVM버전 확인 : **nvm version**
- 현재 설치된 모든 NODE FRAMEWORK 목록 버전 확인 : **nvm ls**
 - * 16.13.1 (Currently using 64-bit executable) : 현재 사용중인 적용중인 버전은 *가 앞에 표시됨
14.18.3
- 특정 노드 프레임워크 버전 활성화 하기: **nvm use 버전명 ex) nvm use 14.18.3**
- 특정 노드 프레임워크 버전 설치하기 : **nvm install 버전명 ex) nvm install 14.18.3**
- 특정 노드 프레임워크 삭제하기 : **nvm uninstall 버전명**

**** nvm 설치 후 반드시 개발 툴을 닫고 다시 열어주세요.**

**** cmd나 VSCode에서 nvm use 시 에러 나는 경우는**

관리자 권한으로 실행 옵션을 통해 cmd나 VSCode를 닫고 다시 실행해서 명령어를 실행하세요.

감사합니다.

Node.js로 구현하는 실시간 분산 채팅서비스

개발환경 구축 및 명령어 실습

강창훈 | 믹스드코드닷컴

[010-2760-5246](tel:010-2760-5246)
ceo@msoftware.co.kr
<https://mixedcode.com>
<https://jiny.academy>