**#AJAX를 위한 Axios설정**

**npm axios install**

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명

**데이터를 가져오는 부분은 분리해서 개발하는 것이 좋음. api 폴더 생성**

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

1. **host 주소 설정 -> AWS 업로드시 API\_SERVER\_HOST 만 변경하면 되도록 설계**

텍스트, 스크린샷, 폰트, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

1. **aynsc await 비동기 처리와 axios 통신을 이용해서 api로부터 데이터를 받아오는 함수들을 만듬**

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**확장성 때문에 파라미터를 여러개 받을 때는 객체로 받는 것이 좋다**

**axios 통신에서 쿼리스트링의 쿼리 파라미터들을 받아오는 방법이다.**

**리턴은 promise 객체임으로 나중에 꺼내서 써야함.**

**###개념숙지 useEffect : 비동기 통신 시 특정 조건 만족 시에만 호출하는 것**

**리액트 컴포넌트는 상태가 변경되면 자동으로 렌더링된다.**

비동기 처리 이므로 나중에 다시 바뀌는 일이 생긴다 -> 다시 렌더링되면서 또 다시 비동기 처리 -> 또 다시 렌더링 =>무한반복

이것을 해결하기 위해서 useEffect를 사용한다.

**#조회기능 구현하기**

실제 동작하는 기능은 컴포넌트에 만들것이므로 새 폴더 todo 생성

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

tno 값에 변경만 렌더링 되도록 하기 위해서 useEffect를 사용한다.

todoApi에서 만들었던 함수 getOne을 통해서 api 통신을 하여 결과로 받은 promise 객체를 console에 찍어보고 setTodo를 통해 초기화한다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

조회한 결과를 표시할 페이지인 ReadPage이다 이전에 작성했던 수정버튼과 조회버튼을 잠시 삭제하고 api통신의 결과를 화면에 반영하는 것을 보기위해 간단하게 수정하였다.

**#API 서버와의 통신**

스프링 백엔드 서버를 실행하고 프론트 서버를 실행하여 localhost:3000/todo/read/123 을 통해 페이지에 접근하면

axios를 통해 백엔드 서버 localhost:8080/api/todo/123으로 get 요청이 잘 날아감을 볼 수 있다.

혹시나 get요청이 두번 날아간다면 프로젝트 index.js에 StrictMode 삭제하지 않은 것임

텍스트, 스크린샷, 소프트웨어, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

**#화면 UI 개선**

**makeDiv jsx 추가하고**

텍스트, 멀티미디어 소프트웨어, 소프트웨어, 그래픽 소프트웨어이(가) 표시된 사진

자동 생성된 설명

return 부분 수정하기.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명

아래는 아까 ReadPage.js에서 tailwindcss 오타난거 수정해줌.

텍스트, 스크린샷, 멀티미디어 소프트웨어, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 번호, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

writer 같은 경우는 최초에 엔티티 만들 때 해당 컬럼 추가 안했었음. 추가하겠음.

추가함. 엔티티클래스에 writer컬럼 추가/ dto클래스에 writer 변수 추가/ 서비스계층 dto<->entity 변환 메서드 두개의 빌더를 통한 조립메서드에 writer부분 추가.

텍스트, 스크린샷, 소프트웨어, 번호이(가) 표시된 사진

자동 생성된 설명