zapier.georgelambert.org · Markdown indexes
VERAE TIME × ZAPIER

Invoices, PO billing, reports

site/packages/zappier/docs/ACCOUNTING.md

Company Accounting Walkthrough

This guide covers the company-side accounting surface: invoicing, purchase-order billing, reports, trends, and CSV export. Everything here runs in the **admin console** at `/admin`.

> Audience: operations and accounting staff. For pricing and customer setup see `USER-MANAGEMENT.md`; for the end-user view see `CUSTOMER-PORTAL.md`.

---

1. Sign in

Open `http://<host>:<port>/admin` and sign in. The development credentials are `demo` / `$$$Adm1n###` (override with `DEMO_ADMIN_USER` / `DEMO_ADMIN_PASSWORD`; the primary account is `ADMIN_USER` / `ADMIN_KEY`).

Admin login
Admin login

---

2. Billing identity per customer

Before invoicing, each customer needs a **billing type** and an **email** — set on the **Customers** tab:

- **Stripe** — metered usage is reported to Stripe by the daily billing job.
- **Purchase order** — invoiced manually with a PO number; issued PO invoices get a 30-day due date automatically.

Customers with billing types
Customers with billing types

Customers can also carry a **prepaid balance** (funded from the customer portal). When an invoice is issued and the balance fully covers the billable amount, the balance is drawn down and the invoice goes straight to **paid**. Partial coverage is left untouched — there are no partial payments.

---

3. Generate invoices

On the **Invoices** tab, pick a **period** (month), optionally narrow to one customer, optionally set a **PO number**, and click **Generate**.

Invoices tab
Invoices tab

Generation rules:

- One invoice per customer with usage in the period, grouped by endpoint.
- The tier **monthly credit** is applied; only the remainder is billable.
- Regenerating a period **replaces drafts** (e.g. after late-arriving usage) and **skips issued/paid invoices** — the result panel lists who was skipped and why.
- Invoice ids are `INV-<period>-<sequence>`, e.g. `INV-2026-07-0002`.

4. Issue, view, collect

Each row shows status, billing type, totals, and due amount:

- **Issue** (draft → issued) — finalizes the invoice; PO invoices get a due date 30 days out. Balance-covered invoices skip straight to **paid**.
- **View** — opens the print-ready invoice in a new tab. Use the browser's **Print** → **Save as PDF** to produce a PDF for the customer.
- **Mark paid** (issued → paid) — record collection for PO invoices.

Print-ready invoice
Print-ready invoice

Filter the table by customer, period, or status with the filter bar.

---

5. Reports and trends

The **Reports** tab answers "who owes what, and how is usage trending?"

Reports tab
Reports tab

- **Billing report** — per-customer calls, gross usage, credits applied, and billable amount for a date range. Filter by customer or billing type (all Stripe customers, all PO customers, or one customer). Summary cards total the selection.
- **Download CSV** — the same rows as `billing-report.csv` with the active filters, ready for the accounting package.
- **Usage trend** — daily or weekly buckets as a bar chart; hover a bar for exact calls and amount.

The same data is available as JSON/CSV from the API: `GET /admin/api/reports/billing?from=…&to=…&customerId=…&billingType=…&format=csv` and `GET /admin/api/reports/usage-trend?bucket=day|week`.

---

6. System snapshot

The **System** tab shows integration health (Zapier app directory, version, triggers, creates) and the current period at a glance: calls, billable amount, open invoice count, and open amount.

System tab
System tab

---

7. Automated Stripe reporting

A daily job (`src/jobs/report-usage.ts`, scheduled separately) reports the billable delta of every Stripe-billed customer to Stripe Billing meter events. It is idempotent: a ledger records the cumulative reported cents per customer per period, and only the delta since the last successful run is sent. PO customers are excluded by having no `stripeCustomerId`.

Environment (`.env` at the project root):

```
STRIPE_SECRET_KEY=sk_live_or_test_...
ZAPPIER_DB=/absolute/path/to/zappier.db   # optional
```

Data notes

- Money is integer **cents** everywhere internally; the UI formats dollars.

- All accounting data lives in the SQLite database (`zappier.db` by default): `invoices`, `invoice_lines`, `billing_ledger`, `customers`.
- The Stripe billing job and the admin console can run from any working directory — all paths resolve from the installation root.