

React Router DOM

2023.06.30.

01. React Router DOM

02. Link

01. React Router DOM

2023.06.30.

1. React Router DOM

- Routing : 네트워크에서 경로를 선택하는 프로세스
- SPA(Single Page Application) 이점을 살려 MPA(Multi Page Application)환경과 달리 페이지 자체를 반환하지 않고 데이터만 불러와 렌더링해주기 때문에 사용자 입장에서 훨씬 빠르다고 느낄 수 있다.

2. 준비

- command 입력 창

```
npm install react-router-dom
```

3. Router

3.1 <BrowserRouter> : URL을 사용하여 브라우저의 주소 표시줄에 현재 위치를 저장하고 브라우저의 내장된 기록 스택을 사용하여 탐색 함.

- <BrowserRouter> 컴포넌트를 리액트 라우터의 도움을 받고 싶은 최상위 <App>컴포넌트를 감싸게 합니다.

```
import { BrowserRouter } from 'react-router-dom';

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

- 경로에 맞추어 해당 컴포넌트를 표현 : <Routes>컴포넌트로 감싸고 <Route>컴포넌트로 적용한다. <Routes>컴포넌트는 <Route>컴포넌트만 가능하며, 적용할 컴포넌트는 <Route> 컴포넌트의 element 프로퍼스로 적용한다.

구문

```
import { Route, Routes } from 'react-router-dom';

<Routes>
  <Route path='경로' element={<표시할 컴포넌트 />} />
  ...
</Routes>
```

예시

```
<ul>
  <li><a href='/'>Home</a></li>
  <li><a href='/topics'>Topics</a></li>
  <li><a href='/contact'>Contact</a></li>
</ul>

<Routes>
  <Route path='/' element={<Home />}/>
  <Route path='/topics' element={<Topics />}/>
  <Route path='/contact' element={<Contact />}/>
</Routes>
```



Hello React Router DOM

- [Home](#)
- [Topics](#)
- [Contact](#)

Topics

Topics...

02. Link

2023.07.06.

1. Link

- 사용자가 클릭 하거나 탭하여 다른 페이지로 이동할 수 있게 해 주는 요소
구문

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

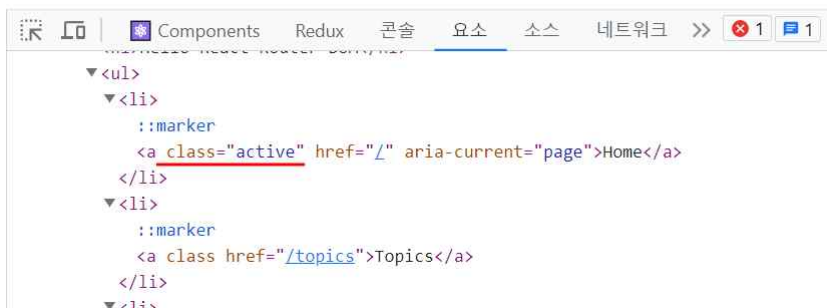
<ul>
  <li><Link to="경로">Home</Link></li>
</ul>
```

2. NavLink

- 링크가 활성(active)인지 보류(pending)인지를 아는 특별한 종류의 <Link>
구문

```
import { NavLink } from 'react-router-dom';

<ul>
  <li><NavLink to="경로">Home</NavLink></li>
</ul>
```



03. 중첩 라우팅

2023.07.06.

하위 요소에 라우팅 적용하기

① 부모 라우터의 `path` 프로퍼시에 `*`를 추가하여 다른 요소가 붙어 라우터로 이동한 페이지 내부에서 다시 다른 컴포넌트를 렌더링 함을 알려 주어야 한다.

예시

```
<Routes>
  <Route path="/" element={<Home />}/>
  <Route path="/topics/*" element={<Topics />}/>
  <Route path="/contact" element={<Contact />}/>
</Routes>
```

② 자식 라우터에서는 부모 라우터의 경로는 생략하고, 페이지 내에서 `deep`하게 이동할 자식 라우터의 경로만 작성하면 된다.

예시

```
<ul>
  <li><NavLink to="1">HTML</NavLink></li>
  <li><NavLink to="2">JS</NavLink></li>
  <li><NavLink to="3">React</NavLink></li>
</ul>

<Routes>
  <Route path="1" element={<HTML />}/>
  <Route path="2" element={<JS />}/>
  <Route path="3" element={<React />}/>
</Routes>
```