

# 13. 리액트 라우터로 SPA 개발하기

---

작성일시: 2022년 5월 10일 오후 7:57 참고 도서: 리액트를 다루는 기술

## 13.1 라우팅이란?

웹 애플리케이션에서 라우팅은 사용자가 요청한 URL에 따라 알맞은 페이지를 보여주는 것을 의미함.

웹 애플리케이션을 만들 때 프로젝트를 하나의 페이지 혹은 여러 페이지로 구성할 수 있다. 이렇게 여러 개의 페이지로 구성된 웹 애플리케이션을 만들 때 페이지 별로 분리된 컴포넌트들을 하나의 프로젝트로 관리하기 위한 것이 라우팅 시스템이다.

리액트에서 라우트 시스템 구축을 위해 사용할 수 있는 것은 크게 두가지 이다.

- 리액트 라우터(React Router)

리액트 관련 라이브러리 중 가장 오래됐고, 가장 많이 사용되고 있습니다. 컴포넌트 기반으로 라우팅 시스템을 설정 할 수 있습니다.

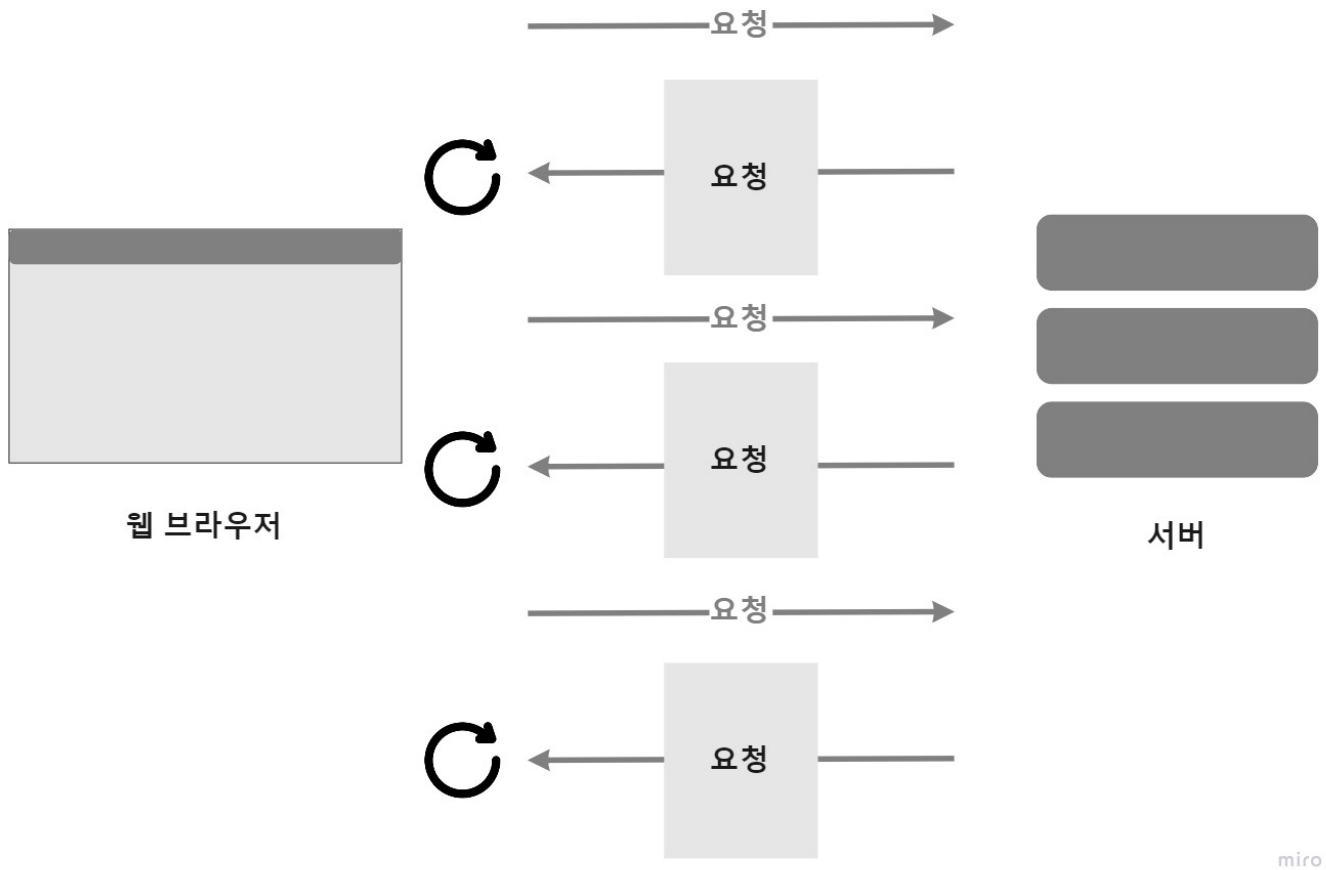
- Next.js

리액트 프로젝트의 프레임워크이다. Create React App 처럼 리액트 프로젝트를 설정하는 기능, 라우팅 시스템, 최적화, 다국어 시스템 지원, 서버 사이트 렌더링 등 다양한 기능을 제공합니다. 이 프레임워크의 라우팅 시스템은 파일 경로 기반으로 작동해서 리액트 라우터의 대안으로 많이 사용되고 있습니다.

## 13.2 싱글 페이지(SAP) 애플리케이션이란?

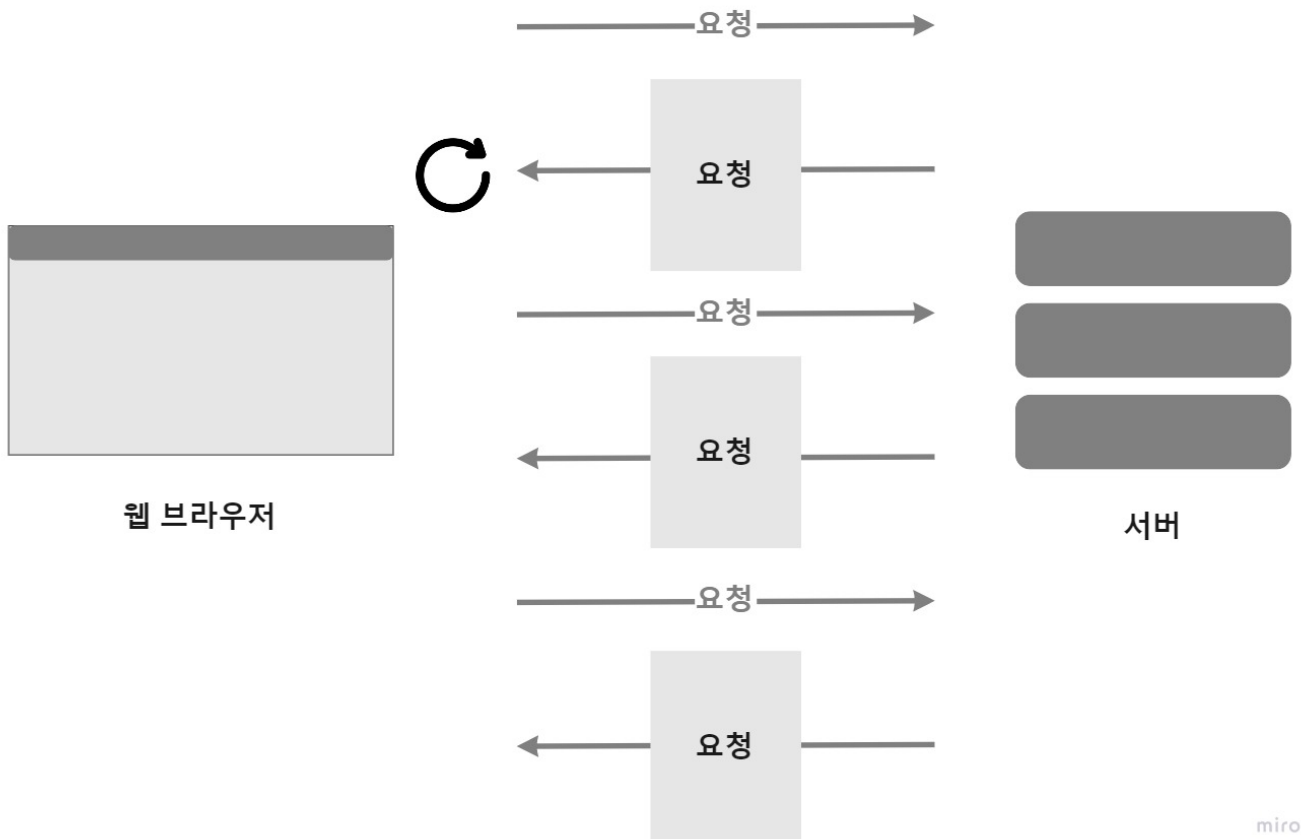
싱글 페이지 애플리케이션이란 하나의 페이지로 이루어진 애플리케이션이라는 의미이다.

### 다중페이지 원리 및 문제점



웹 서버라는 소프트웨어에 웹 브라우저가 접속한다. 이때, 전달되는 URL은 웹서버가 관리하는 폴더에 저장되어 있는 HTML파일 경로를 의미한다. 즉, 웹 브라우저가 웹 서버에 저장되어 있는 웹 페이지를 열람. 페이지 이동시마다 접속, 해제가 이뤄진다. 데이터에 따라 유동적인 html을 생성해주는 템플릿 엔진을 사용하기도 했다. ⇒ 하지만 그만큼 서버 자원을 사용하는 것이고, 트래픽도 더 많이 나올 수 있다.

### 싱글 페이지 원리



뷰 렌더링(Ajax)을 사용자가 담당하게 하여 웹 애플리케이션을 브라우저에 불러와 사용자와의 인터랙션이 발생하면 필요한 부분만 자바스크립트를 사용하여 업데이트 하는 방식으로 사용하게 되었다.

싱글 페이지 애플리케이션은 한 페이지만 존재하지만, 사용자가 경험하기에는 여러 페이지가 존재하는 것처럼 느낄 수 있다. → 브라우저의 History API를 사용하여 브라우저의 주소창의 값만 변경하고 기존 페이지에 띄웠던 화면은 그대로 유지하면서 라우팅 설정에 따라 또 다른 페이지를 보여준다.

### 싱글 페이지(SPA)의 단점

앱의 규모가 커지면 최초 접속시 모든 화면을 구성하는 스크립트를 한번에 모두 로딩하기 때문에 자바스크립트 파일이 너무 커진다.

## 13.3 리액트 라우터 적용 및 기본 사용법

### 13.3.1 프로젝트 생성 및 라이브러리 설치

`$ yarn create react-app 폴더명(숫자 및 소문자만 가능)`

디렉토리 위치로 이동

`$ cd 폴더명`

리액트 라우터를 위한 라이브러리 설치

`$ yarn add react-router-dom`

### 13.3.2 프로젝트에 라우터 적용

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
import { BrowserRouter } from 'react-router-dom';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <BrowserRouter>
    <App />
  </BrowserRouter>,
);
```

### 13.3.3 페이지 컴포넌트 만들기

Home.js

```
import React from 'react'

const Home = () => {
  return (
    <div>
      <h1>홈</h1>
      <p>가장 먼저 보여지는 페이지입니다.</p>
    </div>
  );
};

export default Home;
```

About.js

```
import React from 'react'

const About = () => {
  return (
    <div>
      <h1>소개</h1>
      <p>리액트 라우터를 사용해 보는 프로젝트입니다.</p>
    </div>
  );
};

export default About;
```

### 13.3.4 Route 컴포넌트로 특정 경로에 원하는 컴포넌트 보여주기

```
import React from "react";
import { Route,Routes } from "react-router-dom";
import About from "../pages/About";
import Home from "../pages/Home";

const App = () => {
  return (
    <div>
      <Routes>
        <Route path="/" exact={true} element={<Home/>}/>
        <Route path="/" element={<About/>}/>
      </Routes>
    </div>
  )
}

export default App
```

```
src > Appjs > App
1 import React from "react";
2 import { Route,Routes } from "react-router-dom";
3 import About from "../pages/About";
4 import Home from "../pages/Home";
5
6 const App = () => {
7   return (
8     <div>
9       <Routes>
10        <Route path="/" exact={true} element={<Home/>}/>
11        <Route path="/" element={<About/>}/>
12      </Routes>
13    </div>
14  )
15 }
16
17 export default App
18
```

홈

가장 먼저 보여지는 페이지입니다.

### 12.3.5 Link 컴포넌트를 사용하여 다른 페이지로 이동하는 링크 보여주기

리액트 라우터를 사용하는 프로젝트에서는 a 태그를 바로 사용하면 안된다. a태그는 클릭할때 브라우저에서 새로운 페이지를 불러오기 때문이다. Link 컴포넌트는 새로 불러오는 것을 막고 History API를 통해 브라우저 주소의 경로만 바꾸는 기능이 내장되어 있다.

```
import React from 'react'
import { Link } from 'react-router-dom';

const Home = () => {
  return (
    <div>
      <h1>홈</h1>
      <p>가장 먼저 보여지는 페이지입니다.</p>
      <Link to="/about">소개</Link>
    </div>
  );
};
```

```
export default Home;
```

```
src > pages > Home.js > ...
1 import React from 'react'
2 import { Link } from 'react-router-dom';
3
4 const Home = () => {
5   return (
6     <div>
7       <h1>홈</h1>
8       <p>가장 먼저 보여지는 페이지입니다.</p>
9       <Link to="/about">소개</Link>
10    </div>
11  );
12 };
13
14 export default Home;
```

**홈**

가장 먼저 보여지는 페이지입니다.

[소개](#)

**홈**

가장 먼저 보여지는 페이지입니다.

[소개](#)

**소개**

리액트 라우터를 사용해 보는 프로젝트입니다.

## 13.4 URL 파라미터와 쿼리스트링

- URL 파라미터란

주소 경로에 유동적인 값을 넣는 형태로, ID 또는 이름을 사용하여 특정 데이터를 조회할 때 사용된다.  
ex) /profile/velopart

- 쿼리스트링 파라미터란

주소의 뒷부분에 ? 문자열 이후 key=value로 값을 정의하며 &로 구분하는 형태이다. 주로 키워드 검색, 페이지네이션, 정렬 방식 등 데이터 조회에 필요한 옵션을 전달할 때 사용된다. ex)/articles?page=1&keyword=react

### 13.4.1 URL 파라미터

Profile.js

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

const data={
  velopert:{
    name:'김민준',
    description: '리엑트를 좋아하는 개발자'
  },
  gildong: {
    name:'홍길동',
    description: '고전 소설 홍길동전의 주인공',
  },
};

const Profile = () => {
  const params = useParams();
  const profile =data[params.username];
  return (
    <div>
      <h1>사용자 프로필</h1>
      {profile ?(
        <div>
          <h2>{profile.name}</h2>
          <p>{profile.description}</p>
        </div>
      ) : (
        <p>존재하지 않는 프로필입니다. </p>
      )}
    </div>
  );
};

export default Profile;
```

App.js

```
import React from "react";
import { Route,Routes } from "react-router-dom";
import About from "../pages/About";
import Home from "../pages/Home";
import Profile from "../pages/Profile";

const App = () => {
```

```
return (  
  <div>  
    <Routes>  
      <Route path="/" exact={true} element={<Home/>}/>  
      <Route path="/about" element={<About/>}/>  
      <Route path="/profiles/:username" element={<Profile/>}/>  
    </Routes>  
  </div>  
};  
  
export default App;
```

### Home.js

```
import React from 'react'  
import { Link } from 'react-router-dom';  
  
const Home = () => {  
  return (  
    <div>  
      <h1>홈</h1>  
      <p>가장 먼저 보여지는 페이지입니다.</p>  
  
      <ul>  
        <li>  
          <Link to="/about">소개</Link>  
        </li>  
        <li>  
          <Link to="/profiles/velopert">velopert의 프로필입니다.</Link>  
        </li>  
        <li>  
          <Link to="/profiles/gildong">gildong의 프로필입니다.</Link>  
        </li>  
        <li>  
          <Link to="/profiles/void">존재하지 않는 프로필</Link>  
        </li>  
      </ul>  
    </div>  
  );  
};  
  
export default Home;
```



```

src > pages > Homejs > default
1  import React from 'react'
2  import { Link } from 'react-router-dom';
3
4  const Home = () => {
5    return (
6      <div>
7        <h1>홈</h1>
8        <p>가장 먼저 보여지는 페이지입니다.</p>
9
10       <ul>
11         <li>
12           <Link to="/about">소개</Link>
13         </li>
14         <li>
15           <Link to="/profiles/velopert">velopert의 프로필입니다.</Link> "velopert"
16         </li>
17         <li>
18           <Link to="/profiles/gildong">gildong의 프로필입니다.</Link> "gildong": U
19         </li>
20         <li>
21           <Link to="/profiles/void">존재하지 않는 프로필</Link>
22         </li>
23       </ul>
24     </div>
25   );
26 };
27
28 export default Home;

```

## 홈

가장 먼저 보여지는 페이지입니다.

- [소개](#)
- [velopert의 프로필입니다.](#)
- [gildong의 프로필입니다.](#)
- [존재하지 않는 프로필](#)

← → ↺ 🏠 ⓘ localhost:3002/profiles/gildong

# 사용자 프로필

## 홍길동

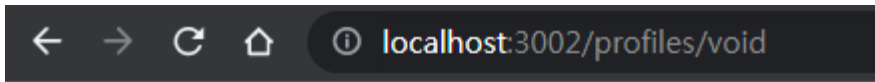
고전 소설 홍길동전의 주인공

← → ↺ 🏠 ⓘ localhost:3002/profiles/velopert

# 사용자 프로필

## 김민준

리액트를 좋아하는 개발자



# 사용자 프로필

존재하지 않는 프로필입니다.

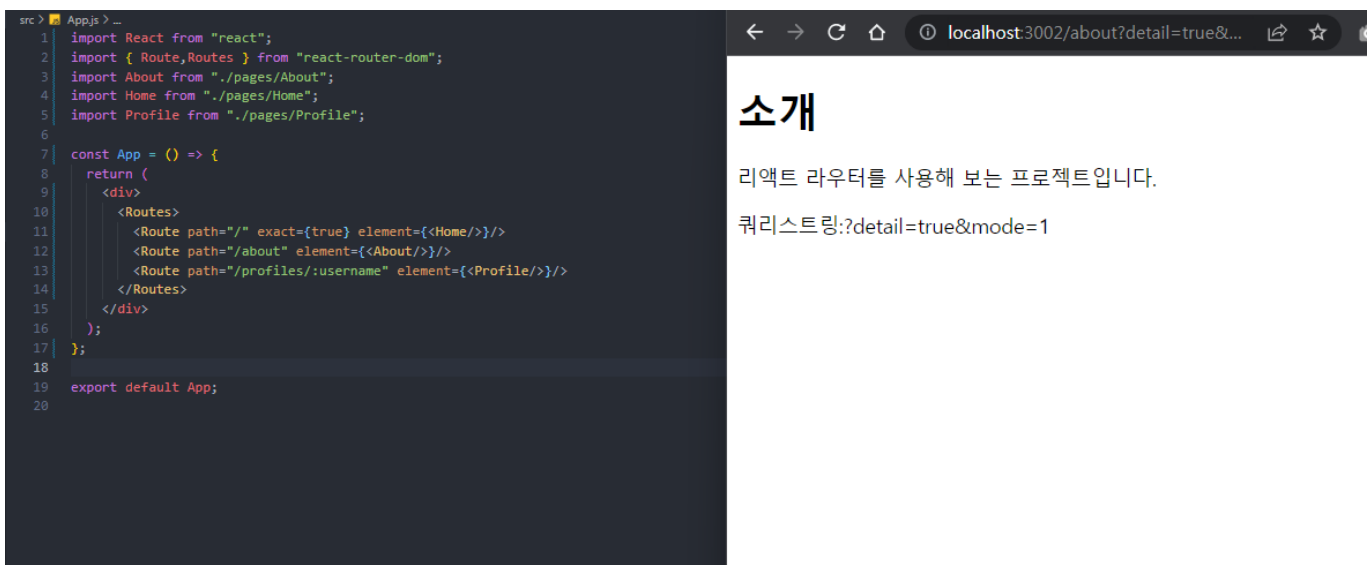
## 13.4.2 쿼리스트링

About.js

```
import { useLocation } from "react-router-dom";

const About = () => {
  const location = useLocation();
  return (
    <div>
      <h1>소개</h1>
      <p>리액트 라우터를 사용해 보는 프로젝트입니다.</p>
      <p>쿼리스트링:{location.search}</p>
    </div>
  );
};

export default About;
```



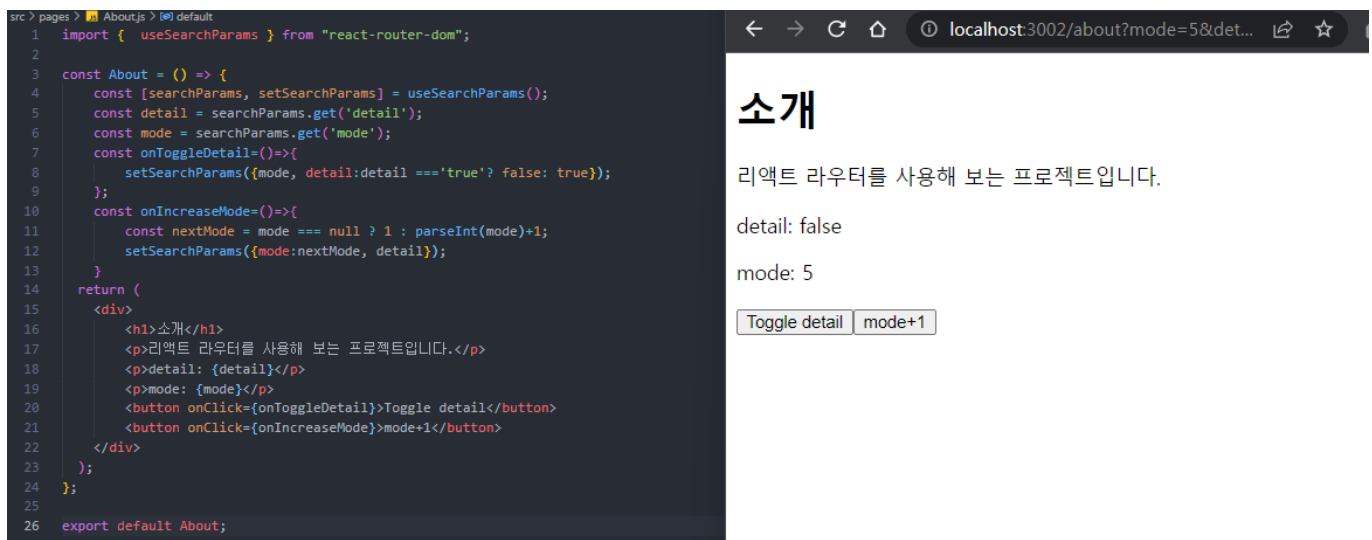
쿼리스트링 파싱하는 코드

## About.js

```
import { useSearchParams } from "react-router-dom";

const About = () => {
  const [searchParams, setSearchParams] = useSearchParams();
  const detail = searchParams.get('detail');
  const mode = searchParams.get('mode');
  const onToggleDetail={()=>{
    setSearchParams({mode, detail:detail === 'true'? false: true});
  }};
  const onIncreaseMode={()=>{
    const nextMode = mode === null ? 1 : parseInt(mode)+1;
    setSearchParams({mode:nextMode, detail});
  }}
  return (
    <div>
      <h1>소개</h1>
      <p>리액트 라우터를 사용해 보는 프로젝트입니다.</p>
      <p>detail: {detail}</p>
      <p>mode: {mode}</p>
      <button onClick={onToggleDetail}>Toggle detail</button>
      <button onClick={onIncreaseMode}>mode+1</button>
    </div>
  );
};

export default About;
```



## 13.5 중첩된 라우트

## Articles.js

```
import React from 'react'
import { Link } from 'react-router-dom'
```

```
const Articles = () => {
  return (
    <ul>
      <li>
        <Link to="/articles/1">게시글 1</Link>
      </li>
      <li>
        <Link to="/articles/2">게시글 2</Link>
      </li>
      <li>
        <Link to="/articles/3">게시글 3</Link>
      </li>
    </ul>
  );
};

export default Articles;
```

#### Article.js

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

const Article = () => {
  const {id} = useParams();
  return (
    <div>
      <h2>게시글{id}</h2>
    </div>
  );
};

export default Article;
```

```

src > pages > Home.js > Home
1  import React from 'react'
2  import { Link } from 'react-router-dom';
3
4  const Home = () => {
5    return (
6      <div>
7        <h1>홈</h1>
8        <p>가장 먼저 보여지는 페이지입니다.</p>
9
10       <ul>
11         <li>
12           <Link to="/about">소개</Link>
13         </li>
14         <li>
15           <Link to="/profiles/velopert">velopert의 프로필입니다.</Link> "velopert"
16         </li>
17         <li>
18           <Link to="/profiles/gildong">gildong의 프로필입니다.</Link> "gildong"
19         </li>
20         <li>
21           <Link to="/profiles/void">존재하지 않는 프로필</Link>
22         </li>
23         <li>
24           <Link to="/articles">게시글 목록</Link>
25         </li>
26       </ul>
27     </div>
28   );
29 };
30
31 export default Home;

```

## 홈

가장 먼저 보여지는 페이지입니다.

- [소개](#)
- [velopert의 프로필입니다.](#)
- [gildong의 프로필입니다.](#)
- [존재하지 않는 프로필](#)
- [게시글 목록](#)

## App.js

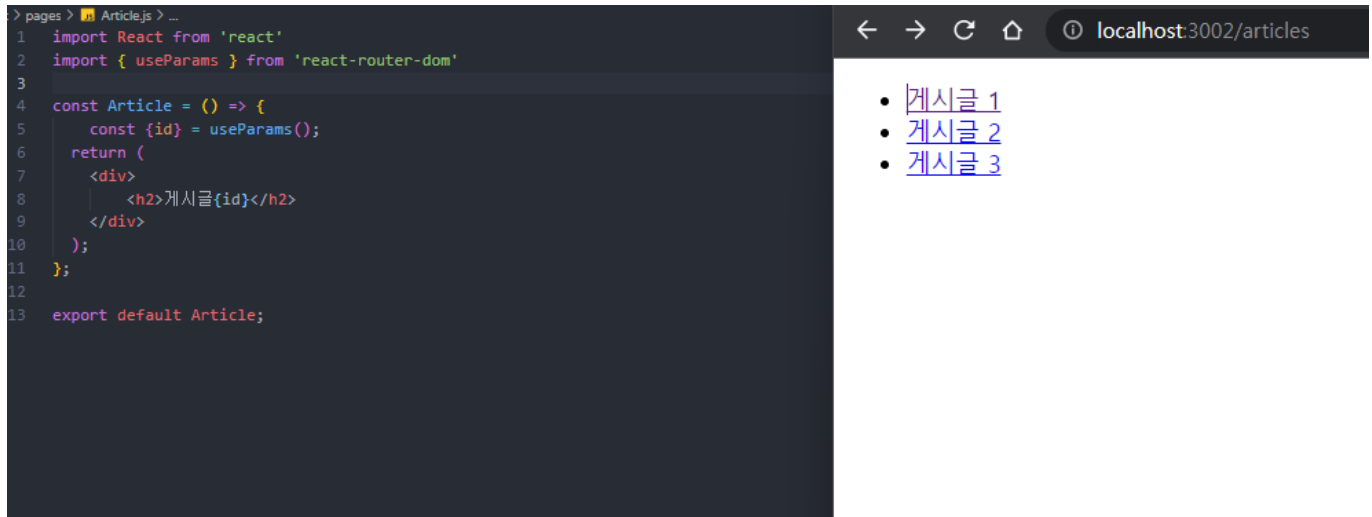
```

import React from "react";
import { Route,Routes } from "react-router-dom";
import About from "./pages/About";
import Article from "./pages/Article";
import Articles from "./pages/Articles";
import Home from "./pages/Home";
import Profile from "./pages/Profile";

const App = () => {
  return (
    <div>
      <Routes>
        <Route path="/" exact={true} element={<Home/>}/>
        <Route path="/about" element={<About/>}/>
        <Route path="/profiles/:username" element={<Profile/>}/>
        <Route path="/articles" element={<Articles/>}/>
        <Route path="/articles/:id" element={<Article/>}/>
      </Routes>
    </div>
  );
};

export default App;

```



## Home.js

```

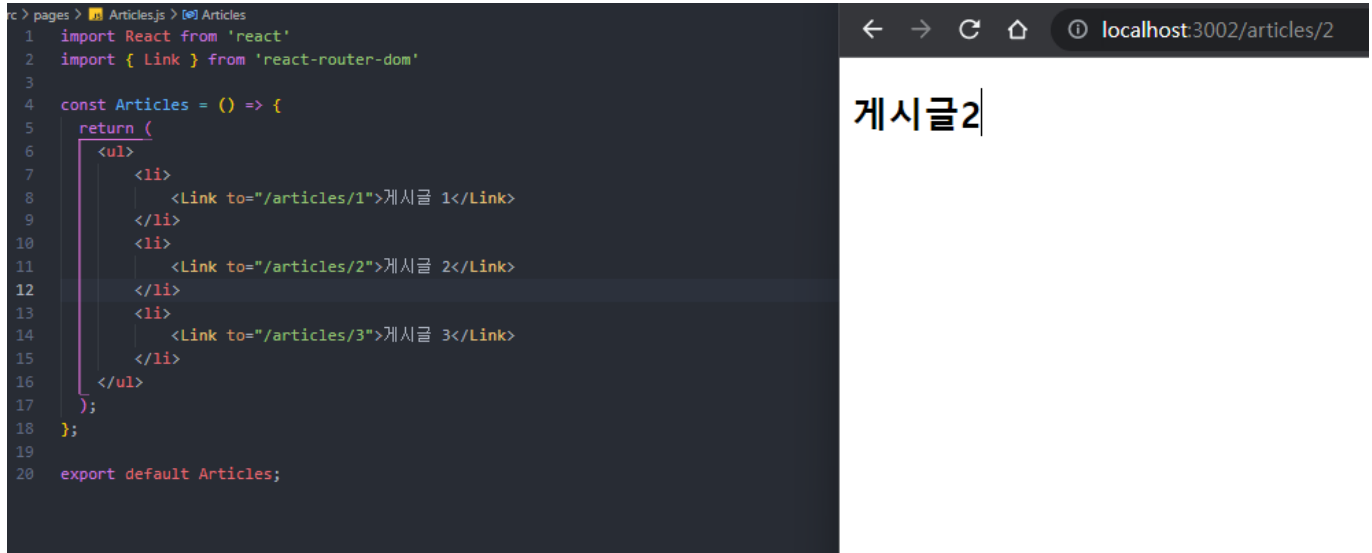
import React from 'react'
import { Link } from 'react-router-dom';

const Home = () => {
  return (
    <div>
      <h1>홈</h1>
      <p>가장 먼저 보여지는 페이지입니다.</p>

      <ul>
        <li>
          <Link to="/about">소개</Link>
        </li>
        <li>
          <Link to="/profiles/velopert">velopert의 프로필입니다.</Link>
        </li>
        <li>
          <Link to="/profiles/gildong">gildong의 프로필입니다.</Link>
        </li>
        <li>
          <Link to="/profiles/void">존재하지 않는 프로필</Link>
        </li>
        <li>
          <Link to="/articles">게시글 목록</Link>
        </li>
      </ul>
    </div>
  );
};

export default Home;

```



## 중첩된 라우트 형태로 설정

### App.js

```

import React from "react";
import { Route,Routes } from "react-router-dom";
import About from "./pages/About";
import Article from "./pages/Article";
import Articles from "./pages/Articles";
import Home from "./pages/Home";
import Profile from "./pages/Profile";

const App = () => {
  return (
    <div>
      <Routes>
        <Route path="/" exact={true} element={<Home/>}/>
        <Route path="/about" element={<About/>}/>
        <Route path="/profiles/:username" element={<Profile/>}/>
        <Route path="/articles" element={<Articles/>}>
          <Route path=:id" element={<Article/>}/>
        </Route>
      </Routes>
    </div>
  );
};

export default App;

```

### Articles.js

```

import React from 'react'
import { Link, Outlet } from 'react-router-dom'

```

```
const Articles = () => {
  return (
    <div>
      <Outlet/>
      <ul>
        <li>
          <Link to="/articles/1">게시글 1</Link>
        </li>
        <li>
          <Link to="/articles/2">게시글 2</Link>
        </li>
        <li>
          <Link to="/articles/3">게시글 3</Link>
        </li>
      </ul>
    </div>
  );
};

export default Articles;
```

```
1 import React from 'react'
2 import { Link, Outlet } from 'react-router-dom'
3
4 const Articles = () => {
5   return (
6     <div>
7       <Outlet/>
8       <ul>
9         <li>
10           <Link to="/articles/1">게시글 1</Link>
11         </li>
12         <li>
13           <Link to="/articles/2">게시글 2</Link>
14         </li>
15         <li>
16           <Link to="/articles/3">게시글 3</Link>
17         </li>
18       </ul>
19     </div>
20   );
21 };
22
23 export default Articles;
```

localhost:3002/articles/1

## 게시글1

- [게시글 1](#)
- [게시글 2](#)
- [게시글 3](#)

### 13.5.1 공통 레이아웃 컴포넌트

Layout.js

```
import React from 'react'
import { Outlet } from 'react-router-dom'

const Layout = () => {
  return (
    <div>
      <header style={{background: 'lightgray', padding:16, fontSize:24}}>
        Header
      </header>
      <Outlet/>
    </div>
  );
};
```



```
        </header>
        <main>
          <Outlet/>
        </main>
      </div>
    );
  };

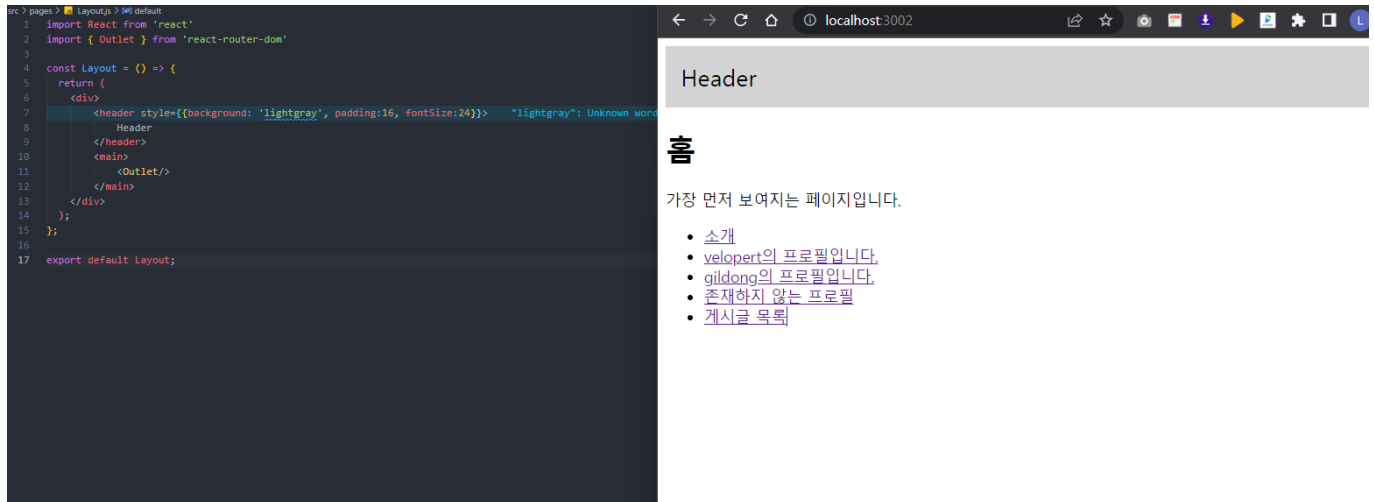
  export default Layout;
```

### App.js

```
import React from "react";
import { Route,Routes } from "react-router-dom";
import Layout from "./pages/Layout";
import About from "./pages/About";
import Article from "./pages/Article";
import Articles from "./pages/Articles";
import Home from "./pages/Home";
import Profile from "./pages/Profile";

const App = () => {
  return (
    <div>
      <Routes>
        <Route element={<Layout/>}>
          <Route path="/" exact={true} element={<Home/>}/>
          <Route path="/about" element={<About/>}/>
          <Route path="/profiles/:username" element={<Profile/>}/>
        </Route>
        <Route path="/articles" element={<Articles/>}>
          <Route path=":id" element={<Article/>}/>
        </Route>
      </Routes>
    </div>
  );
};

export default App;
```



### 13.5.2 index props

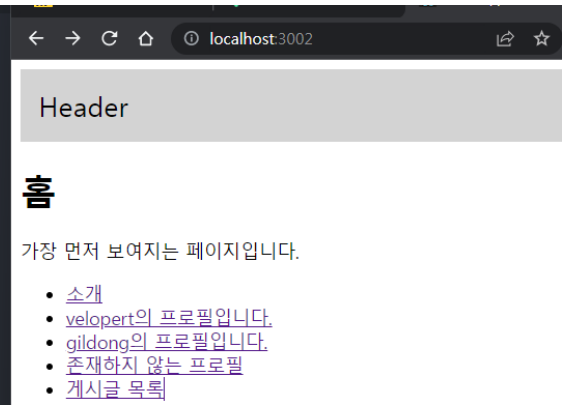
App.js

```
import React from "react";
import { Route,Routes } from "react-router-dom";
import Layout from "./pages/Layout";
import About from "./pages/About";
import Article from "./pages/Article";
import Articles from "./pages/Articles";
import Home from "./pages/Home";
import Profile from "./pages/Profile";

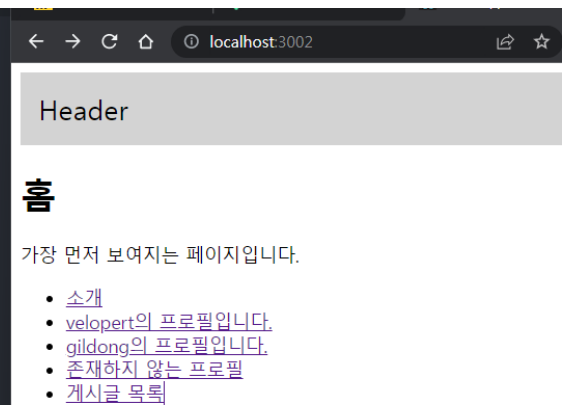
const App = () => {
  return (
    <div>
      <Routes>
        <Route element={<Layout/>}>
          <Route index element={<Home/>}/>
          <Route path="/about" element={<About/>}/>
          <Route path="/profiles/:username" element={<Profile/>}/>
        </Route>
        <Route path="/articles" element={<Articles/>}>
          <Route path=":id" element={<Article/>}/>
        </Route>
      </Routes>
    </div>
  );
};

export default App;
```

```
src > App.js ...
1 import React from "react";
2 import { Route,Routes } from "react-router-dom";
3 import Layout from "../pages/Layout";
4 import About from "../pages/About";
5 import Article from "../pages/Article";
6 import Articles from "../pages/Articles";
7 import Home from "../pages/Home";
8 import Profile from "../pages/Profile";
9
10 const App = () => {
11   return (
12     <div>
13       <Routes>
14         <Route element={<Layout/>}>
15           <Route index element={<Home/>}>
16             <Route path="/about" element={<About/>}>
17               <Route path="/profiles/:username" element={<Profile/>}>
18             </Route>
19           <Route path="/articles" element={<Articles/>}>
20             <Route path=:id" element={<Article/>}>
21           </Route>
22         </Routes>
23       </div>
24     );
25   };
26
27 export default App;
28
```



```
src > App.js ...
1 import React from "react";
2 import { Route,Routes } from "react-router-dom";
3 import Layout from "../pages/Layout";
4 import About from "../pages/About";
5 import Article from "../pages/Article";
6 import Articles from "../pages/Articles";
7 import Home from "../pages/Home";
8 import Profile from "../pages/Profile";
9
10 const App = () => {
11   return (
12     <div>
13       <Routes>
14         <Route element={<Layout/>}>
15           <Route index element={<Home/>}>
16             <Route path="/about" element={<About/>}>
17               <Route path="/profiles/:username" element={<Profile/>}>
18             </Route>
19           <Route path="/articles" element={<Articles/>}>
20             <Route path=:id" element={<Article/>}>
21           </Route>
22         </Routes>
23       </div>
24     );
25   };
26
27 export default App;
28
```



## 13.6 리액트 라우터 부가 기능

### 13.6.1 useNavigate

useNavigate는 Link 컴포넌트를 사용하지 않고 다른 페이지로 이동해야 하는 상황에 사용하는 Hook입니다.

Layout.js

```
import React from 'react'
import { Outlet, useNavigate } from 'react-router-dom'

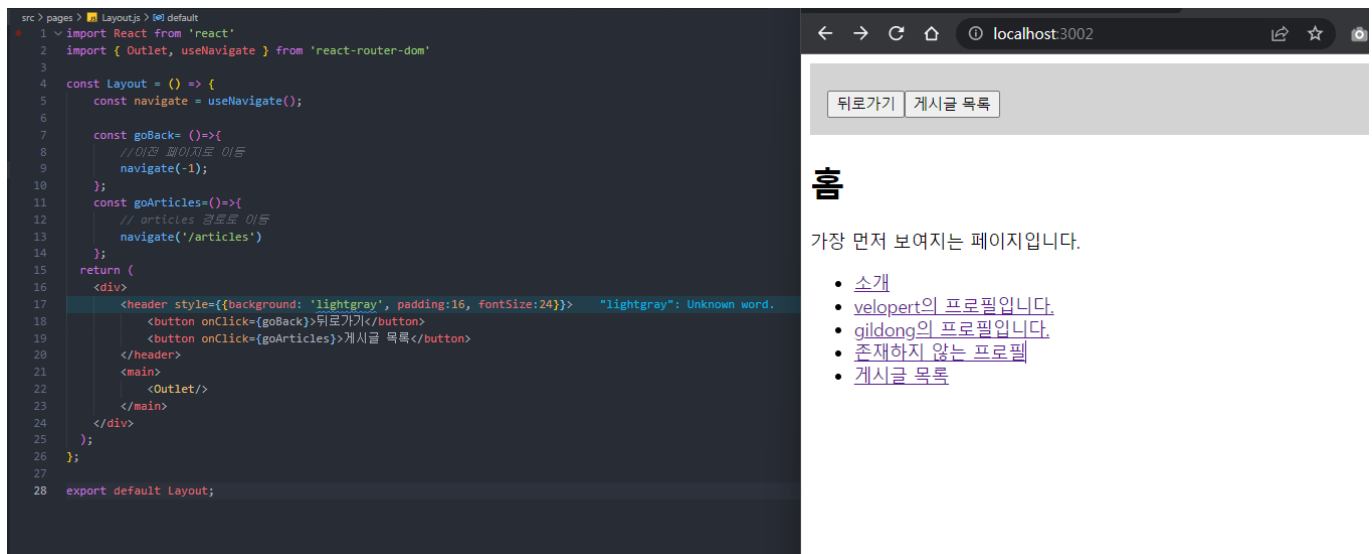
const Layout = () => {
  const navigate = useNavigate();
```

```

const goBack= ()=>{
  //이전 페이지로 이동
  navigate(-1);
};
const goArticles=()=>{
  // articles 경로로 이동
  navigate('/articles')
};
return (
  <div>
    <header style={{background: 'lightgray', padding:16, fontSize:24}}>
      <button onClick={goBack}>뒤로가기</button>
      <button onClick={goArticles}>게시글 목록</button>
    </header>
    <main>
      <Outlet/>
    </main>
  </div>
);
};

export default Layout;

```



## replace 옵션 사용

### Layout.js-goArticles

```

const goArticles=()=>{
  // articles 경로로 이동
  navigate('/articles', {replace: true});
  // 홈에서 다른 페이지 이동 후, 뒤로가기를 눌렀을 때 전 페이지가 아닌 홈으로 돌아간
  다.
};

```

## 13.6.2 NavLink

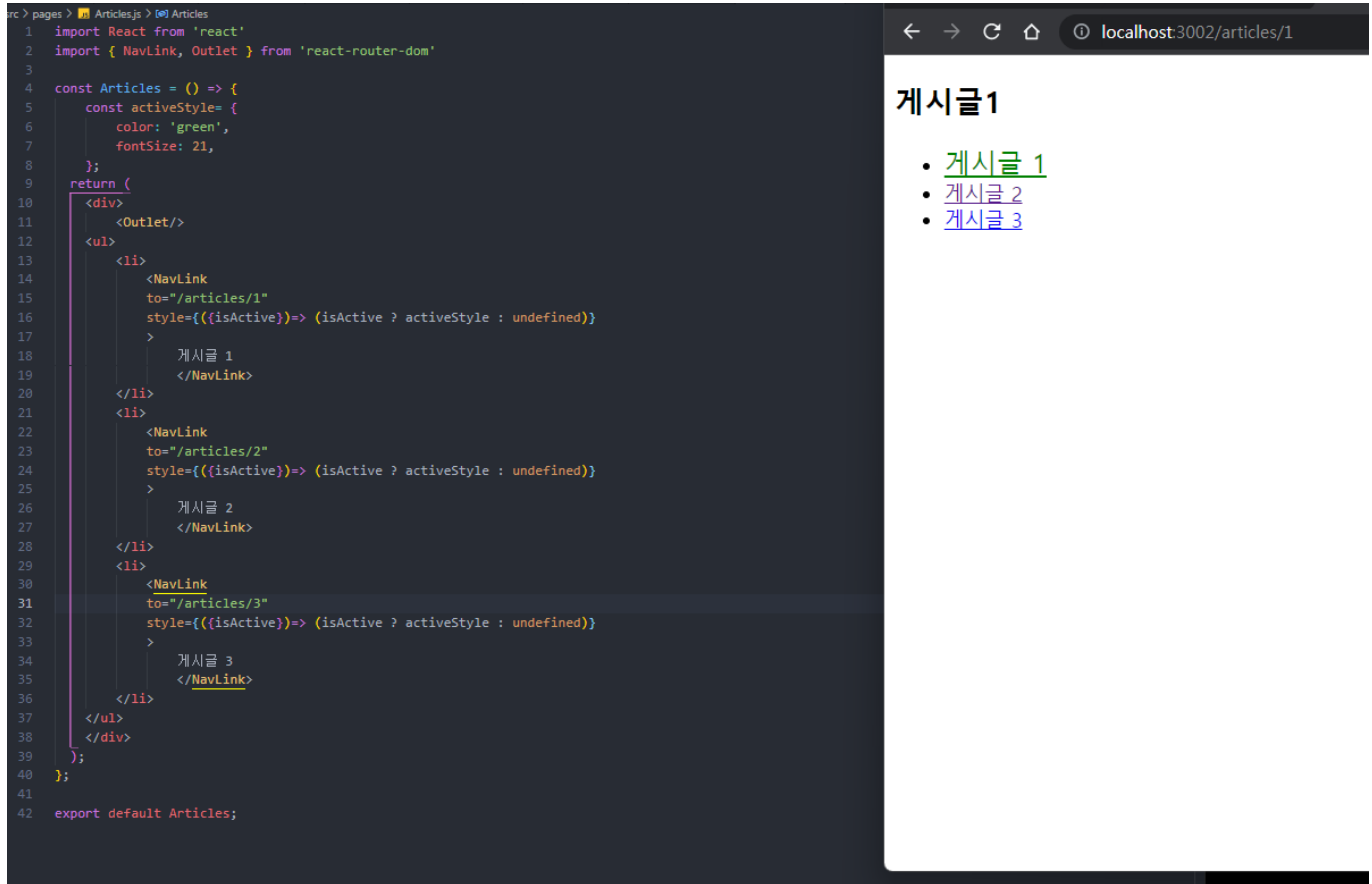
NavLink 컴포넌트는 링크에서 사용하는 경로가 현재 라우트 경로와 일치하는 경우 특정 스타일 또는 CSS 클래스를 적용하는 컴포넌트 입니다.

Articles.js

```
import React from 'react'
import { NavLink, Outlet } from 'react-router-dom'

const Articles = () => {
  const activeStyle= {
    color: 'green',
    fontSize: 21,
  };
  return (
    <div>
      <Outlet/>
      <ul>
        <li>
          <NavLink
            to="/articles/1"
            style={({isActive})=> (isActive ? activeStyle : undefined)}
          >
            게시글 1
          </NavLink>
        </li>
        <li>
          <NavLink
            to="/articles/2"
            style={({isActive})=> (isActive ? activeStyle : undefined)}
          >
            게시글 2
          </NavLink>
        </li>
        <li>
          <NavLink
            to="/articles/3"
            style={({isActive})=> (isActive ? activeStyle : undefined)}
          >
            게시글 3
          </NavLink>
        </li>
      </ul>
    </div>
  );
};

export default Articles;
```



## NavLink로 리팩터링 하기

### Articles.js

```

import React from 'react'
import { NavLink, Outlet } from 'react-router-dom'

const Articles = () => {
  return (
    <div>
      <Outlet/>
      <ul>
        <ArticleItem id={1}/>
        <ArticleItem id={2}/>
        <ArticleItem id={3}/>
      </ul>
    </div>
  );
};

const ArticleItem = ({id})=>{
  const activeStyle= {
    color: 'green',
    fontSize: 21,
  };
  return (
    <li>

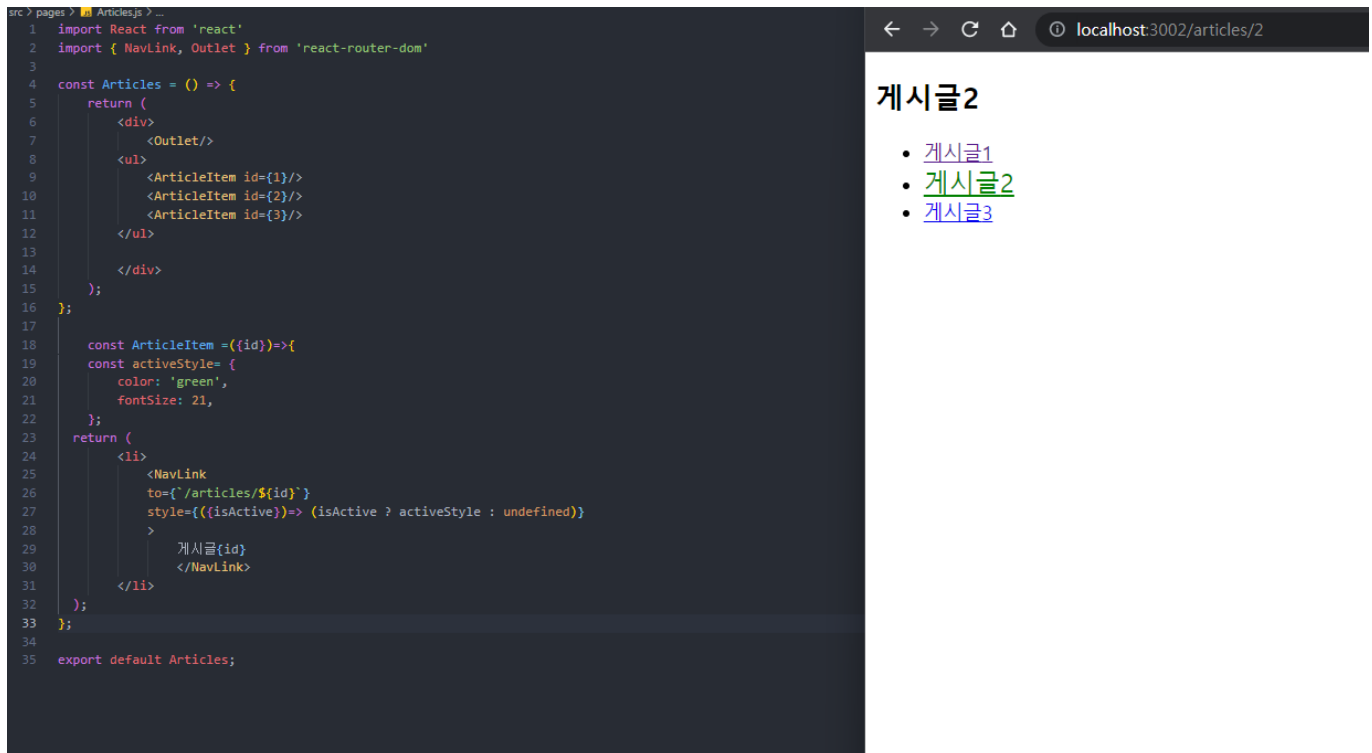
```

```

      <NavLink
        to={` /articles/${id}`}
        style={({isActive})=> (isActive ? activeStyle : undefined)}
      >
        게시글{id}
      </NavLink>
    </li>
  );
};

export default Articles;

```



### 13.6.3 NotFound 페이지 만들기

이 페이지는 사정에 정의되지 않은 경로에 사용자가 진입했을 때 보여는 페이지입니다. 즉 페이지를 찾을 수 없을 때 나타나는 페이지 입니다.

NotFound.js

```

import React from 'react'

const NotFound = () => {
  return (
    <div
      style={{
        display: 'flex',
        alignItems: 'center',
        fonstSize: 64,
        position: 'absolute',
        width: '100%',

```

```

        height: '100%',
      }}
    >
      404
    </div>
  );
};

export default NotFound;

```

### App.js

```

import React from "react";
import { Route,Routes } from "react-router-dom";
import Layout from "./pages/Layout";
import About from "./pages/About";
import Article from "./pages/Article";
import Articles from "./pages/Articles";
import Home from "./pages/Home";
import Profile from "./pages/Profile";
import NotFound from "./pages/NotFound";

const App = () => {
  return (
    <div>
      <Routes>
        <Route element={<Layout/>}>
          <Route index element={<Home/>}/>
          <Route path="/about" element={<About/>}/>
          <Route path="/profiles/:username" element={<Profile/>}/>
        </Route>
        <Route path="/articles" element={<Articles/>}>
          <Route path=":id" element={<Article/>}/>
        </Route>
        <Route path="*" element={<NotFound/>}/>
      </Routes>
    </div>
  );
};

export default App;

```

### 13.6.4 Navigate 컴포넌트

Navigate 컴포넌트는 화면에 보여주는 순간 다른 페이지로 이동을 하고 싶을 때 사용하는 컴포넌트 입니다. 즉, 페이지를 리다이렉트 하고 싶을 때 사용합니다.

### Login.js



```
import React from 'react'

const Login = () => {
  return (
    <div>로그인 페이지</div>
  );
};

export default Login;
```

### MyPage.js

```
import React from 'react'
import { Navigate } from 'react-router-dom'

const MyPage = () => {
  const isLoggedIn = false;

  if(!isLoggedIn){
    return<Navigate to="/login" replace={true}/>
  }
  return <div>마이 페이지</div>
};

export default MyPage;
```

### App.js

```
import React from "react";
import { Route,Routes } from "react-router-dom";
import Layout from "./pages/Layout";
import About from "./pages/About";
import Article from "./pages/Article";
import Articles from "./pages/Articles";
import Home from "./pages/Home";
import Profile from "./pages/Profile";
import NotFound from "./pages/NotFound";
import MyPage from "./pages/MyPage";
import Login from "./pages/Login";

const App = () => {
  return (
    <div>
      <Routes>
        <Route element={<Layout/>}>
          <Route index element={<Home/>}/>
          <Route path="/about" element={<About/>}/>
          <Route path="/profiles/:username" element={<Profile/>}/>
        </Route>
      </Routes>
    </div>
  );
};
```

```
    <Route path="/articles" element={<Articles/>}>
      <Route path=":id" element={<Article/>}/>
    </Route>
    <Route path="/login" element={<Login/>}/>
    <Route path="/mypage" element={<MyPage/>}/>
    <Route path="*" element={<NotFound/>}/>
  </Routes>
</div>
);
};

export default App;
```

```
1 import React from 'react'
2 import { Navigate } from 'react-router-dom'
3
4 const MyPage = () => {
5   const isLoggedIn = false;
6
7   if(!isLoggedIn){
8     return<Navigate to="/login" replace={true}/>
9   }
10  return <div>마이 페이지</div>
11 };
12
13 export default MyPage;
```

localhost:3002/login

로그인 페이지