axios로 학생 목록 구현

**학습목표**

json-server, axios를 활용하여 학생 조회, 등록, 수정, 삭제 구현

react router를 활용하여 학생 목록 화면, 학생 수정 화면 구현

**목차**

[1. 라우터 2](#_Toc152767308)

[1) 개요 2](#_Toc152767309)

[2) 예제 3](#_Toc152767310)

[3) 예제 수정 7](#_Toc152767311)

[2. students4 프로젝트 9](#_Toc152767312)

[1) 개요 9](#_Toc152767313)

[2) 프로젝트 생성 10](#_Toc152767314)

[3) 구현 11](#_Toc152767315)

[4) 실행 17](#_Toc152767316)

# 라우터

## 개요

### single page app

웹페이지를 벗어나서 다른 웹페이지로 넘어갈 때,

이전 웹페이지에서 실행되던 자바스크립트 코드는 모두 종료된다.

웹페이지를 새로고침할 때도, 실행되던 자바스크립트 코드는 모두 종료된다.

따라서 프론트엔드 앱은 웹페이지 하나에서만 실행 가능하다.

그 웹페이지를 벗어나 다른 웹페이지로 넘어가거나,

그 웹페이지를 새로고침하면,

실행 중이던 프론트엔드 앱은 종료된다.

따라서 여러 화면으로 구성된 프론트엔드 앱도 하나의 웹페이지에서 실행된다.

### router

여러 화면으로 구성된 프론트엔드 앱을 구현하기 위한 라이브러리를 router 라고 부른다.

router를 이용해서 여러 화면들을 구현하면,

그 화면들이 별도의 웹페이지인 것 처럼 보이고,

화면이 바뀌는 것이, 웹페이지가 바뀌는 것 처럼 보인다.

그렇지만 각 화면은 웹페이지가 아니고 리액트 컴포넌트일 뿐이다.

화면이 바뀌는 것은 화면에 표시된 리액트 컴포넌트가 바뀌는 것이다.

### 프로젝트 생성

|  |
| --- |
| npx create-react-app router1 --template typescript |

### router 설치

프로젝트 디렉토리에서 다음 명령을 실행

|  |
| --- |
| npm install react-router react-router-dom |

### styled-components 설치

|  |
| --- |
| npm install styled-components @types/styled-components |

## 예제

### Home.tsx - 화면

|  |
| --- |
| import React from 'react'  function Home() {    return (      <div>        <h1>Home</h1>        <p>첫 화면 입니다.</p>      </div>    )  }  export default Home |

### Notice.tsx - 화면

|  |
| --- |
| import React from 'react'  function Notice() {    return (      <div>        <h1>Notice</h1>        <p>공지 화면 입니다.</p>      </div>    )  }  export default Notice |

### About.tsx - 화면

|  |
| --- |
| import React from 'react'  function About() {    return (      <div>        <h1>About</h1>        <p>소개 화면 입니다.</p>      </div>    )  }  export default About |

### Header.tsx - 메뉴

|  |
| --- |
| import React from 'react'  import { Link } from 'react-router-dom'  import styled from 'styled-components'  const Div = styled.div`    & a { margin-right: 20px; }  `;  function Header() {    return (      <Div>        <Link to='/'>Home</Link>        <Link to='/notice'>Notice</Link>        <Link to='/about'>About</Link>        <hr />      </Div>    )  }  export default Header |

<Link to='/'>Home</Link>

<Link to='/notice'>Notice</Link>

<Link to='/about'>About</Link>

Link 컴포넌트에 의해서 a 태그가 생성된다.

a 태그를 클릭하면 웹브라우저 주소칸의 URL이 '/', '/notice', '/about'로 바뀌지만,

웹페이지가 전환되는 것은 아니다.

### Footer.tsx

|  |
| --- |
| import React from 'react'  function Footer() {    return (      <div>        <hr />        <div>Footer</div>      </div>    )  }  export default Footer |

### App.tsx

|  |
| --- |
| import { BrowserRouter, Routes, Route } from 'react-router-dom'  import styled from 'styled-components'  import Header from './Header'  import Home from './Home'  import About from './About'  import Notice from './Notice'  import Footer from './Footer'  const Div = styled.div`    margin: 20px;  `;  function App() {    return (      <BrowserRouter>        <Div>          <Header />          <Routes>            <Route path="/" element={<Home/>} />            <Route path="/about" element={<About/>} />            <Route path="/notice" element={<Notice/>} />          </Routes>          <Footer />        </Div>      </BrowserRouter>    )  }  export default App |

<Routes>

 <Route path="/" element={<Home/>} />

 <Route path="/about" element={<About/>} />

 <Route path="/notice" element={<Notice/>} />

 <Route path="/" element={<Home/>} />

</Routes>

현재 URL이 '/' 이면 <Home /> 컴포넌트가 화면에 보이고,

'/notice 이면 <Notice /> 컴포넌트가 화면에 보이고,

'/about' 이면 <About /> 컴포넌트가 화면에 보인다.

A screenshot of a computer

Description automatically generated with medium confidence

### 개발자창의 Components 탭

A screenshot of a computer

Description automatically generated

웹브라우저 개발자창의 Components 탭에서, 렌더링된 컴포넌트와 그 속성값들을 확인할 수 있다.

## 예제 수정

### Page.tsx - 화면

|  |
| --- |
| import React from 'react'  type Props = {    title: string,    body: string  }  function Page({title, body}: Props) {    return (      <div>        <h1>{ title }</h1>        <p>{ body }</p>      </div>    )  }  export default Page |

각각의 화면들에서 출력되는 데이터만 다를 뿐이라면, 화면 컴포넌트 하나로 여러 화면을 구현할 수 있다.

### App.tsx

|  |
| --- |
| import { BrowserRouter, Routes, Route } from 'react-router-dom'  import styled from 'styled-components'  import Header from './Header'  import Page from './Page'  import Footer from './Footer'  const Div = styled.div`    margin: 20px;  `;  function App() {    return (      <BrowserRouter>        <Div>          <Header />          <Routes>            <Route path="/" element={<Page title='Home' body='첫 화면입니다.' />} />            <Route path="/notice" element={<Page title="Notice" body='공지 화면입니다.' />} />            <Route path="/about" element={<Page title="About" body='소개 화면입니다.' />} />          </Routes>          <Footer />        </Div>      </BrowserRouter>    )  }  export default App |

### Header.tsx

|  |
| --- |
| import { NavLink } from 'react-router-dom'  import styled from 'styled-components'  const Div = styled.div`    & a { padding: 3px 20px; color: black; text-decoration: none; }    & a.active { background-color: #eee; }  `;  function Header() {    return (      <Div>        <NavLink to='/'>Home</NavLink>        <NavLink to='/notice'>Notice</NavLink>        <NavLink to='/about'>About</NavLink>        <hr />      </Div>    )  }  export default Header |

Link 대신 NavLink 컴포넌트를 사용하면, 현재 URL과 일치하는 a 태그에 자동으로 'active' 클래스 이름이 주어진다. 예: <a class='active'>

A screenshot of a computer

Description automatically generated with medium confidence

# students4 프로젝트

## 개요

### URL로 이동하기

import { useNavigate } from 'react-router-dom'

const navigate = useNavigate();

navigate('/edit/' + student.id);

'/edit/3' 형태의 URL로 이동한다.

### URL과 컴포넌트 등록

<Route path="/edit/:id" element={<StudentEdit />} />

### URL에서 id 값 꺼내기

import { useParams } from 'react-router-dom'

const params = useParams(); // /edit/3 URL에서 3 부분을 꺼내기 위한 객체

const id: number = params.id ? parseInt(params.id) : 0; // URL에서 id 값을 꺼낸다

## 프로젝트 생성

### 프로젝트 생성

|  |
| --- |
| npx create-react-app students4 --template typescript |

### axsios 설치

|  |
| --- |
| npm install axios |

### react router 설치

|  |
| --- |
| npm install react-router react-router-dom |

### styled-components 설치

|  |
| --- |
| npm install styled-components @types/styled-components |

## 구현

### students-db.json 생성

|  |
| --- |
| { "departments":    [ {"id":1,"name":"소프트웨어공학과","shortName":"소프","phone":"02-2610-9563"},      {"id":2,"name":"컴퓨터공학과","shortName":"컴공","phone":"02-2610-9565"},      {"id":3,"name":"정보통신공학과","shortName":"정통","phone":"02-2610-9568"},      {"id":4,"name":"글로컬IT공학과","shortName":"글티","phone":"02-2610-9569"}],  "students":    [ {"id":1,"studentNo":"201532006","name":"최하은","phone":"010-4361-1145","sex":"여","email":"choi064@skhu.net","departmentId":1},      {"id":2,"studentNo":"201532007","name":"김진영","phone":"010-3415-1238","sex":"남","email":"kim073@skhu.net","departmentId":3},      {"id":3,"studentNo":"201532008","name":"나철진","phone":"010-8703-1239","sex":"남","email":"na088@skhu.net","departmentId":3},      {"id":4,"studentNo":"201532009","name":"이익수","phone":"010-7875-1251","sex":"남","email":"lee097@skhu.net","departmentId":4},      {"id":5,"studentNo":"201532010","name":"이제문","phone":"010-7700-1333","sex":"남","email":"lee107@skhu.net","departmentId":1},      {"id":6,"studentNo":"201532011","name":"김영우","phone":"010-2090-1421","sex":"남","email":"kim112@skhu.net","departmentId":1},      {"id":7,"studentNo":"201532012","name":"주한요","phone":"010-4624-1467","sex":"남","email":"joo124@skhu.net","departmentId":3},      {"id":8,"studentNo":"201532013","name":"김숙홍","phone":"010-3791-1522","sex":"남","email":"kim133@skhu.net","departmentId":1},      {"id":9,"studentNo":"201532014","name":"홍영수","phone":"010-2897-1525","sex":"남","email":"hong142@skhu.net","departmentId":4},      {"id":10,"studentNo":"201532015","name":"박원종","phone":"010-7655-1724","sex":"남","email":"park157@skhu.net","departmentId":1},      {"id":11,"studentNo":"201432015","name":"변준호","phone":"010-2245-1750","sex":"남","email":"byeon152@skhu.net","departmentId":1},      {"id":12,"studentNo":"201532016","name":"고희정","phone":"010-5691-1943","sex":"여","email":"ko165@skhu.net","departmentId":3},      {"id":13,"studentNo":"201432016","name":"신철대","phone":"010-3221-1956","sex":"남","email":"sin163@skhu.net","departmentId":4},      {"id":14,"studentNo":"201532017","name":"서윤정","phone":"010-4310-1965","sex":"여","email":"seo174@skhu.net","departmentId":3},      {"id":15,"studentNo":"201432017","name":"오화순","phone":"010-8527-2048","sex":"여","email":"oh178@skhu.net","departmentId":3},      {"id":16,"studentNo":"201532018","name":"박재인","phone":"010-2616-2144","sex":"남","email":"park182@skhu.net","departmentId":1},      {"id":17,"studentNo":"201532019","name":"최현복","phone":"010-7312-2171","sex":"남","email":"choi197@skhu.net","departmentId":2},      {"id":18,"studentNo":"201432019","name":"임봉영","phone":"010-7475-2192","sex":"남","email":"lim197@skhu.net","departmentId":2},      {"id":19,"studentNo":"201532020","name":"장영만","phone":"010-1527-2270","sex":"남","email":"jang201@skhu.net","departmentId":2},      {"id":20,"studentNo":"201432020","name":"박주용","phone":"010-1640-2396","sex":"남","email":"park201@skhu.net","departmentId":4},      {"id":21,"studentNo":"201532021","name":"황하지","phone":"010-6682-2463","sex":"남","email":"hwang216@skhu.net","departmentId":2},      {"id":22,"studentNo":"201532022","name":"유환치","phone":"010-3951-2464","sex":"남","email":"yu223@skhu.net","departmentId":4},      {"id":23,"studentNo":"201532023","name":"성희지","phone":"010-4267-2555","sex":"남","email":"seong234@skhu.net","departmentId":1},      {"id":24,"studentNo":"201532024","name":"박철용","phone":"010-2135-2585","sex":"남","email":"park242@skhu.net","departmentId":3},      {"id":25,"studentNo":"201532025","name":"이주연","phone":"010-7775-2588","sex":"남","email":"lee257@skhu.net","departmentId":4},      {"id":26,"studentNo":"201532026","name":"김철지","phone":"010-1611-2712","sex":"남","email":"kim261@skhu.net","departmentId":2},      {"id":27,"studentNo":"201532027","name":"이화상","phone":"010-2703-2794","sex":"남","email":"lee272@skhu.net","departmentId":4},      {"id":28,"studentNo":"201532028","name":"백영규","phone":"010-6971-2845","sex":"남","email":"pak286@skhu.net","departmentId":3},      {"id":29,"studentNo":"201532029","name":"김화인","phone":"010-6683-2856","sex":"여","email":"kim296@skhu.net","departmentId":4},      {"id":30,"studentNo":"201532030","name":"김만기","phone":"010-4194-2860","sex":"남","email":"kim304@skhu.net","departmentId":1},      {"id":31,"studentNo":"201632006","name":"김진우","phone":"010-3031-2894","sex":"남","email":"kim063@skhu.net","departmentId":4},      {"id":32,"studentNo":"201632007","name":"박남영","phone":"010-8388-2997","sex":"남","email":"park078@skhu.net","departmentId":2},      {"id":33,"studentNo":"201632008","name":"이식동","phone":"010-7392-3031","sex":"남","email":"lee087@skhu.net","departmentId":2},      {"id":34,"studentNo":"201632009","name":"박하정","phone":"010-2627-3076","sex":"여","email":"park092@skhu.net","departmentId":4}  ]  } |

### types.ts

|  |
| --- |
| export type Student = {    id: number,    studentNo: string,    name: string,    email: string,    phone: string,    sex: string,    departmentId: number,  } |

### StudentItem.tsx

|  |
| --- |
| import React from 'react'  import type { Student } from './types'  import { useNavigate } from 'react-router-dom'  type Props = {    student: Student  }  function StudentItem({ student }: Props) {    const navigate = useNavigate();    const onClick = () => navigate('/edit/' + student.id);    return (      <tr onClick={onClick}>        <td>{ student.id }</td>        <td>{ student.studentNo }</td>        <td>{ student.name }</td>        <td>{ student.email }</td>        <td>{ student.phone }</td>        <td>{ student.sex }</td>        <td>{ student.departmentId }</td>      </tr>    )  }  export default StudentItem; |

id 값이 3인 tr 태그를 클릭하면 '/edit/3' URL로 넘어간다.

### StudentList.tsx

|  |
| --- |
| import React, { useState, useEffect } from 'react'  import { useNavigate } from 'react-router-dom'  import styled from 'styled-components';  import axios from 'axios';  import StudentItem from './StudentItem'  import type { Student } from './types'  const Div = styled.div`    & > button { padding: 0.2em 1.5em; font-size: 11pt;      margin-bottom: 5px; float: right;    }    & table { border-collapse: collapse; width: 100%; }    & thead { background-color: #eee; text-align: center; }    & td { border: 1px solid gray; padding: 4px; }    & td:nth-child(1) { text-align: center; width: 30px; }    & tr:hover { background-color: #ffe; cursor: pointer; }  `;  function StudentList() {    const [students, setStudents] = useState<Student[]>([]);    const [errorMsg, setErrorMsg] = useState<string|null>(null);    // 백엔드에서 학생 목록을 조회    const loadStudents = async () => {      try {        const response = await axios.get("students");        setStudents(response.data);      } catch (error) {        const s = error instanceof Error ? error.message : error;        setErrorMsg(String(s));      }    };    // 마운트될 때 즉 컴포넌트 화면에 보이게 될 때 마다,    // loadStudents 함수를 호출한다.    useEffect(() => { loadStudents(); }, []);    // 학생 등록 버튼을 클릭할 때 "/create" URL로 이동하기 위한 함수    const navigate = useNavigate()    const onCreate = () => navigate('/create');    const trlist = students.map(student =>      <StudentItem key={student.id} student={student} />);    return (      <Div>        <button type="button" onClick={onCreate}>등록</button>        <table>          <thead>            <tr><td>id</td><td>학번</td><td>이름</td><td>이메일</td>            <td>전화</td><td>성별</td><td>학과</td>            </tr>          </thead>          <tbody>            {trlist}          </tbody>        </table>        { errorMsg && <div className="error">{errorMsg}</div> }      </Div>    )  }  export default StudentList |

### StudentEdit.tsx

|  |
| --- |
| import React, { useEffect, useState } from 'react'  import { useNavigate, useParams } from 'react-router-dom'  import styled from 'styled-components';  import axios from 'axios';  import type { Student } from './types'  const Div = styled.div`    & > div { margin-bottom: 20px; }    & input[type=text], input[type=email], input[type=tel] {      padding: 8px; width: 300px; font-size: 12pt; }    & select { padding: 8px; width: 300px; font-size: 12pt; }    & > div > label { margin-right: 20px; font-size: 12pt; }    & button { padding: 0.4em 1.5em; font-size: 12pt; margin-right: 5px; }  `;  const emptyStudent: Student = { id: 0, studentNo: '', name: '', email: '',    phone: '', sex: '', departmentId: 0  }  function StudentEdit() {    const [student, setStudent] = useState<Student>(emptyStudent);    const [errorMsg, setErrorMsg] = useState<string|null>(null);    const params = useParams(); // /edit/3 URL에서 3 부분을 꺼내기 위한 객체    const id: number = params.id ? parseInt(params.id) : 0; // URL에서 id 값을 꺼낸다    const onError = (error: any) => {      const s = error instanceof Error ? error.message : error;      setErrorMsg(String(s));    };    // id로 학생 조회    const loadStudent = async () => {      try {        // id 값으로 학생을 조회        const response = await axios.get("/students/" + id);        setStudent(response.data);      } catch (error) {        onError(error);      }    }    // 컴포넌트가 마운트 될 때, id 값으로 학생을 조회한다  // 학생등록 /create URL인 경우에는 id 값이 주어지지 않고,  // 이때는 학생을 조회하지 않는다    useEffect(() => { if (id) loadStudent(); }, []);    const navigate = useNavigate();    // 저장 버튼 클릭    const onSave = async () => {      try {        if (id > 0)          await axios.put("/students/" + id, student); // 학생 수정화면인 때        else          await axios.post("/students/", student); // 학생 등록 화면일 때        navigate('/'); // 목록 화면으로 나간다      } catch (error) {        onError(error);      }    }    // 삭제 버튼 클릭    const onDelete = async () => {      try {        if (window.confirm("삭제하시겠습니까?")) {        await axios.delete("/students/" + id);          navigate('/'); // 목록 화면으로 나간다        }      } catch (error) {        onError(error);      }    }    // 취소 버튼 클릭    const onCancel = () => navigate('/'); // 목록 화면으로 나간다    // 입력 폼 이벤트 구현    const onChange = (e: any) => {    setStudent({ ...student, [e.target.name]: e.target.value });  };    return (      <Div>        <h1>학생 { id ? '수정' : '등록' }</h1>        <div>          <input type="text" placeholder="학번" onChange={onChange}                 name="studentNo" value={student.studentNo} />        </div>        <div>          <input type="text" placeholder="이름" onChange={onChange}                 name="name" value={student.name} />        </div>        <div>          <input type="email" placeholder="이메일" onChange={onChange}                 name="email" value={student.email} />        </div>        <div>          <input type="tel" placeholder="전화번호" onChange={onChange}                 name="phone" value={student.phone} />        </div>        <div>          <label>            <input type="radio" name="sex" value="남"                  onChange={onChange} checked={student.sex === "남"} />            <span>남자</span>          </label>          <label>            <input type="radio" name="sex" value="여"                   onChange={onChange} checked={student.sex === "여"} />            <span>여자</span>          </label>        </div>        <div>          <select value={student.departmentId} name="departmentId" onChange={onChange}>            <option value={0}>학과를 선택하세요</option>            <option value={1}>소프</option>            <option value={2}>컴공</option>            <option value={3}>정통</option>            <option value={4}>인공지능</option>          </select>        </div>        <div>          <button type="button" onClick={onSave}>저장</button>          { (id > 0) && <button type="button" onClick={onDelete}>삭제</button> }          <button type="button" onClick={onCancel}>취소</button>        </div>        { errorMsg && <div className="error">{errorMsg}</div> }      </Div>    )  }  export default StudentEdit |

{ (id > 0) && <button type="button" onClick={onDelete}>삭제</button>}

id 값이 0 이 아닐 때만, 삭제 버튼이 보인다.

즉 수정 화면에서만 삭제 버튼이 보인다.

### StudentRoot.tsx

|  |
| --- |
| import React from 'react'  import { BrowserRouter, Routes, Route } from 'react-router-dom'  import styled from 'styled-components';  import StudentEdit from './StudentEdit';  import StudentList from './StudentList';  const Div = styled.div`    border: 1px solid gray; width: 800px; margin: 10px auto;    box-shadow: 5px 5px 5px #DDD; padding: 30px;    & .error { margin-top: 20px; color: red; border: 1px solid #faa; padding: 10px 20px;               border-radius: 10px; background-color: #fee; }  `;  function StudentRoot() {    return (      <Div>        <BrowserRouter>          <Routes>            <Route path="/" element={<StudentList />} />            <Route path="/edit/:id" element={<StudentEdit />} />            <Route path="/create/" element={<StudentEdit />} />          </Routes>        </BrowserRouter>      </Div>    )  }  export default StudentRoot; |

<Route path="/edit/:id" element={<StudentEdit />} />

URL이 /edit/3 형태이면, StudentEdit 화면 컴포넌트를 출력한다.

<Route path="/create/" element={<StudentEdit />} />

URL이 /create 일 때도, StudentEdit 화면 컴포넌트를 출력한다.

### App.tsx

|  |
| --- |
| import StudentRoot from './StudentRoot';  import axios from 'axios';  axios.defaults.baseURL = "http://localhost:3000";  function App() {    return (      <div>        <StudentRoot />      </div>    )  }  export default App |

## 실행

|  |
| --- |
| json-server --watch src/students-db.json |

|  |
| --- |
| npm run start |