GatsBy Routing

#TIL/dev/web/staticGen

`src/pages`에 컴포넌트 다 때려넣고 작업하니까 yarn start는 정상적으로 동작하나 yarn build에서는 계속 빌드를 실패했다. yarn build는 빌드 시점에 모든 페이지들을 렌더링해야해서 실패하는 듯.
혼자서는 너무 삽질만할 것 같아서 민식띠한테 도움을 요청하고 함께 디버깅했는데 엄청 빠르게 원인을 파악할 수 있었다. pages 안에 있는 리액트 컴포넌트들은 GatsBy Core가 자동으로 routing을 생성해줘서 생긴 이슈였다. 에러 로그를 보고 이런 것들을 유추해낸 민식띠가 정말 대단한 듯.

사용법

GatsBy의 라우팅은 일반 React 프로젝트의 react-router를 쓰는 방식과 다르다. GatsBy는 페이지를 만들기 위해 3가지 방법을 제공한다.

1. gatsby-node.js가 제공하는 createPages API를 사용하는 것.
2. `src/pages`에 컴포넌트를 추가하면 GatsBy Core가 자동으로 라우팅을 생성해줌.
3. Plugin 사용하기

GatsBy가 pages 에 있는 리액트 컴포넌트에 대한 페이지를 만들때 자동으로 path까지 생성해준다!
➜ src/pages 안의 js 파일들은 각각의 page를 구성한다. path는 file structure에 매핑된다. 예를 들면, `src/pages/home.js`의 path는 `yourwebsite.com/home`이 된다! 다만, index.js는 예외인데, Index.js의 경우 이것이 포함된 directory까지를 경로로 갖는다.

Routing

Routing 할때 Props를 전달하는 방법

Link의 state props를 사용하면된다.

```
<Link to={`/photos/${id}`} state={{ fromFeed: true }} > View Photo </Link>



const Photo = ({ location, photoId }) => {
  if (location.state.fromFeed) {
```

```
    return <FromFeedPhoto id={photoId} />
  } else {
    return <Photo id={photoId} />
  }
}
```

근데 이거 typescript에서 사용하게 될때, Photo의 props interface를 어떻게 정의해야할지모르겠다. 일단 지금은 dynamic하게 props를 전달할 필요가 없을것같아서 각각의 페이지별로 컴포넌트를 생성했다. 나중에 필요가 생기면 그때 더 공부해보면 될듯

Gatsby Link API