



LACSPACE

Developer Documentation

WebKit — for web apps & backends

@lacspace/rate-limit

Rate limiting

v1.1.1 · zero dependencies · generated 2026-09-01

Overview

Fixed-window, sliding-window and token-bucket algorithms over a pluggable store (in-memory built in; bring your own Redis). Standard RateLimit-* headers. Drop into any API route, middleware or edge function.

NOTE

When to use — Protecting endpoints (login, contact, APIs) from abuse with token-bucket, fixed- or sliding-window limits.

Package: @lacspace/rate-limit · v1.1.1

Kit: WebKit — for web apps & backends

Runtime: zero dependencies

Install

```
npm i @lacspace/rate-limit
```

Quick example

```
await rateLimit({ limit: 10, windowMs: 60000 }).check(ip)
```

Returns: { **success**, **remaining** }

Tips & warnings

TIP

Return `rateLimitHeaders()` so clients see `RateLimit-\*` and `Retry-After`.

TIP

Key by user id when authenticated, IP otherwise.

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

The default store is in-memory per instance; use a shared store (Redis) for accurate limits across a fleet.

Links & full reference

- npm: <https://www.npmjs.com/package/@lacspace/rate-limit>

- Source & full README: <https://github.com/lacspace/npm-packages/tree/main/rate-limit>
- Docs: lacspace.com/docs/rate-limit

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

LACSPACE
DEVELOPER DOCS