React Query

The MISSING
Data-fetching
Library for React

What?

React Query is a library for fetching and caching data in React applications.

It provides a declarative API for fetching, storing, and retrieving data. React Query is designed to be used with React Hooks.

Why?

It can help make your application more responsive by caching data and automatically refetching it when necessary.

React Query can improve the performance of your application by batching requests and avoiding unnecessary network traffic.

Makes it easier to handle errors by providing a builtin error handling mechanism.

Without React Query

```
    const [isLoading, setIsLoading] = useState(false)
    const [isError, setIsError] = useState(false)
    const [fetchedData, setFetchedData] = useState([]);
    useEffect(() => {
        (async function () {
            setIsLoading(true)
            try {
                const res = await axios.get("https://some-url/api/data-
route");
                setIsError(false)
                setFetchedData(res.data.data)
            } catch (error) {
                setIsError(true)
                setFetchedData([])
            }
            setIsLoading(false)
        })();
    }, []);
```

With React Query

```
const { isLoading, error, data, isFetching } = useQuery("some-data", ()
=>axios.get(
    "https://some-url/api/data-route"
  ).then((res) => res.data)
);
```