



LACSPACE

Developer Documentation

React Kit — hooks, state, data & UX

@lacspace/query

Data fetching & cache

v1.0.0 · zero deps · react 18+ · generated 2026-09-01

Overview

useQuery + useMutation with a shared cache, request de-duplication, stale-while-revalidate and focus/reconnect revalidation — an SWR-lite in ~2 KB.

NOTE

When to use — Fetching, caching and revalidating server data in React without hand-rolling loading/error state — a lighter alternative to React Query or SWR when you want zero dependencies.

Package: @lacspace/query · v1.0.0

Kit: React Kit — hooks, state, data & UX

Runtime: zero deps · react 18+

Install

```
npm i @lacspace/query
```

Quick example

```
const { data } = useQuery("/me", fetchMe)
```

Key API

useQuery(key, fetcher, options?) -> { data, error, isLoading, isFetching, refetch }.

useMutation(fn, options?) -> { mutate, mutateAsync, isPending, data, error, reset }.

mutate(key, data?, opts?) Revalidate or optimistically update a cache entry.

prefetchQuery(key, fetcher) Warm the cache before render.

getQueryData / setQueryData / clearQueryCache Read, write and clear the shared cache.

Tips & warnings

TIP

The cache is keyed and shared: two components calling `useQuery` with the same key share one request and re-render together. Use array keys for params: `useQuery(["user", id], ...)`.

TIP

Stale-while-revalidate is the default — cached data shows instantly while a fresh copy loads in the background. Tune with `staleTime`.

TIP

Disable a query until a dependency is ready by passing a ``null`/`false`` key: ``useQuery(id ? ["user", id] : null, ...)``.

TIP

After a mutation, call ``mutate(key)`` to revalidate, or ``setQueryData(key, data)`` for an optimistic update.

WARNING

The cache lives in module memory — it resets on full page reload and is per-tab. It's a client cache, not persistence.

WARNING

Fetchers must throw on failure (e.g. non-2xx) for ``error`/`isError`` to populate — a resolved promise is treated as success.

Links & full reference

- npm: <https://www.npmjs.com/package/@lacspace/query>
- Source & full README: <https://github.com/lacspace/npm-packages/tree/main/query>
- Docs: lacspace.com/docs/query

The npm page and GitHub README carry the complete API reference and more examples.