


## 1) Vue.js : 웹프론트엔드 프레임워크, 컴포넌트 기반의 SPA를 구축할 수 있게 해주는 프레임워크

- 사용성
  - web-app 제작시 유용
  - F/E에서는 SPA 3대장(React, Angular, Vue) 중 가장 쉽고, 간단한 문법
  - 코드짤 때 right way가 있음
  - HTML 렌더링 빠름
  - 업데이트 잘됨
- 비교
  - react : 함수와 자바스크립트에 가까움, HTML, CSS 도 JS로 해버리자,
    - 라이브러리, 초기에는 쉬우나 복잡해지면 실력만큼 구현해야함
  - vue : HTML, CSS, JS 각각의 특성을 살려서 편하게 만들자
    - 프레임워크, 이미 갖춰진 채로 시작
  - 라이브러리 : 코드 컨트롤의 주체가 개발자, 내가 필요할 때 부름, 쉽게 대체가 가능
    - 다만 라이브러리가 프레임워크를 부를 수 있음, 그땐 규칙을 따라야 함
  - 프레임워크 : 프레임워크의 규칙에 따라 개발, 프레임워크가 나에게 규칙을 알려줌
    - django
- 관련 사이트
  - 한국어 : <https://v2.ko.vuejs.org/>
  - 영어 : <https://vuejs.org/>
  - 입문자 가이드 : <https://simplevue.gitbook.io/intro/>
  - 애플코딩 vue 강의 :  Vue 0강 : 카카오가 리액트넵두고 Vue 쓰는 이유
- 컴포넌트(Component)
  - 웹을 구성하는 로고, 메뉴바, 버튼, 모달창 등 웹 페이지 내의 다양한 UI 요소
  - 재사용 가능하도록 구조화 한 것
- SPA(Single Page Application)
  - 하나의 페이지 안에서 필요한 영역 부분만 로딩 되는 형태
  - 빠른 페이지 변환, 적은 트래픽 양

## 강의 요약) 웹서버가 죽지 않게 하기 위해(역할 분산)

- 백엔드 서버의 존재 이유
  - 다중, 복합 처리가 필요할 경우
  - spring과 spring boot 가 계산하는일이 더빠르게 진행되도록?
- spa로 db접속 가능한데 쓰는 이유
  - spa는 브라우저단에서 사용자의 컴퓨터 자원을 사용, 단일쓰레드로 한번에 하나씩만 처리
  - 현재 각 회사들이 webserver 에서 처리하기 위해 비용증가, was나 db 처리하지 않고
- 자료처리를 브라우저단에서 전부 해결 못함
  - 적은 규모는 상관없지만 규모가 커지면 문제 발생

## 2) Node : 노드를 설치하여 사용해야 Vue 의 간략히 실행 환경 세팅

- **CMD 에서의 설치 과정**

### 2-1) Node 미설치시 설치

- nvm 설치여부 확인 : CMD > nvm 입력
  - nvm 다운 로드 : [nvm-setup.exe](https://github.com/coreybutler/nvm-windows#installation--upgrades) 실행 (node를 까는 것)
    - <https://github.com/coreybutler/nvm-windows#installation--upgrades>
    - <https://github.com/coreybutler/nvm-windows/releases>
  - 참고로 <https://nodejs.org/en/download> 여기서 다운시 윈도우에 직접 설치하므로 차후 버전관리 힘들다고 함

### 2-2) Node 설치 완료

- 확인 : CMD > nvm (뭔가 주룩 뚝)
- 버전 확인 : nvm -- version
  - 1.1.11 : 이걸 노드의 버전을 확인하는 것
  -

### 2-3) NVM(Node version Manager) 설치 및 사용

- LTS 최신버전 설치 : nvm install --lts
  - LTS 중 하나 설치 : nvm install 18.14.1
  - nvm 을 깔아 노드 버전 관리 프로그램을 설치하는 것
- 설치된 목록 확인 : nvm ls
  - 20.10.0
- node 사용 : nvm use <version>
  - nvm use 20.10.0
- 사용중인 버전 확인 : nvm current
  - v20.10.0

### 2-3) NVM 관련 설명

- node.js() : 자바스크립트 기반, JS로 만들어진 프로그램의 실행 환경 제공(런타임)
- 이벤트 처리와 비동기식 모델 작업
- 실시간 처리
- 입출력이 자주 발생하는 경우
- 원래는 서버에 설치해야 하나 학습상 우리 컴퓨터에 설치
- JS 만들어진 파일을 웹이 아닌 서버단에서 실행 가능
  - node를 기반으로 react 와 vue 가 실행되면 JS 기반
  - node(운영체제) 깔아야 vue(app) 가 된다고 보면 됨

## 3) Visual Studio Code : [User Installer x64](#)

- terminal 통해서 VS Code 에서도 설치가능하나 생략함

#### 4) NPM(node package manager) : node.js 가 실행되는 환경을 만들기 위한 프로젝트 생성

- 자바스크립트 패키지 매니저
- 외부라이브러리, 도구
- package.json 파일 생성을 통해 의존성(라이브러리)
- 자료처리를 위한 로직구조

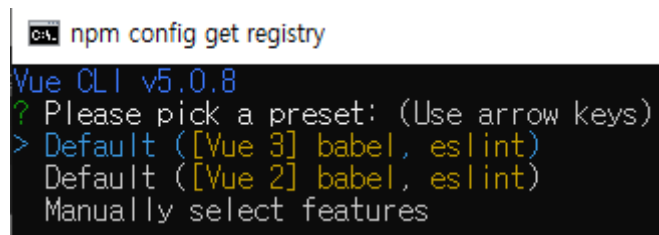
- npm install : 프로젝트 생성
  - 생성할 프로젝트 디렉토리가 존재해야 함
- npm server : 로컬 서버 구동

#### 4-1) vue cli : 기본 vue 개발 환경을 설정해주는 도구로 vue-cli 가 기본적인 프로젝트 세팅을 해줌

- CMD > cd work > md test > cd test > npm install -g @vue/cli
  - npm install -g @vue/cli
    - 노드에서 패키지를 설치할 것이다 vue cli를
      - vue cli 가 생성이 되어야 vue project를 생성할 수 있다
  - dir : 83 package-lock.json 확인됨
    - 기본적으로 vue 가 실행할 환경에 대한 설치, vue 가 설치된건 아님
- CMD > vue --version (현재 c:/work/test 위치)
  - vue 를 생성하기 위한 cli 명령어 집단

#### 5) Vue project 생성

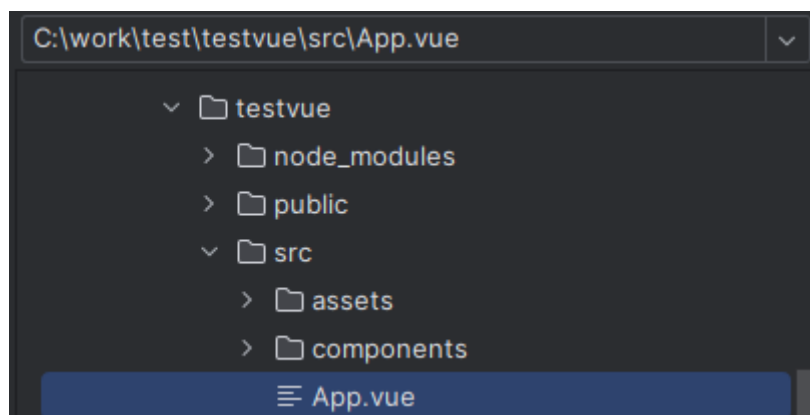
- vue create testvue > 화면 보이면 > enter
  - vue 의 프로젝트(testvue)를 만듦



```
C:\> npm config get registry
Vue CLI v5.0.8
? Please pick a preset: (Use arrow keys)
> Default ([Vue 3] babel, eslint)
  Default ([Vue 2] babel, eslint)
  Manually select features
```

#### 6) 현재까지의 과정 정리

- 노드(런타임, 운영체제) 설치 우선 > Vue cli 라는 패키지 생성 프로그램 설치 > Vue 프로젝트 생성



설치 잘되면 확인되는 디렉토리

## 7) IntelliJ 에서 확인

- vue 명령어로 작성된 것 확인
- <template> , <style> , <script> : vue 에서 HTML, CSS, JS 의 사용
- babel : 웹팩 기능 사용?
- jsconfig.json
- package.json : dependencies(의존성) 등 자동 생성됨

## 8) CMD 에서 실행(c:/work/test/testvue) : 프로젝트가 생성되어 있는 디렉토리에서 실행

- npm run serve : 내 호스트에 뷰 프로젝트를 띄워줌

```
DONE Compiled successfully in 6711ms

App running at:
- Local:   http://localhost:8080/
- Network: http://192.168.219.197:8080/

Note that the development build is not optimized.
To create a production build, run npm run build.
```

<http://192.168.219.197:8080/> 접속됨

## 8-1) 현재 과정 삭제

- npm uninstall -g @testvue
  - 파일은 남지만 노드에서 삭제됨
- rd testvue /s > y
  - 하위 디렉토리까지 삭제

## 9) IntelliJ 환경에서의 전체 과정 진행

- Terminal 실행 : Java로 만든것도 node 기반으로 실행
- md testvue : 디렉토리 만듦, 및 확인도 됨
- cd testvue : testvue 디렉토리로 이동해서
- npm init -y : 초기화 및 package.json이 보임, 원래 install 전에 해야하는 듯
- dir : package.json 확인됨
- nvm ls : 사용중인 nvm 확인됨
- npm install -g @vue/cli : 기본적인 프로젝트 세팅
  - vue -- version : cli 버전 확인
- vue create testvue : testvue 프로젝트 생성
  - 5) 화면처럼 확인됨

## 9-1) [manual select features](#)

- manual select features > enter
- typescript 원하면 spacebar 로 선택 > enter
  - Babel : es6이상, TypeScript로 개발 시 브라우저 버전에 상관없이 자동 전환해주는 기능
  - TypeScript : TypeScript 개발시 지원해주는 기능
  - Progressive Web App (PWA) Support : Web App 개발을 지원해주는 기능
  - Router : Vue Router를 위한 기능, 파일간의 위치 경로설정, [컴포넌트](#)로 묶여 있음
  - Vuex : Vue에서 상태관리를 위한 기능, vue [생명주기](#)
  - CSS Pre-processors : CSS 작성을 위한 CSS 전 처리 기능
  - Linter / Formatter : JavaScript coding Convention, JS 코딩시 빠른 변환
  - Unit Testing : 단위 테스트
  - E2E Testing : end to end Testing, 통합 테스트
- 2.x

- vue 버전 선택
- use class-style > y
- use babel > y
- use history made for router > y
  - router history 사용할지의 여부
  - proper : 상속, vue는 양방향 전달이 가능 / react 는 단방향
- ESLint + airbnb config : 표준가이드, 주로 채택
- pick additional lint feature : Lint on save > enter
  - 코딩 컨벤션이 잘 지켜졌는지 체크하는 시점
- where do you prefer placing config for babel, eslint, etc : In package.json > enter
  - Babel, ESLint 설치를 하게 되면 파일 설정들을 별도로 config files에 만들지 package.json에 넣을지 선택
- save this as a preset for future projects > y
  - 앞서 선택한 옵션을 저장해 다음 프로젝트 생성시 동일한 preset을 만들것인가
- save preset as > enter
  - preset 의 이름 입력
  - preset 생성 확인 : vue create vue-preset

#### 9-2) npm run serve (c:/work/test/testvue/testvue)

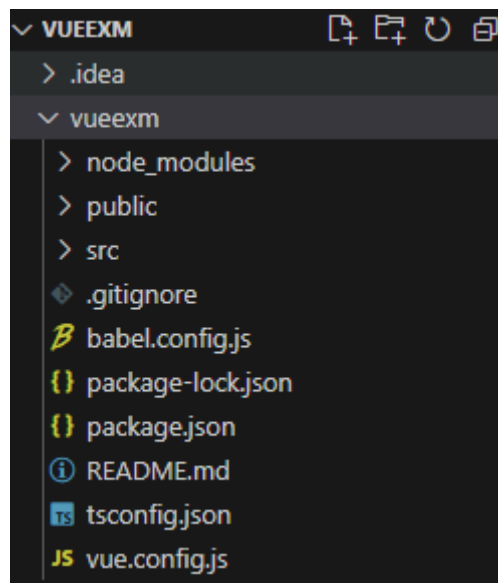
- cd testvue > npm run serve
- <http://192.168.219.197:8080/> 접속됨
  - 종료 : ctrl + c

#### 10) VS Code (c:/work/test)

- npm list -g : 노드 전체에 관련된 내용

```
PS C:\work\test> npm list -g
C:\Program Files\nodejs -> .\
├─ @vue/cli@5.0.8
├─ corepack@0.17.0
└─ npm@9.5.1
```

#### 10-1) VS Code 에서 Vue 프로젝트 목록 확인



- node\_modules : 노드 모드 설치 디렉토리
- public : 정적파일 저장처
- favicon : 아이콘

- `index.html` : 레이아웃 웹페이지, 브라우저에서 시현되는 `html`
- `src` : `source file`, 원본자료
- `src ~ components + public` : 빌드, 배포시
- `app` : 시작파일 `src` 밑에서 작업, `vue`의 시작
- `package.json` : `vueexm` 기준의 `json` 파일
  - 상위 디렉토리에 있다면 그것은 `node` 기준의 `json` 파일