



LACSPACE

Developer Documentation

DX Kit — everyday utilities

@lacspace/money

Money without float bugs

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

Overview

$0.1 + 0.2 \neq 0.3$ — so never store money as a float. Keeps amounts as integer minor units, refuses to add different currencies, splits a bill without losing a cent (`allocate`), handles zero- and three-decimal currencies, and formats with Intl. Immutable, typed.

NOTE

When to use — Any prices, totals, or financial math — where floating-point cents bugs and currency mix-ups are unacceptable.

Package: `@lacspace/money` · v1.0.0

Kit: DX Kit — everyday utilities

Runtime: zero dependencies · isomorphic

Install

```
npm i @lacspace/money
```

Quick example

```
money(10, "USD").allocate([1, 1, 1])
```

Returns: `[$3.34,$3.33,$3.33]`

Key API

`money(major, ccy)` / `Money.fromMinor(minor, ccy)` Construct.

`.add/``.subtract/``.multiply/``.allocate/``.split` Currency-safe operations.

`.format(locale?)` / `.toMinor()` / `.toJSON()` Output.

Tips & warnings

TIP

Store and compute in integer minor units; only format for display with `.format(locale)`.

TIP

Split bills with `.allocate([ratios])` / `.split(n)` — the remainder is distributed so the total is always exact.

TIP

Zero-decimal (JPY) and three-decimal (BHD/KWD) currencies are handled automatically.

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

Arithmetic across different currencies throws — convert first with your own FX rate.

WARNING

`.toMajor()` returns a JS number for convenience; prefer `.format()` for display to avoid float rounding.

Links & full reference

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

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