



LACSPACE

Developer Documentation

React Kit — hooks, state, data & UX

@lacspace/virtual

List virtualization

v1.0.0 · zero deps · react 18+ · generated 2026-08-25

Overview

Render 100k rows smoothly — headless windowing with fixed or dynamically-measured sizes, overscan and scroll-to-index. `useVirtualizer` in ~2 KB.

NOTE

When to use — Rendering long lists or grids (hundreds to hundreds of thousands of rows) without janking the browser — only the visible rows are mounted. A zero-dependency TanStack-Virtual alternative.

Package: @lacspace/virtual · v1.0.0

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

Runtime: zero deps · react 18+

Install

```
npm i @lacspace/virtual
```

Quick example

```
const v = useVirtualizer({ count, ... })
```

Key API

`useVirtualizer(options)` `count`, `getScrollElement`, `estimateSize`, `overscan`, `horizontal`, `gap`.

`getVirtualItems()` The rows to render -> { `index`, `start`, `size`, `end`, `key` }.

`getTotalSize()` Total scrollable size for the spacer element.

`measureElement(el)` / `scrollToIndex(i)` Dynamic measurement and programmatic scrolling.

Tips & warnings

TIP

It's headless: you own the markup. Give the scroll container a fixed height + `overflow:auto`, render a spacer of `getTotalSize()`px, and absolutely-position each item from `getVirtualItems()` at `transform: translateY(item.start)`.

TIP

For variable row heights, set `data-index={item.index}` and `ref={v.measureElement}` on each row — it measures real heights and corrects the layout.

TIP

Tune `overscan` (default 5) to render a few extra rows above/below the viewport for smoother fast-scrolling.

TIP

``scrollToIndex(i, { align })`` jumps to any row, even one far outside the current window.

WARNING

The scroll element must actually scroll (a bounded height with ``overflow:auto``); if the page itself scrolls instead, offsets won't track.

WARNING

``estimateSize`` should be close to the real size — a wildly wrong estimate makes the scrollbar jump as rows are measured.

Links & full reference

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

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