Sec01

Creative-gh 간략 요약

작성자: 강태경



# 실전 코딩

목적 : 실무에서는 웹 디자이너와 협업으로 프로젝트가 이루어지기도 함. 웹 디자이너가 html /css 코딩한것을 리액트/벡엔드 코딩을 추가하여 완성함, 그것을 대비하여 디자이너가 준 소스를 부트스트랩 테마로 대신하여 (html/css/js) 리액트 및 스프링 부트 코딩을 연습하는 챕터임

1. : creative-gh
2. 설치패키지

* # 1) 메뉴 라이브러리 설치  
  npm i react-router-dom  
  npm i @types/react-router-dom
* # 2) 벡엔드 연동 라이브러리 설치  
  npm i axios
* # 3) pre css 컴파일러 : sass  
  npm i sass
* # 4) Material Page component 업그레이드

npm i @mui/material @emotion/react @emotion/styled

* # 4-1) 소스에서 임포트 사용법 : <Pagination />

import Pagination from '@mui/material/Pagination';

* # 5) 타입 설치 : jquery, jqueryui type 넣기

npm i --save-dev @types/jquery

npm i --save-dev @types/jqueryui

* 6) npm install simplelightbox 설치

npm install simplelightbox

* 6-1) import SimpleLightbox from "simplelightbox" 설치

// scripts.js 파일에 import

import SimpleLightbox from "simplelightbox";

// App.tsx 또는 Home.tsx 에 import

import "simplelightbox/dist/simple-lightbox.css";

* # 외부 라이브러리 설치시 타입을 (@types) 설치해야 타입스크립트 에러가 발생하지 않음 , 하지만 타입스크립트가 최근에 유행하는 코딩방법이라서 외부 라이브러리에서 지원않할 수 있음. 이때 임시로 타입에러를 방지하는 방법임

# \* 외부 라이브러리 타입이 없을 경우 처리 : 타입 설정

* # 1) tsconfig.json 파일 - compilerOptions 속성에 아래 추가

"typeRoots": ["./types", "./node\_modules/@types"], // 보통 types 폴더를 만들어 타입 정의

* # 2) types/외부라이브러리명/index.d.ts 파일 생성 후 아래 추가

declare module '외부라이브러리명';

1. 부트스트랩 테마(html 기반 테마)를 리액트로 변환하는 일반적인 순서 :   
   (이 방법은 모든 테마에 적용되지는 않을 수 있으며 추가 설정과 조정이 필요한 테마가 있을 수 있음)

* Index.html 작업

1. 원본 테마를(creative-gh) 선정하고 다운로드
2. 적절한 폴더에 리액트 프로젝트(frontend-typescript) 생성
3. creative-gh의 img, css, js 폴더 -> 리액트의 src/assets 밑에 같은 이름으로 복사
4. 원본 테마(creative-gh) 의 index.html 에 head 태그 부분에 css 링크들을 -> 리액트의 public/index.html 에 똑같이 복사
5. Styles.css 링크 : 외부 라이브러리가 아님, 테마 개발자가 제작한 css 파일임  
   => src/assets/css 로 3) 에서 이동 복사 했으므로 여기서는 삭제 및 주석으로 막음  
   => bootstrap.js 링크도 삭제(사용 않 할 예정)  
   => SimpleLightbox plugin CSS 링크 제거 : npm 설치 및 import 예정
6. 원본 테마(creative-gh) 의 index.html 에 body 태그 맨 밑에 js 링크 삭제 : script 태그로 작성된 scripts.js 링크 삭제, 그외 부트스트랩 js 링크도 삭제(필요없음)  
   => 리액트 폴더 : src/assets/js 로 3) 에서 이동되었음 (필요없음)  
   => SimpleLightbox plugin js 링크 제거 : npm 설치 및 import 예정

* Scripts.js 파일 작업(테마 개발자가 만든 js)

1. Window.addEventLisener() 함수 제거 : 화면이 뜰때 실행되는 바닐라 함수로 react 의 useEffect() 함수로 작성할 예정
2. Export default function initMain() { 기존소스} : 기존소스에 임의의 함수의 실행 블럭({})으로 실행되게 하고 다른 js 파일에서 import 할 수 있게 export 기능을 추가
3. Home.tsx 에 useEffect() 함수를 작성하고 initMain() 함수를 실행함

* Scripts.css 파일 작업(테마 개발자가 만든 css)

1. App.tsx 또는 Home.tsx 에 css import 함
2. 이미지 경로가 원본과 리액트 폴더 경로가 달라 에러가 발생할 수 있음 css 파일이 있는 위치를 토대로 조정해야함

* Home.tsx : 이미지 경로 조정 , a 태그 및 img 태그의 src 에 이미지 경로가 리액트로 변경되면 조정되어야 함 : require() 함수 이용 또는 import 이용해서 적용해야함

1. 부트스트랩 테마가 리액트로 변경되었으며 적절하게 홈페이지를 변경하고(이미지, 글 등 수정) 벡엔드 게시판 일부를 테마에 적용함

* Home.tsx 에 원본 테마 body 쪽 html 부분을 복사해서 붙여넣기함

1. 원본 테마 body 쪽 html 부분을 복사해서 붙여넣기함
2. Html 부분에서 리액트로 전환시 주의점
3. Class -> className 변경
4. For -> htmlFor 변경
5. Style = “” => style ={속성: 값 + “단위”} 형태로 변경
6. 리액트 내에 <script> 태그 포함되면 안됨 : 제거