



LACSPACE

Developer Documentation

[App Kit](#) — [ship a polished app](#)

@lacspace/validate

Zod's ergonomics, zero deps

[v1.0.0](#) · [zero dependencies](#) · [isomorphic](#) · [generated 2026-08-25](#)

Overview

parse / safeParse, objects, arrays, enums, unions and full type inference — plus coercion ("42" -> 42, "true" -> true) so one schema validates a JSON body and an HTML form. Field errors flatten() straight into a form. The validation you reach for, small enough to drop into any edge function.

NOTE

When to use — Validating anything untrusted — form bodies, API payloads, env vars, query strings — with type inference, in a package small enough for edge functions.

Package: @lacspace/validate · v1.0.0

Kit: App Kit — ship a polished app

Runtime: zero dependencies · isomorphic

Install

```
npm i @lacspace/validate
```

Quick example

```
User.safeParse(input)
```

Returns: { **success**, **data** | **error** }

Key API

v.string()/number()/boolean()/object()/array()/enum()/union() Schema builders.

.optional()/default()/refine()/transform() Modifiers.

parse() / safeParse() / Infer<> Validate and infer types.

Tips & warnings

TIP

Use ``v.coerce.number()`` / ``v.coerce.boolean()`` for FormData and query strings, which arrive as strings.

TIP

``schema.safeParse(x)`` returns ``{ success, data | error }``; call ``error.flatten()`` to get ``{ field: message }`` for forms.

TIP

``Infer<typeof schema>`` gives you the exact TypeScript type — one source of truth for runtime and compile time.

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

`.strict()` rejects unknown keys — pair it with `@lacspace/form`, which strips internal fields (honeypot, timestamp) before validation.

Links & full reference

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

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