

zapier.georgelambert.org · Markdown indexes

VERAE TIME × ZAPIER

## server

site/docs/modules/zapier-accounting-export/server.md

## zapier-accounting-export/server

**Package:** zapier-accounting-export**Source:** packages/zapier-accounting-export/src/server.js**Lines:** 62

## What this module is

---

Implementation module in `zapier-accounting-export`. The tables below are extracted from the source (signatures + JSDoc).

## Exports

---

No named `export` / `module.exports` keys detected.

## Types / interfaces / classes

---

None extracted.

## Functions

---

| Name              | Parameters                                                   | Param types (JSDoc) | Returns              | Calls (same file / helpers) |
|-------------------|--------------------------------------------------------------|---------------------|----------------------|-----------------------------|
| <code>edge</code> | <code>pathname</code>                                        | —                   | <code>unknown</code> | see Call graph              |
| <code>send</code> | <code>code</code> , <code>type</code> ,<br><code>body</code> | —                   | <code>unknown</code> | see Call graph              |
| <code>json</code> | <code>code</code> , <code>obj</code>                         | —                   | <code>unknown</code> | see Call graph              |

## What it imports / requires

---

- `node:fs`
- `node:http`
- `node:path`
- `node:url`
- `./nats-billing.js`

## Call graph (identifiers invoked)

---

`replace`, `join`, `dirname`, `fileURLToPath`, `edge`, `fetch`, `createServer`, `async`, `writeHead`, `end`,  
`send`, `stringify`, `readFileSync`, `json`, `match`, `billingRequest`, `text`, `get`,  
`encodeURIComponent`, `listen`, `write`

Each identifier is a call site in this file. Follow the import list to see the defining module; open that module's MD for parameter and return types.

## Return values (how to read this)

---

- HTTP route handlers return Express `res.json(...)` bodies (see route docs).
- Zapier `perform` functions return a **single object** (creates) or an **array** (triggers/searches).
- Pricing functions return integer **cents** on `Quote.totalCents`.