



Platform User Manual

For the two vendor seats — Platform Administrator and Platform Business Officer — who run the service every tenant business runs on, and cross tenants to do it.

COVERS

platform_admin, business_officer

VERSION

2026-09-23

UPDATED WITH

docs/REPORTING_PLAN.md – Tokens generated, and their catalog value (both platform reports)

Who this is for

Two seats, deliberately split so no single account both runs the service and decides what a customer pays:

SEAT	WHAT IT IS FOR	ISSUES TOKENS?
Platform Administrator	Runs the service — onboards businesses, keeps the platform healthy, resets a locked-out tenant administrator	Never
Platform Business Officer	Commercial terms across every tenant — plans, pricing tier changes, closing an account, recording what a tenant paid the vendor	Never

Neither seat may issue a token to any device, on any business's kit — that stays with the tenant's own staff. Neither may do the other's job: an administrator cannot change a plan, an officer cannot reset a tenant administrator's password.

Any list on any screen — tenants, requests, a report's own tables — shows 5 rows at a time by default. The box beside it lets you choose 10, 25, 50, 100, 200, 500 or All, filter by typing, and sort by clicking a column heading; **Download CSV** saves every row currently matching your filter, not only the page on screen.

In this manual

1. Businesses

2. Add a business

3. Requests

4. Connections

Businesses

[Both vendor seats read](#)[Business Officer changes the plan](#)

Every tenant on the platform, and the seats at each.

- **Tenants** — every business, its plan, and its billing state at a glance.
- **People at a business** — who holds a seat there, and as what. Identity only — no password, no allow-list, no limits — sorted most senior first.
- **Change a plan** (Business Officer) — move a tenant onto a different plan directly. This is the vendor's own act, recorded separately from a tenant asking and being approved (see Requests). The same screen's **Paid connections** section switches a sold-separately integration (Prospect today) on or off for this one tenant, and sets whether it is billed as a flat fee per connection or priced per device sent — it also shows how many times the tenant has used it this period. No invoice is generated here yet; this only turns the connection on and keeps the count.
- **Edit a business** — its own contact and billing details.

Add a business

[Platform Administrator](#)

Onboard a new tenant — its name and contact details, its starting plan, and its first administrator seat, created in the same step so the new business can sign in immediately.

Requests

Both vendor seats read

Business Officer decides

Every plan-change or closure request a tenant has asked for, across every business, pending ones first.

1. Open a request to see what the tenant asked for and why.
2. **Plan requests** — a Business Officer approves or declines. An approval applies the new plan immediately; nothing changes on a decline beyond the record of the decision.
3. **Closure requests** — approved either *immediately* (issuing stops now) or *at the end of the paid term* (runs to the date already paid for, then stops). Closing a tenant never deletes anything — every device, token, payment and customer record stays readable, which is what regulators and the business itself expect.

Connections

Platform Administrator

The outside services PayGO talks to — the **apigw** payment gateway, **SMS gateways** and the **mail (SMTP)** server — added and changed here, with no developer and no redeploy.

- **+ Add connection** — pick the provider, then fill in only what it needs: endpoints, app name or username, IDs, keys or passwords, sender ID. Keys and passwords are stored encrypted and are **never shown again**; when editing, leave a secret blank to keep the stored one.
- **Your authenticator code** — adding, editing, rotating or re-enabling a connection asks for a fresh 6-digit code. Disabling does not: stopping a connection is never harder than starting one.
- **apigw** — PayGO generates the signing key and shows it **once**, with the address apigw must send payments to. Set the same key in apigw (PAYGO_APIGW_KEY), then press **Test**: "Accepted by apigw" means both sides hold the same key. **Rotate key** issues a new one while the old one keeps working for the grace period you choose.
- **SMS gateways** — Africa's Talking, Beem, NextSMS, or any other gateway with an HTTP JSON API. Tick **Platform default** for every business without its own gateway, or list the businesses that use it. **Test** sends one real SMS to the number you give.
- **Email (SMTP)** — host, port (465 or 587), username, password, from address. **Test** sends one email to you.
- **Allowed domains** — PayGO only ever calls these hosts (https, public addresses). The provider's own domains are added for you.
- **Allowed IPs** (payment gateways only) — optional; payments must come from these addresses *as well as* carry a valid signature. The line "PayGO sees your address as ..." shows how PayGO reads an address — check it before relying on an IP list.
- Every change is recorded on the **Security** screen with who made it.

Security

Platform Administrator

Is somebody trying to get in — across every business, last 24 hours. See `docs/SECURITY_AUDIT.md` for the full "what to do about it" guidance; this is the summary.

TILE	NORMAL	ACT WHEN
Failed logins	a handful	more than 20 in a day, or many against one account
Sign-ins	roughly your staff count	sign-ins at hours nobody works
Blocked attempts	0	any — a lockout fired
Sessions revoked	0 unless an account changed	unexplained
Locked now	0	any — named, with the unlock time

Most attempted names the accounts being guessed at. What to do, in order: confirm the account holder was not simply locked out by their own typing; if not, reset that password; check sessions revoked for evidence the old token was still used; then check the device ledger for anything issued in that window.

Reports

Both vendor seats — each sees only their own type

A platform-wide snapshot, never one tenant's own report re-sent.

1. **Report** — pick Daily, Weekly or Monthly and press View. **Platform Administrator** sees *Operational health*: tenants near a plan ceiling, billing holds, security and penetration attempts (platform-wide and by tenant — failed logins, lockouts, forged/invalid tokens and API keys), 3rd-party reachability (SMS and payment-callback delivery, whether email is configured), and storage document counts. **Business Officer** sees *Commercial summary*: revenue by currency and pending plan requests.
2. **Tenant** — a dropdown beside Period, "All tenants" by default. Choosing one business there narrows every section below it to that tenant alone — the ceiling/billing figures, the security totals, tokens, devices, revenue and pending requests all become that one business's own numbers, not the platform's. A hint above the report says so while a tenant is chosen. The one exception is **Storage**, which answers a platform-growth question ("is this collection growing the way it should"), not a per-tenant one, so it always stays platform-wide. Clear the dropdown back to "All tenants" to return to the full snapshot.
3. **Tokens generated** — on both reports, the same section: tokens issued in the period, a breakdown by type (Add time, Set time, Lock device, Unlock permanently, Resynchronise counter) with days granted, how many distinct devices were given a token, and a **catalog value** figure priced off each tenant's own Prices screen — every token counts its device's product price, added up, so a device topped up several times in the period counts its price that many times. This is *not* the same figure as **Revenue this period** above it on the Commercial report, which is actual money the vendor has received from tenants — the two are shown side by side, distinctly worded, on purpose. A token whose device's model has no price set is still counted as a token, but not toward the value figure, and shown separately as "no price on file." A **By tenant** table breaks the same tokens, days and catalog value down per business, named rather than shown by internal id — type a business's name in that table's own filter box to narrow the table to just that tenant without changing the "Tenant" dropdown above (the dropdown narrows the whole report; this filter box narrows only this one table, the same convention every list on this platform uses).
4. **Devices** — also on both reports: how many devices are loaded on the platform (or, with the Tenant filter set, that one business's own devices), and their state right now — *locked*, *in use*, *permanently unlocked*, and *not yet used* (the device's own counter has never moved off 1: no token has ever been issued to it). This is a snapshot of the devices themselves, not of what happened in the report's own period — a kit tokenised months ago and untouched since still has a state today.
5. **Reports you receive** — turn on Daily/Weekly/Monthly to have your own report emailed automatically, and an SMS alert for same-day issues (a real lockout, for the Administrator; a pending plan request, for the Officer).

Storage is a document count, not usage against a live quota — a growth figure to watch, not yet a real comparison against an allocation. Email reachability reports only whether SMTP is configured right now, not a historical failure rate.

What either seat may do to a tenant

Three deliberately bounded powers — see CLAUDE.md's own "What the vendor may do to a customer's business" for the full reasoning.

POWER	WHO	BOUND
Read who holds a seat at a business	Both	Identity only — no password, no limits
Reset a locked-out password	Administrator only	Only a Tenant Administrator seat — the goal is to unlock the business, not take over any account. Forces a change at next sign-in and drops any session already open.
Close (or reopen) a business's account	Business Officer only	Sets a closed date; issuing stops, nothing is deleted, every record stays readable. Reversible.

GreenLeaf Technology Solutions Co. Ltd — PayGO Token Service. This manual is a living document — see CLAUDE.md's own instruction that it is updated in the same change as any screen it describes.