

React-query-2

☰ 태그	
📅 날짜	@2023년 5월 15일

coination 프로젝트를 진행하면서 사용한 것을 토대로 작성하겠다.

기본구조

상위 컴포넌트

```
import { QueryClient, QueryClientProvider } from "react-query";

const queryClient = new QueryClient();

const Rq = ()=>{

  return(
    <QueryClientProvider client={queryClient} >
      <DataChart />
      <Data />
    </QueryClientProvider>
  )
}
```

QueryClientProvider : 리액트 쿼리를 사용하기 위한 Provider로, 리액트 쿼리를 감싸주는 역할을 한다. Provider 안에 선언된 내용은 Query Client에 접근이 가능하다.

하위 컴포넌트

```
const {isLoading, error, data, isFetching} = useQuery(QueryKey , QueryFn)

-----

const Data = () => {

  const Coin_Data = "data";
```

```
const {isLoading, error, data, isFetching} = useQuery(
  Coin_Data, () => {
    return fetch(`https://api.bithumb.com/public/ticker/ALL_KRW`)
      .then((res) => res.json())
      .then((res) => res.data) // return값이 data에 담기게 된다.
    }
  , { //options
    enabled: !!Coin_Data, // 해당 변수가 있을 때만 요청을 보낸다
    refetchInterval: 1000 // 1초마다 갱신
  }
);

if(isLoading){
  return <h1>로딩중!</h1>;
}
if(error){
  return <h1>에러 발생!</h1>;
}
}
```

useQuery는 두가지 인수를 받는다.

useQuery("QueryKey", fetcher함수)

첫번째 인자 queryKey는 임의의 고유 식별자를 넣어주면 된다고한다.

고유 식별자가 있으면, 그 변수 값을 적어주고, 나는 Coin_Data라는 변수에 "data"를 담아서 사용하였다.

두번째 인자는 QueryFn(fetcher함수) 인데, Promise객체를 반환하는 함수, api로 url을 불러올 함수를 작성한다.

두번째 인자에 사용될 수 있는 ajax 라이브러리로는 fetch, axios 등이 있다.

- queryKey는 useQuery가 캐싱을 관리 할 수 있도록 도와주는 유일한 키라고 한다.

동일한 queryKey의 값은 다시 서버에 요청을 보내지 않는다. 맨 처음의 요청의 결과, 이미 캐싱된 값을 사용한다.

추가적으로 **세번째 인자로 options**이 들어갈 수 있다.

위와 같이 세번째 인자로 `refetchInterval`를 설정하여 실시간으로 데이터를 계속 받아올 수 있다.

- 만약 데이터가 로딩중이라면, `isLoading/isFetching`이 `true`가 된다.
- `isLoading`은 캐시가 없을 때, `isFetching`은 캐시 유무에 상관없이
 - `error`가 발생하면, 그 정보가 `error`에 담긴다.
 - `fetch`가 성공하면 그 데이터는 `data`에 담기게 된다.
 - `enabled` 옵션이 있으면, 해당 변수가 `true`인 경우에만 `useQuery`가 실행된다.

※ 서버에서 단순히 데이터를 조회하는 것이 아니라, 데이터의 추가/수정/삭제가 필요할 땐, `useQuery`가 아니라 `useMutation`을 사용한다.

이전에는 빗썸 api를 통해 가져온 데이터를 `useState`에 저장하여 사용하는 프로젝트였는데, 버튼을 눌러야만 저장이 가능했고 렌더링 되는 시점에서 바로 보여지지 않는 불편함이 있어서 `useQuery`를 사용하게 되었다.