

MERN stack을 활용한 프로젝트 개발

비트캠프 KDT 5기 수강생 최건일

< 목차 >

1. MERN stack이란?
2. Node.js
 - a. Node란
 - b. 환경설정
3. React.js
 - a. React.js란
 - b. Next.js
 - c. 환경설정
4. Express.js
 - a. Express.js란
 - b. 환경설정
5. Mongo DB
 - a. MongoDB란
 - b. 환경설정
6. MERN stack을 활용한 CRUD 구현
 - a. 페이지 구성
 - b. 라이브러리
(Redux, RTK, Axios, @material-ui, mongoose, Bcrypt, JWT)
7. 참조

1. MERN stack

MERN stack은 M: MongoDB(NoSQL 데이터베이스), E: Express.js (서버), R: React.js(프론트), N: Node.js(실행 환경)을 줄인 말이다. MERN 스택은 위 네가지 기술로 웹 사이트를 개발하는 것을 말하며 모두 자바스크립트와

관련 있는 기술이다. 즉, 하나의 언어로 프론트, 백, DB까지 풀스택 개발을 진행하는 것을 말한다.

자바스크립트로 개발을 진행하기 때문에 통일성과 안정성을 가지고, 네가지 구성 모두 오픈소스로 무료로 사용 가능하다는 장점이 있다.

2. Node.js

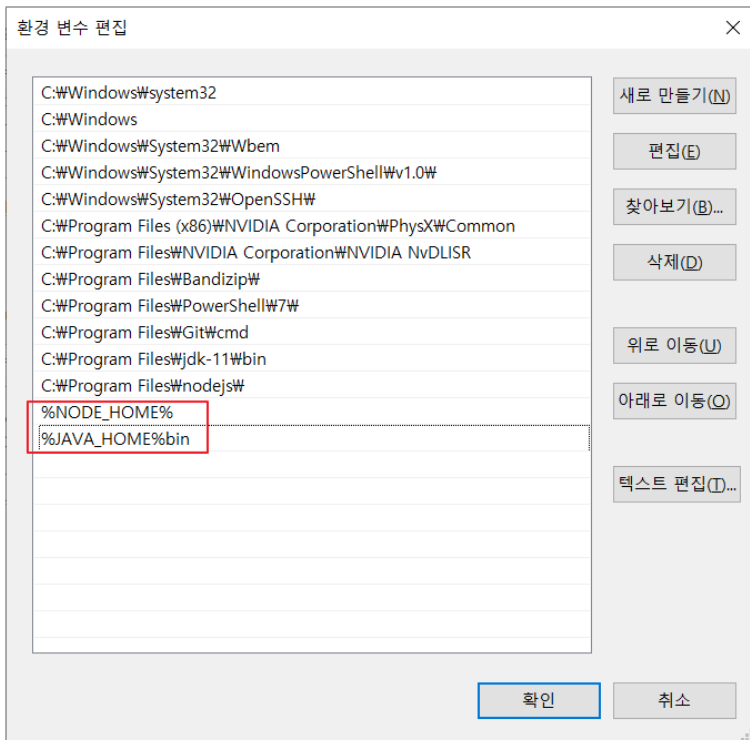
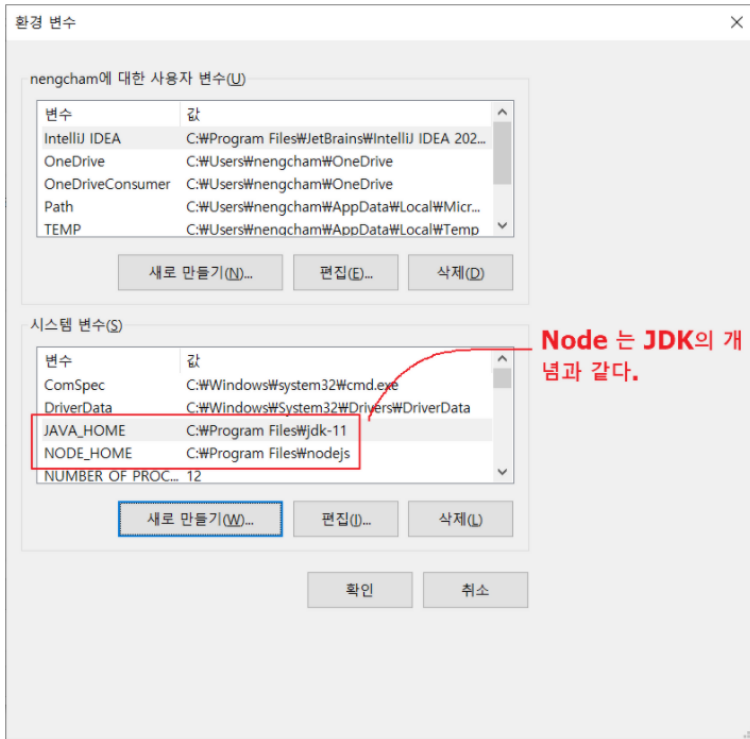
a. Node.js란?

Node.js는 구글 크롬의 자바스크립트 엔진에 기반해 만들어진 서버사이드 플랫폼이다. 자바스크립트의 런타임 환경이며, 브라우저가 아닌 환경에서도 자바스크립트 프로그램을 실행 할 수 있게 한다.

Node.js는 확장성이 있는 네트워크 어플리케이션 개발에 사용되는 소프트웨어 플랫폼이다. 특히 서버사이드에서 많이 사용되고 있다. 사용되는 언어로는 자바스크립트를 사용하며, 내장 HTTP 서버 라이브러리를 포함하고 있어 더 많은 통제에서 벗어나 여러가지 기능을 가능하게 한다.

b. Node.js 환경설정

Node.js는 자바스크립트 기반이기 때문에 Editing tool 인 Visual Studio Code로 쉽게 개발환경을 구축 할 수 있다. Window 운영체제에서는 공식 사이트를 통해 설치파일을 받아 진행 할 수 있다. 다운받은 설치파일로 설치를 하면 npm도 같이 설치된다. 이후 시스템속성 > 환경변수 설정에 들어가 설정을 완료한다. 시스템변수에서 새로 만들디 버튼 클릭 후 변수이름과 변수 값을 아래와 같이 입력한다. 시스템 변수 중 Path를 선택한 후 편집 버튼을 클릭하여 아래와 같은 형태로 변수를 등록한다.



이후 cmd에서 노드, npm 설치여부를 확인 후 yarn 을 설치한다.

```
node --version
npm --version // 노드, npm 버전 체크
npm i -g yarn // yarn 설치
```

yarn은 npm(node package manager)보다 높은 성능과 보안은 가진 매니저이다.

3. React.js

a. React.js란?

React.js는 사용자 인터페이스를 만들기 위해 사용되는 자바스크립트 라이브러리이다. 리엑트는 **facebook**에서 제공해주는 프론트엔드 라이브러리라고 볼 수 있으며 싱글 페이지 어플리케이션(**SPA**)이나 모바일 애플리케이션 개발 시 토대로 사용될 수 있다. 현재 가장 많은 점유율을 가진 자바스크립트 라이브러리이다.

리엑트를 사용하는 이유와 특징은 다음과 같다. 리엑트는 데이터 흐름이 한 방향으로 흐르는 단방향 데이터 흐름을 가진다. 복잡한 앱에서도 데이터흐름에서 일어나는 변화를 보다 예측 가능할 수 있다. 리엑트는 컴포넌트 (Component)기반 구조이다. 컴포넌트는 프로그래밍에서의 한 부분을 의미하며, 재사용이 가능한 최소단위 (Atom)을 뜻한다. 한 페이지 내에서도 여러 각 부분을 독립된 컴포넌트로 만들고, 이 컴포넌트를 조립해 화면을 구성한다. 이러한 컴포넌트 기반의 구조는 전체 코드를 파악하기 쉬우며 기능, UI단위로 캡슐화하여 코드를 관리하기 때문에 코드를 관리하기 때문에 재사용성이 높고, 유지보수 및 관리가 용이하다는 장점을 가진다. 리엑트는 자바스크립트 문법과 **HTML** 태그가 혼용되어 사용되는 방식은 **JSX**문법을 사용한다. **HTML** 친화적이고 마크업과 로직을 분리하지 않고 **UI**작성이 가능하다는 장점이 있다.

b. Next.js란?

Next.js는 생산을 위한 리엑트 프레임워크 이다. 넥스트는 서버측 랜더링(**SSR : Server Side Rendering**) 및 정적 웹사이트 생성과 같은 **React** 기반 웹 애플리케이션 기능을 가능하게 한다. 넥스트를 사용하여 개발된 **SSR** 웹 사이트 는 서버에서 미리 랜더링 된 페이지를 가져올수

있게 해주므로 사용자에게 더 좋은 경험을 주고, 검색 엔진에 잘 노출 될 수 있도록 하는 **SEO(Search Engine Optimization)**을 가능하게 한다. 또한 **SSR**과 **CSR**을 혼합한 형태 또한 구현 가능하다. **Next.js**는 **pre-rendering** 뿐만 아니라 페이지 기반 라우팅 시스템을 제공하기 때문에 개발경험 (**DX : Developer Experience**) 또한 높여준다.

c. 환경설정

React는 마이크로소프트에서 개발한 에디터 툴인 **Visual Studio Code**(이하 **vscode**)로 작업을 진행 하였다. **vscode**는 웹 기술 기반의 에디터라서 매끄럽게 지원하고, 개발 환경 확장이 용이하다는 장점을 가진다.

vscode 설치 후 **next.js**를 설치한다. 새 프로젝트를 진행할 폴더를 생성 후 **cmd**를 열어 아래의 명령어를 입력한다.

```
npx create-next-app 프로젝트 이름
```

패키지가 설치되면 새 앱폴더로 디렉토리를 변경 한 후 **code .** 를 사용하여 **vscode**에서 **next.js** 가 만들어 진 것을 확인할 수 있다. 이후 **vscode** 터미널에서 '**yarn dev**' 명령어 를 입력하여 지정된 포트에 '**Welcome to next.js**' 화면이 나오면 **next** 설치가 완료된 것이다.

4. Express.js

Express.js는 노드기반 웹 프레임워크 중 하나로, MIT 허가서로 라이선스 되는 자유-오픈소스 소프트웨어 이다. 웹 애플리케이션, API 개발을 위해 설계되었으며, **Node.js**의 핵심 모듈인 **HTTP**와 **Connect** 컴포넌트를 기반으로 하는 사실상의 표준 서버 프레임워크로 불린다. 그러한 컴포넌트들을 미들웨어(**middleware**)라고 하며, 설정보다는 관례(**convention over configuration**)와 같은 프레임워크의 철학을 지탱하는 주춧돌에 해당한다. 즉, 개발자들은 특정 프로젝트에 필요한 라이브러리를 어떤 것이든 자유롭게 선택할 수 있으며, 이는 개발자들에게

유연함과 높은 수준의 맞춤식 구성을 보장한다.

익스프레스는 웹 앱에 **MVC** 형태의 구조를 제공하며, **RESTful API** 등의 기능을 제공한다.

a. 환경설정

Express.js의 환경설정은 **next.js**와 유사한 방식으로 진행된다. 새 프로젝트를 진행 할 폴더를 생성 후 **cmd**를 열어 아래의 명령어를 입력한다.

```
npx express-generator
```

이후 터미널에서 '**node app**' 명령어를 입력 후 지정된 포트에 맞춰 접속하여 실행이 완료 되었는지 확인다.

5. Mongo DB

MongoDB는 오픈소스 문서지향(Document- Oriented) 크로스플랫폼 데이터 베이스이다. **Document**는 **JSON** 형태의 **key-value** 쌍으로 이루어진 데이터 구조를 말한다. 스키마가 없기 때문에 각 필드는 서로 다른 데이터타입을 가질 수 있고, 데이터 베이스에 저장된 **Documents**도 각기 다른 다양한 필드를 가질 수 있다.

a. 환경설정

MongoDB는 **Docker**를 사용하여 진행하였다. 먼저 공식 홈페이지에서 도커를 설치 후 **cmd** 창에서 아래 명령어를 입력하여 **mongoDB** 이미지를 다운받고 초기설정을 한다.

```
docker pull mongo
docker run --name mongoddb -e
MONGO_INITDB_ROOT_USERNAME=root -e
MONGO_INITDB_ROOT_PASSWORD=root -d -p
27017:27017 mongo
```

이후 **MongoDB GUI**툴인 **Compass** 혹은 **mongosh**를 이용하여 데이터베이스를 관리할 수 있다. **Mongosh**를 이용할 경우 아래의 명령어로 상태를 확인, 관리할 수 있다.

```
mongosh "mongodb://localhost:27017" --username
```

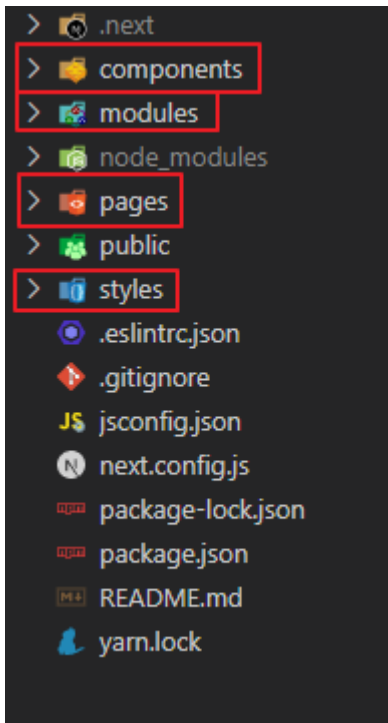
```
"root" --password "root"
```

```
show dbs // db확인
use db이름 // db 선택
show collections // db내의 컬렉션 확인
db.컬렉션이름.find() // 컬렉션 내부 확인
```

6. MERN stack을 활용한 CRUD 구현

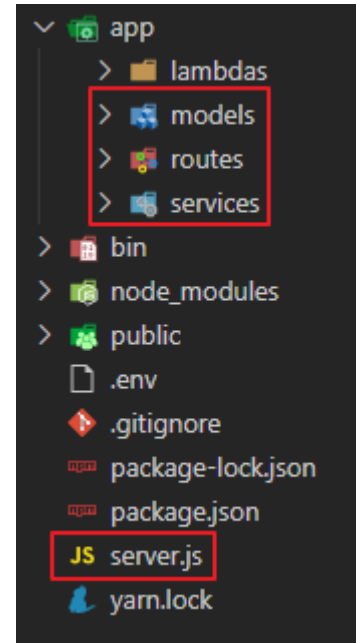
a. 페이지 구성

React, Next의 페이지 구성은 크게 components, pages, modules, styles로 구성되어 있다. components는 화면을 구성하는 React의 부분으로 상태를 저장하지 않은 컴포넌트들로 구성되어 있다. 부모의 관계를 가지는 pages 에서 bubbling capturing 으로 데이터와 이벤트가 전달된다. pages는 Next의 부분이며 components와 modules가 연결되어 있는 구성이다. modules는 상태를 관리하는 Redux 부분이다. styles는 컴포넌트 화면 ui/ux를 꾸미는 css 파일들을 모은 폴더이다.



Express의 폴더 구성은 아래의 사진과 같다. Next.js 에서 넘어오는 데이터는 엔트리포인트인 server.js를 통해

들어오고 서버의 구성요소인 라우터는 routes 폴더에, 미들웨어는 services 폴더에 구성하였다. Models 폴더는 mongoDB와 연결된다.



b. 사용한 라이브러리

Redux

Saga

Axios

Material-ui

dotenv

Mongoose

Bcrypt

JWT

7. 참조

<https://nodejs.org/>

<https://hanamon.kr/nodejs>

<https://reactjs.org/>

<https://velog.io/@kyusung/react-summary>

<https://nextjs.org/>

<https://expressjs.com/>

<https://wikibook.co.kr/article/what-is-expressjs/>

<https://www.mongodb.com/>

<https://evandde.github.io/chocolatey/>