



LACSPACE

Developer Documentation

React Kit — hooks, state, data & UX

@lacspace/store

Global state in ~1 KB

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

Overview

Create a store, use selectors, no provider — built on `useSyncExternalStore` with a `persist` middleware and shallow equality. A Zustand-lite that also works outside React.

NOTE

When to use — Sharing state across components without prop-drilling or a Context provider — and when Redux/MobX feel like too much. Also great for state you need to read or write from outside React.

Package: @lacspace/store · v1.0.0

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

Runtime: zero deps · react 18+

Install

```
npm i @lacspace/store
```

Quick example

```
const useStore = create((set) => ({ n: 0 })))
```

Key API

create(initializer) Returns a `useStore` hook + attached `getState/setState/subscribe`.

createStore(initializer) The vanilla, framework-agnostic store (no React).

useStore(selector?, equalityFn?) Subscribe to the whole state or a selected slice.

persist(initializer, options) `localStorage/sessionStorage` middleware with `partialize` + `version`.

shallow(a, b) One-level equality for multi-value selectors.

Tips & warnings

TIP

`create((set, get) => ({ ... })))` returns a hook that is ALSO the store API — call useStore.getState() / useStore.setState() / useStore.subscribe() anywhere, even outside components.`

TIP

Pass a selector to subscribe to just one slice: `useStore((s) => s.count)` — the component re-renders only when that slice changes.

TIP

Selecting multiple values? Return an object and pass ``shallow`` as the equality fn to avoid needless re-renders: ``useStore((s) => ({ a: s.a, b: s.b })), shallow)``.

TIP

Wrap the initializer in ``persist(fn, { name })`` to save to `localStorage`; use ``partialize`` to persist only part of the state.

WARNING

``setState`` shallow-merges by default — pass ``replace: true`` to swap the whole state object.

WARNING

``persist`` reads storage on creation (SSR-safe); the first server render uses initial state, so guard against layout shift if persisted values change what renders.

Links & full reference

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

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