



LACSPACE

Developer Documentation

App Kit — ship a polished app

@lacspace/form

Typed forms + spam guard

v1.0.0 · zero dependencies · isomorphic · generated 2026-09-01

Overview

Turn a FormData into typed, validated data with a honeypot + timing spam guard, and get back either your data or per-field errors ready to re-render. `createForm().action` matches Next.js `useActionState`, so a full contact form is a few lines. Framework-agnostic, zero-dep.

NOTE

When to use — Handling form submissions on the server (Next.js Server Actions) with validation and spam protection, without a form library.

Package: @lacspace/form · v1.0.0

Kit: App Kit — ship a polished app

Runtime: zero dependencies · isomorphic

Install

```
npm i @lacspace/form
```

Quick example

```
contact.action(prev, formData)
```

Returns: { **ok**, **data** | **errors** }

Key API

createForm(opts) -> { handle, action } bound to a schema + spam options.

formDataToObject(fd) FormData -> plain object (repeated keys -> arrays).

honeypotProps(name) / timestampValue() Client helpers for spam protection.

Tips & warnings

TIP

`createForm({ schema }).action` matches useActionState`, so a full contact form is a few lines.`

TIP

Add a `honeypot`` field and `minSubmitMs`` to block bots; internal fields are stripped before validation.

TIP

The result echoes `values``, so you can re-render the form with what the user typed on error.

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 honeypot input must be visually hidden but present in the DOM — use the provided `honeypotProps()` helper.

Links & full reference

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

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