



LACSPACE

Developer Documentation

Backend Kit — server-side essentials

@lacspace/signed-url

Signed & expiring URLs / tokens

v1.0.0 · built on @lacspace/crypto · generated 2026-09-01

Overview

HMAC-signed, expiring URLs and tokens over Web Crypto — magic-login links, secure download links, unsubscribe links and one-time actions. Tamper-proof, timing-safe, with `magicLink()` / `signUrl()` helpers. Fills a real gap: no self-hosted, zero-dep option existed.

NOTE

When to use — Creating expiring, tamper-proof links — magic login, password reset, time-limited downloads.

Package: `@lacspace/signed-url` · v1.0.0

Kit: Backend Kit — server-side essentials

Runtime: built on `@lacspace/crypto`

Install

```
npm i @lacspace/signed-url
```

Quick example

```
await verifyUrl(request.url, { secret })
```

Returns: `{ valid: true }`

Tips & warnings

TIP

Keep expiries short and include a purpose/scope in the payload so a link can't be reused elsewhere.

WARNING

The signing secret must stay server-side — never ship it to the client.

Links & full reference

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

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