

3/5 개발일지

1. 시작 전

- 프론트엔드 : REACT
- 백엔드 : SpringBoot (기존엔 장고섬)
- 데이터베이스 : MariaDB(기존 SQL섬)

2. JDK 다운로드 및 Spring설치

▼ 2-1

Java Archive Downloads - Java SE 17

 <https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html>

- 호환성이 높은 자바 17 선택
- 이에 맞는

▼ 2-2

<https://start.spring.io/>

자바 17에 호환되는 옵션 선택

3. 첫 문제 봉착

- ▼ localhost:8080 포트로 접속을 했는데 아래와같이 연결을 거부함



사이트에 연결할 수 없음

localhost에서 연결을 거부했습니다.

다음 방법을 시도해 보세요.

- 연결 확인
- [프록시 및 방화벽 확인](#)

ERR_CONNECTION_REFUSED

새로고침

세부정보 숨기기

인터넷 연결 확인

케이블을 확인하고 사용 중인 라우터, 모뎀 또는 기타 네트워크 기기를 재부팅하시기 바랍니다.

Chrome이 방화벽이나 백신 프로그램이 설정된 네트워크에 액세스할 수 있도록 허용합니다.

이미 네트워크 액세스가 허용된 프로그램으로 되어 있는 경우 목록에서 삭제한 뒤 다시 추가합니다.

프록시 서버를 사용하는 경우

프록시 설정을 확인하거나 네트워크 관리자에게 문의하여 프록시 서버가 작동하는지 확인하세요. 프록시 서버를 사용하지 않으려면 다음 단계를 따르세요. Chrome 메뉴 > 설정 > 고급 설정 표시 > 프록시 설정 변경... > LAN 설정으로 이동하여 '사용자 LAN에 프록시 서버 사용'을 선택 취소합니다.

문제 해결을 위해 다음과같이 생각할 수 있음.

▼ 문제 해결법

1. 애플리케이션 포트 확인

- 스프링 부트 애플리케이션이 실제로 **8080** 포트에서 실행되고 있는지 확인해야 합니다. **application.properties** 또는 **application.yml** 파일에서 서버 포트가 다르게 설정되어 있지 않은지 확인하세요.

2. 네트워크 인터페이스 설정

- 애플리케이션이 **localhost** 또는 **127.0.0.1** 에서 수신 대기하고 있는지 확인하세요. VPN 사용 중이거나 특정 네트워크 설정으로 인해 **localhost** 가 올바르게 해석되지 않을 수 있습니다.

3. 방화벽 설정

- Windows 방화벽이나 보안 소프트웨어가 **8080** 포트로의 접속을 차단하고 있을 수 있습니다. 방화벽 설정을 확인하여 해당 포트로의 접근을 허용해 보세요.

4. 다른 프로세스가 포트를 사용 중

- **8080** 포트를 다른 애플리케이션이 이미 사용하고 있을 가능성이 있습니다. 이 경우, 스프링 부트 애플리케이션이 해당 포트에서 제대로 실행되지 않을 수 있습니다. 포트 사용 상황을 확인하려면, 커맨드 라인에서 다음 명령어를 실행하세요:

```
shellCopy code
netstat -aon | findstr :8080
```

실행 중인 프로세스가 있다면, 해당 프로세스 ID(PID)를 사용하여 해당 프로세스를 종료하거나, 스프링 부트 애플리케이션의 포트를 변경해야 할 수 있습니다.

5. 애플리케이션 로그 확인

- 스프링 부트 애플리케이션의 로그를 자세히 확인하여, 시작 과정에서 어떤 경고나 오류 메시지가 출력되었는지 확인하세요. 특정 구성이나 의존성 문제로 인해 애플리케이션이 정상적으로 작동하지 않을 수 있습니다.

6. VPN 설정 확인

- 앞서 언급했듯이, VPN을 사용 중인 경우 VPN 설정이 로컬 네트워크 접속을 방해할 수 있습니다. VPN 연결을 일시적으로 해제하고 접속을 시도해 보세요.

이러한 점검 사항들을 확인한 후에도 문제가 해결되지 않는다면, 애플리케이션의 구체적인 구성이나 네트워크 설정을 다시 검토할 필요가 있습니다. 때로는 애플리케이션의 특정 로그 메시지나 설정이 문제의 해답을 제공할 수 있습니다.




터미널에서만 실행되면 어차피 백엔드 서버는 구동중이니 상관없다는 결론에 이름.

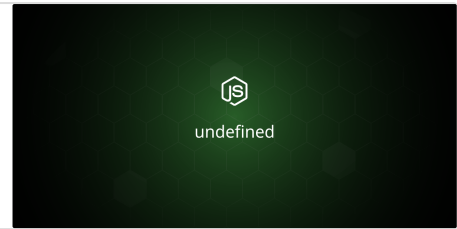
4. 프론트엔드 구성(React 설치)

▼ Node.js, npm 설치

Node.js

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

 <https://nodejs.org/en>



▼ 아래는 node 터미널에서 실행한 목록입니다.

```
PS C:\Users\sdkim\Desktop\Project\projects> node -v
v20.11.1
```

```
PS C:\Users\sdkim\Desktop\Project\projects> npm -v
10.2.4
```

```
PS C:\Users\sdkim\Desktop\Project\projects> npm install -g create-
react-app
```

```
npm WARN deprecated tar@2.2.2: This version of tar is no longer
supported, and will not receive security updates. Please upgrade asap.
```

```
added 66 packages in 2s
```

```
4 packages are looking for funding
run
```

```
npm fund for details
```

```
npm notice
```

```
npm notice New minor version of npm available! 10.2.4 → 10.5.0
```

```
npm notice Changelog:
```

```
https://github.com/npm/cli/releases/tag/v10.5.0
```

```
npm notice Run npm install -g npm@10.5.0 to update!
```

```
npm notice
```

```
PS C:\Users\sdkim\Desktop\Project\projects> cd web1
```

```
PS C:\Users\sdkim\Desktop\Project\projects\web1> dir
```

```
디렉터리: C:\Users\sdkim\Desktop\Project\projects
\web1
```

Mode	LastWriteTime	Length	Name
------	---------------	--------	------

```

d----- 2024-03-05 오전 9:41      .gradle
d----- 2024-03-05 오전 9:43      bin
d----- 2024-03-05 오전 9:41      build
d----- 2024-03-05 오전 9:14      gradle
d----- 2024-03-05 오전 9:14      src
-a----- 2024-03-05 오전 9:14    444 .gitignore
-a----- 2024-03-05 오전 9:14    437 build.gradle
-a----- 2024-03-05 오전 9:14   8692 gradlew
-a----- 2024-03-05 오전 9:14   2868 gradlew.bat
-a----- 2024-03-05 오전 9:14     574

```

HELP.md

```
-a----- 2024-03-05 오전 9:14     26 settings.gradle
```

```
PS C:\Users\sdkim\Desktop\Project\projects\web1> cd src
```

```
PS C:\Users\sdkim\Desktop\Project\projects\web1\src> dir
```

```

디렉터리: C:\Users\sdkim\Desktop\Project\projects
\src

```

Mode	LastWriteTime	Length	Name
d-----	2024-03-05 오전 9:14		main
d-----	2024-03-05 오전 9:14		test

```
PS C:\Users\sdkim\Desktop\Project\projects\web1\src> cd main
```

```
PS C:\Users\sdkim\Desktop\Project\projects\web1\src\main> dir
```

```

디렉터리: C:\Users\sdkim\Desktop\Project\projects
\src\main

```

Mode	LastWriteTime	Length	Name
d-----	2024-03-05 오전 9:14		java
d-----	2024-03-05 오전 10:35		resources

```
PS C:\Users\sdkim\Desktop\Project\projects\web1\src\main> cd
resources
```

```
PS C:\Users\sdkim\Desktop\Project\projects\web1\src\main\resources>
dir
```

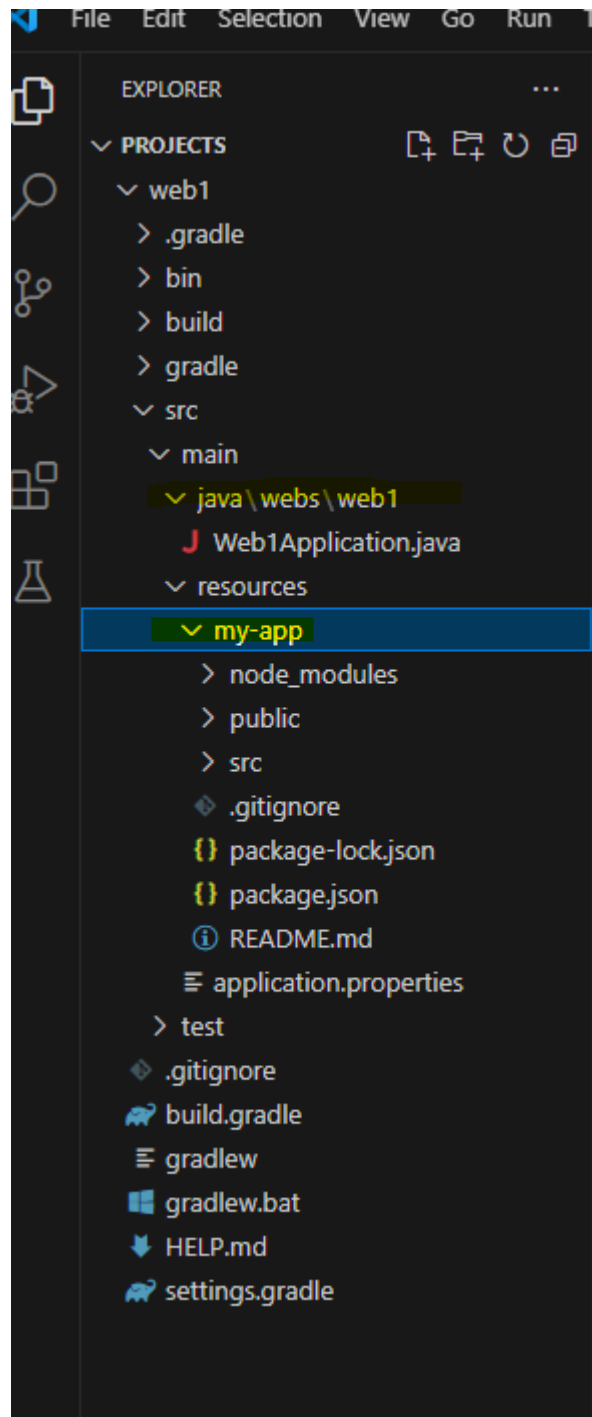
디렉터리: C:\\Users\\sdkim\\Desktop\\Project\\projects\\web1\\src\\main\\resources

Mode	LastWriteTime	Length	Name
------	---------------	--------	------

- a---- 2024-03-05 오전 9:56 0 application.properties

```
PS C:\Users\sdkim\Desktop\Project\projects\web1\src\main\resources>
npx create-react-app my-app
```

- 프론트엔드(my-app), 백엔드(java/..) 가 잘 설치된 모습입니다.



Compiled successfully!

You can now view my-app in the browser.

Local: <http://localhost:3000>

On Your Network: <http://172.22.32.1:3000>

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

webpack compiled successfully

5. 프로그램 설계서 작성

과제 수행 계획서

- 시퀀스 다이어그램까지 작성 완료함.

6. 프론트엔드 개발 시작(React)

▼ 6-1 문제봉착

- a. react-router-dom을 설치했지만, 설치된 패키지가 자꾸 useRef(Null)오류를 일으킴.
- b. node 터미널에선 정상적으로 컴파일 되었다고함.

▼ 오류난 모습

ERROR

Cannot read properties of null (reading 'useRef')

TypeError: Cannot read properties of null (reading 'useRef')

at Object.useRef (

<http://localhost:3000/static/js/bundle.js:34935:25>)

at BrowserRouter (

<http://localhost:3000/static/js/bundle.js:31134:55>)

at renderWithHooks (

<http://localhost:3000/static/js/bundle.js:56219:22>)

at mountIndeterminateComponent (

<http://localhost:3000/static/js/bundle.js:59503:17>)

at beginWork (

<http://localhost:3000/static/js/bundle.js:60799:20>)

at HTMLUnknownElement.callCallback (

<http://localhost:3000/static/js/bundle.js:45815:18>)

at Object.invokeGuardedCallbackDev (

<http://localhost:3000/static/js/bundle.js:45859:20>)

at invokeGuardedCallback (

<http://localhost:3000/static/js/bundle.js:45916:35>)

at beginWork\$1 (

<http://localhost:3000/static/js/bundle.js:65780:11>)

at performUnitOfWork (

<http://localhost:3000/static/js/bundle.js:65028:16>)

ERROR

Cannot read properties of null (reading 'useRef')

TypeError: Cannot read properties of null (reading 'useRef')

at Object.useRef (

<http://localhost:3000/static/js/bundle.js:34935:25>)

at BrowserRouter (

<http://localhost:3000/static/js/bundle.js:31134:55>)

at renderWithHooks (

<http://localhost:3000/static/js/bundle.js:56219:22>)

at mountIndeterminateComponent (

<http://localhost:3000/static/js/bundle.js:59503:17>)

at beginWork (

<http://localhost:3000/static/js/bundle.js:60799:20>)

at HTMLUnknownElement.callCallback (

<http://localhost:3000/static/js/bundle.js:45815:18>)

at Object.invokeGuardedCallbackDev (

<http://localhost:3000/static/js/bundle.js:45859:20>)

at invokeGuardedCallback (

<http://localhost:3000/static/js/bundle.js:45916:35>)

at beginWork\$1 (

<http://localhost:3000/static/js/bundle.js:65780:11>)

at performUnitOfWork (

<http://localhost:3000/static/js/bundle.js:65028:16>)

ERROR

Cannot read properties of null (reading 'useRef')

TypeError: Cannot read properties of null (reading 'useRef')

at Object.useRef (

<http://localhost:3000/static/js/bundle.js:34935:25>)

at BrowserRouter (

<http://localhost:3000/static/js/bundle.js:31134:55>)

at renderWithHooks (

<http://localhost:3000/static/js/bundle.js:56219:22>)

at mountIndeterminateComponent (

<http://localhost:3000/static/js/bundle.js:59503:17>)

at beginWork (

<http://localhost:3000/static/js/bundle.js:60799:20>)

at beginWork\$1 (

<http://localhost:3000/static/js/bundle.js:65758:18>)

at performUnitOfWork (
http://localhost:3000/static/js/bundle.js:65028:16)
at workLoopSync (
http://localhost:3000/static/js/bundle.js:64951:9)
at renderRootSync (
http://localhost:3000/static/js/bundle.js:64924:11)
at recoverFromConcurrentError (
http://localhost:3000/static/js/bundle.js:64416:24)

▼ 6-2 문제해결

1. 오류가 난 이유는 패키지의 의존성 문제였음 (react-router-dom 패키지를 실수로 루트디렉토리 바깥에 설치함)
2. 의존성 문제를 해결하기 위해 여러가지 문제가 있었지만, 프로젝트 초반이라 node.js까지 지우고 새로 깔았음
3. 새로 디렉토리를 간 김에 C:\ 하위에 새로 프로젝트를 생성함

▼ 6-3 App.js, footer.js, navbar.js, mainpage.js 생성

- App.js

```
import { Routes, Route, BrowserRouter } from 'react-router-dom';
import { useState, useEffect } from 'react';
import './App.css';
import MainPage from './components/js/mainpage';
import NavBar from './components/js/navbar';
import Footer from './components/js/footer';

const App = () => {

  // 아래 주석문들은 백엔드의 jwt와 staff 알고리즘이 만들어지면 다

  // const [isLoggedIn, setIsLoggedIn] = useState(false);
  // const [isStaff, setIsStaff] = useState(false);

  // useEffect(() => {
```

```

//   const token = localStorage.getItem('token');
//   if(token) {
//       setIsLoggedIn(true);
//   } else {
//       setIsLoggedIn(false);
//   }
// }, []);

// useEffect(() => {
//   const level = Number(localStorage.getItem('level'))
//   if (level>=2 && isLoggedIn) {
//       setIsStaff(true);
//   } else {
//       setIsStaff(false);
//   }
// }, [isLoggedIn]);

return (
  <BrowserRouter>
    <NavBar />
    <Routes>
      <Route path="/" element={<MainPage />} />
    </Routes>
    <Footer />
  </BrowserRouter>
);
}

export default App;

```

- footer.js

```

const Footer = () => {
  return(
    <h1>This is footer</h1>
  );
}

```

```
export default Footer;
```

- navbar.js

```
import { useState, useEffect } from "react";

const NavBar = () => { // 차후에 isloggedin등등을 파라미터로 추;

  return(
    <h1>This is Navbar</h1>
  );
}

export default NavBar;
```

- mainpage.js

```
const MainPage = () => {
  return(
    <h1>hi</h1>
  );
}

export default MainPage;
```

- JWT 방식으로 헤더에 인증토큰과 스탭프면 레벨을 같이 넣어서 인증알고리즘을 만들 예정
- 위 방식을 사용하기 위해 백엔드와 데이터베이스의 개발 필요성을 느낌

7. 버전관리 (gitlab)