LACSPACE

Developer Documentation

Utils Kit — the missing standard library

@lacspace/cache

LRU + TTL + stale-while-revalidate

v1.0.0 · zero dependencies · isomorphic · generated 2026-09-01

Overview

An in-memory cache that does the three things you actually want — bound the size (LRU), expire entries (TTL) and hide latency (stale-while-revalidate) — plus wrap()/memoize() that cache any async function and de-duplicate concurrent in-flight calls. 100 callers, one fetch.

> **NOTE**
> When to use — Caching expensive async work (DB/API calls) in-process, with size bounds and freshness control — and de-duplicating stampedes.

Package: @lacspace/cache · v1.0.0

Kit: Utils Kit — the missing standard library

Runtime: zero dependencies · isomorphic

Install

```
npm i @lacspace/cache
```

Quick example

```
cache.wrap(key, fetchUser, { ttl })
```

Returns: **de-duped + cached**

Key API

createCache({ max, ttl }) -> get/set/has/delete/clear/keys/size/wrap.

cache.wrap(key, fn, { ttl, staleWhileRevalidate }) Cached async with de-dup + SWR.

memoize(fn, opts) Memoized function keyed by arguments, with `.cache`.

Tips & warnings

> **TIP**
> `cache.wrap(key, fn)` de-duplicates concurrent callers — 100 simultaneous misses trigger exactly one `fn()`.

> **TIP**
> Add `staleWhileRevalidate` to serve the cached value instantly after expiry while refreshing in the background.

> **TIP**
> `memoize(fn, { ttl })` wraps a function and keys by its arguments; use `.cache` for manual control.

TIP
Zero-dependency and isomorphic — safe to import on the server, in the browser, on the edge and in React Native. It tree-shakes, so you only ship what you import.

WARNING
This is a per-process, in-memory cache — it doesn't share across serverless instances or survive restarts. For shared state, back it with Redis via your own store.

WARNING
Cache keys are strings; for object args, `memoize` uses `JSON.stringify` by default — pass a custom `key` for stable hashing.

Links & full reference

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

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