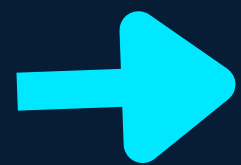


Still using **useEffect()** for data fetching in React?

Let's talk about
how **React Query**
can make your life
easier!



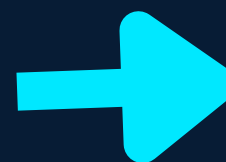
Why Not Use `useEffect()`?

- Repeated logic for **loading**.
- Manually handling caching, re-fetching, and **data synchronization**.
- Complex state management for **API** calls, **error**, and **success states**.



Enter **React** **Query**

React Query **simplifies data**
fetching by automating
much of this logic.

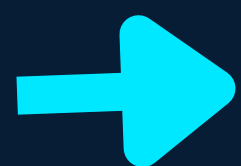


Key **Benefits** of React Query

- **Caching.**
- **Automatic Refetching** **Complex state.**
- **Simplified** **State.**
- **Background** **Refetching**



How **React Query** Replaces **useEffect**



Instead of this ↓

```
useEffect(() => {  
  fetchData();  
}, []);
```

Use this ↓

```
const { data, error, isLoading } =  
  useQuery('key', fetchData);
```



React Query is perfect for:

- **Pagination.**
- **Infinite scrolling**
- **Data synchronization across components**



Conclusion

React Query simplifies data fetching, reduces boilerplate code, and boosts performance.

Why stick with useEffect when React Query does it all?





Faizan Rauf
Website Developer

Need a Website for
Your Business?

Let's Discuss in DM

