

리액트 라우터

Install

```
npm install react-router-dom
# or
yarn add react-router-dom
```

BrowserRouter

리액트 라우터를 적용할 때는 `react-router-dom` 에 내장되어있는 `<BrowserRouter>` 라는 컴포넌트를 사용하여 감싸면 됩니다.

이 컴포넌트는 HTML5의 History API를 사용하여 페이지를 새로고침 하지 않고도 주소를 변경하고, 현재 주소에 관련된 정보를 props로 쉽게 조회하거나 사용할 수 있습니다.

Route

`<Route>` 컴포넌트를 사용하면 어떤 규칙을 가진 경로에 어떤 컴포넌트를 보여줄지 정의할 수 있습니다.

`exact` 라는 props를 사용하면 중복된 규칙을 방지할 수 있습니다.

Use

```
import React from 'react';
import { BrowserRouter, Route } from 'react-router-dom';
import Main from './Main';
import Sub from './Sub';

const Router = () => {
  <BrowserRouter>
    <Route path="/" component={Main} exact={true} />
    <Route path="/sub" component={Sub} />
  </BrowserRouter>
}
```

Link

`<Link>` 컴포넌트는 클릭하면 다른 주소로 이동시켜주는 컴포넌트입니다. HTML의 `<a>` 태그와 유사하지만 리액트 라우터를 사용할때에는 `<a>` 태그를 사용하면 안됩니다. 페이지를 전환하면서 페이지를 새로 불러오기 때문에 리액트 내에서 들고 있던 정보들을 모두 잃어버리게 되어 렌더링된 컴포넌트들도 모두 사라지고 처음부터 렌더링하게 됩니다.

`<Link>` 컴포넌트를 사용하면 페이지를 새로 불러오지 않고 HTML5 History API를 사용하여 페이지의 주소만 변경하고, 페이지 전환을 방지하는 기능이 내장되어 있습니다.

Use

```
import React from 'react';
import { Route, Link } from 'react-router-dom';
import Main from './Main';
import Sub from './Sub';
...
return (
  <div>
    <ul>
      <li><Link to="/">메인</Link></li>
      <li><Link to="/Sub">서브</Link></li>
    </ul>
    <Route path="/" component={Main} exact={true} />
    <Route path="/sub" component={Sub} />
  </div>
)
```

Switch

`<Switch>` 컴포넌트로 여러 `<Route>` 컴포넌트를 감싸면 그 중 일치하는 라우트만 렌더링해 줍니다. 일치하는 규칙이 없을 경우 "페이지를 찾을 수 없습니다." 와 같은 것을 구현할 수 있습니다.

Use

```
import React from 'react';
import { BrowserRouter, Route, Switch } from 'react-router-dom';
import Main from './Main';
import Sub from './Sub';
import PageNotFound from './PageNotFound';

const Router = () => {
  <BrowserRouter>
    <Switch>
      <Route path="/" component={Main} exact={true} />
      <Route path="/sub" component={Sub} />
    </Switch>
  </BrowserRouter>
}
```

```
// 페이지를 찾을 수 없습니다.  
<Route component={PageNotFound} />  
</Switch>  
</BrowserRouter>  
}
```

더 자세한 내용 및 예제코드를 보고자 한다면 다음 링크를 참고하면 좋습니다.

[REACT-TRAINING : REACT-ROUTER](#)