



INDITION COMMERCE

Point of Sale

Setup Guide — software & hardware installation, from zero to first sale:
the module, Stripe, the card reader, the cash drawer, and going live.

Point of Sale Module · Version 1.3 · Updated 2026-07-19



© 2026 Indition Corp. All rights reserved. Indition™ and the Indition logo are trademarks of Indition Corp. All other product names, logos, and brands are the property of their respective owners.

Who this is for

This guide takes a new store from nothing to a working register. It covers the **software** (installing and enabling the POS module, configuring settings, granting staff access) and the **hardware** (the Stripe card reader, the cash drawer, the receipt printer, and the computer that runs the register).

Do the work in the order below. Always set up and test in **Stripe test mode** first, then switch to live keys at the very end. A first-time, single-store install takes roughly 60–90 minutes.

Contents

- | | |
|---|---------------------------------------|
| 1. How the system fits together | 9. Card readers (Stripe Terminal) |
| 2. What you need before you start | 10. Multi-store & Stripe Connect |
| 3. Software setup (module & settings) | 11. Cash drawer & receipt printer |
| 4. Store & staff configuration | 12. Touch POS & the register computer |
| 5. Inventory: stock control & locations | 13. Test sale & go-live |
| 6. Payment methods & manual card entry | 14. Troubleshooting |
| 7. Receipts: print, email & text | 15. Appendix: settings reference |
| 8. Stripe account & keys | |

1. How the system fits together

The register is a web page served from your Indition site at `/pos/`. It talks to three things:

- **Your Indition site** — the catalogue, customers, inventory, sales, and reports all live here. The register is just a front end onto the same data as your webshop.
- **Stripe** — for in-person card payments. The card reader talks to Stripe directly over Wi-Fi; your site tells the reader what to charge and gets the result back through a Stripe *webhook*.
- **Local peripherals** — the cash drawer and receipt printer are driven by a tiny helper program ("drawer agent") running on the register computer, because browsers can't open USB devices on their own.

Modules involved: `POS` (the register and admin), `POSPFront` (the public `/pos/` entry point), `StripeTerminal` (card-reader plumbing and the webhook), and the existing `Shop`, `InStorePickup` (stores/locations) and `Inventory` modules. Stripe Connect is only needed for multi-tenant / per-store payout setups.

2. What you need before you start

Item	Detail
Indition admin login	With administrator rights (to install modules and edit settings).
A Stripe account	You'll use test mode keys first, then live keys at go-live.
Card reader	A Stripe Reader S700 (recommended) or S710 — countertop, Wi-Fi. For a handheld option, a Stripe M2 (Bluetooth, paired through the companion app). See section 9 for which setup each one uses.
Store Wi-Fi	Network name and password. The reader needs reliable Wi-Fi with outbound internet access.
Register computer	Any modern PC, Mac, or tablet with an up-to-date browser. A USB barcode scanner is strongly recommended.
Cash drawer + receipt printer	Optional but typical: an ESC/POS receipt printer with an RJ-12 ("DK-Port") cash drawer attached.

3. Software setup (module & settings)

3.1 Install the modules

Through the Indition module manager (or the module installer), install:

- **POS** — the register and its back-office screens, roles, and settings.
- **StripeTerminal** — the card-reader connection and the secure link that returns card results to your site.
- **InStorePickup** — adds the "Is POS Location" and Stripe Terminal Location settings to your store records. Make sure it's up to date.

Indition installs and verifies these modules for you during provisioning. Once they're in place you'll see the **POS** menu in the back office and the **StripeTerminal** entry in the module manager. There's nothing to run yourself — if the POS menu is missing, contact Indition support.

3.2 Enable POS and pick stores

- 1 Go to **POS → Settings** (`/admin/POS/admin/settings/index`).
- 2 Turn on **POS Module Enabled** (the master switch — off by default).
- 3 Under **Stores**, select the store(s) that will use the register. At least one is required.
- 4 Save.

3.3 Configure the basics

The defaults are sensible — most installs only need Stripe keys plus store selection. Review these now; the full list is in the [appendix](#):

- **Tax** — leave *Override Real-Time Tax Quoting* off to quote live tax automatically, or turn it on and set a flat *Default Tax Rate*.
- **Payment methods** — Cash and Card are on by default; enable Check, Split, and Manual Card Entry if you need them ([section 6](#)).
- **Card processing fee** — off by default. If you pass card costs to customers, enable it and set the percent (default 2.9) and fixed amount (default 0.30). Cash/check are never charged.
- **Inventory** — if the Inventory module is enabled, review block-oversell, override, and per-store locations ([section 5](#)).
- **Receipts** — set the header/footer and, if wanted, email and text receipts ([section 7](#)).
- **Quick keys, tipping, fractions, touch UI** — optional polish.

4. Store & staff configuration

4.1 Flag the store as a POS location

- 1 Open the store record at **InStorePickup** → **Stores** (`/admin/InStorePickup/admin/dealer/admin`).
- 2 Set **Is POS Location** = ON.
- 3 Leave **Stripe Terminal Location ID** blank for now — you'll paste the `tml_...` ID in section 8.
- 4 Save.

Important: A reader will refuse to take a payment if its store isn't flagged as a POS Location (you'll see a "Store is not flagged as a POS Location" error). Don't skip this.

4.2 Grant staff access

Under **POS** → **Staff & Store Access**, give each employee a role and (in multi-store setups) their stores and home store:

Role	Grants
POS Cashier	Ring up sales and take payment.
POS Manager	Cashier rights + void sales, approve oversells, view reports and analytics.
POS Admin	Manager rights + manage card readers, settings, and staff access.

- With a **single** POS store, anyone you give a role is auto-assigned that store — no picker appears.
- With **multiple** stores, tick the stores each person can work at and choose one **home store** for sign-in.
- Role changes take effect at the user's next register sign-in.

Note: Register sign-in uses the same accounts as the back office (the global admin users). If you turn on *Require cashiers to be assigned to a store*, unassigned staff can't bind a register until you assign them here.

4.3 Screen lock & PIN (optional)

For shared or unattended terminals, turn on the **screen lock** so the register locks itself and resumes with a short PIN instead of a full sign-in. A cashier signs in to start their shift and out for a break; in between, the

screen locks on a timeout and a PIN taps it back open. It's off by default and applies only to the POS roles you choose — it never affects any other admin login. Configure it under **POS → Settings → Screen Lock & PIN**.

Setting	Default	What it does
Enable screen lock & PIN unlock (<code>POS_PIN_UNLOCK_ENABLED</code>)	Off	Master switch. When off, the terminal never locks and no PIN is required — behaviour exactly as before.
Idle timeout (seconds) (<code>POS_LOCK_IDLE_TIMEOUT_SECS</code>)	300	Lock after this many seconds of inactivity. <code>0</code> disables the idle lock (e.g. <code>300</code> = 5 minutes).
Re-lock interval (seconds) (<code>POS_LOCK_INTERVAL_SECS</code>)	0	Force a re-lock this many seconds after the last unlock, regardless of activity — useful to require a PIN between customers. <code>0</code> disables. Deferred while a sale is in progress.
Minimum PIN length (<code>POS_PIN_MIN_LENGTH</code>)	4	Fewest digits a PIN may have (4+).
PIN rotation (days) (<code>POS_PIN_ROTATION_DAYS</code>)	90	Require a new PIN this many days after it was last set. <code>0</code> = never expire.
Max unlock attempts (<code>POS_PIN_MAX_ATTEMPTS</code>)	5	After this many wrong PINs in a row, the terminal signs the cashier out and requires a full username/password login.
Roles the lock applies to (<code>POS_PIN_LOCK_ROLES</code>)	Cashier, Manager, Admin	Turn the lock on per POS role. A person is locked only when their POS role is enabled here.

First unlock: the first time the lock applies to a cashier, the register prompts them to choose their own PIN (respecting the minimum length). There's no shared PIN to distribute. A forgotten PIN, or too many wrong tries, is reset simply by signing in again with the normal email and password.

Enforced server-side: a locked terminal genuinely cannot transact — the lock is checked on the server, not just hidden in the browser — and a sale in progress is never interrupted mid-flight.

5. Inventory: stock control & locations

When the **Inventory** module is enabled, the POS deducts stock from the correct location on every sale and can guard against overselling. These controls live under **POS → Settings → Inventory** and (for locations) under **POS → Settings → Stores**. If the Inventory module is not enabled, these options have no effect and untracked products always sell.

5.1 Block oversell (with manager override)

Block Oversell (`POS_BLOCK_OVERSELL` , default **ON**) stops a cashier *before payment* if a tracked line exceeds the on-hand quantity at the register's store. A **POS Manager** can authorise the sale anyway. Serialized items are always checked; untracked items are never blocked.

5.2 Inventory override on out-of-stock

Allow Inventory Override (`POS_ALLOW_INVENTORY_OVERRIDE` , default **OFF**) lets a cashier sell a tracked item that shows out of stock — for the common case where the item is physically in hand but the count is wrong. When the sale completes, the shortfall is *received* into inventory and then sold, so on-hand reconciles with an audit trail instead of going negative.

The receive is booked at a configurable **cost basis** (`POS_INVENTORY_OVERRIDE_COST_BASIS`):

Cost basis	Receives the shortfall at...
Last purchase cost (<code>last</code>)	The most recent purchase cost recorded for the item.
Average purchase cost (<code>average</code>)	The item's average purchase cost.
\$0 (<code>zero</code>)	Zero cost (treat as no cost basis).

Use with care: Override is a self-healing convenience, not a substitute for accurate counts. Leave it OFF unless your staff are trusted to use it correctly, and reconcile receives during cash-up. Block Oversell and Inventory Override are independent — with both ON, the cashier is warned, a manager can authorise, and the shortfall is then received on completion.

5.3 Per-store warehouse & location selection

For each POS-enabled store you choose the inventory **warehouse** and the **location(s)** the register draws stock from, marking one as **primary**. On-hand shown and sold at the register is summed across the chosen locations; the *primary* location is the one debited on each sale and where an override receive books stock.

- 1 In the Inventory module, make sure the warehouse(s) you want are **linked to the store**. POS only offers warehouses already associated with that store.
- 2 Go to **POS → Settings → Stores**. Beneath each enabled store, pick a **warehouse**.
- 3 Tick the **location(s)** the register sells from, and mark exactly one as **primary**.
- 4 Save.

Fallback: If a store is left unconfigured here, it falls back to its **default inventory location** — the register still sells and deducts stock, just from the store's default location.

6. Payment methods & manual card entry

Payment methods are toggled under **POS → Settings → Payments**. Cash and Card are on by default.

Method	Setting	Notes
Cash	<code>POS_PAYMENT_CASH_ENABLED</code> (on)	Tender, change, and drawer kick.
Credit / debit (reader)	<code>POS_PAYMENT_CARD_ENABLED</code> (on)	In-person card via Stripe Terminal.
Check	<code>POS_PAYMENT_CHECK_ENABLED</code> (off)	Records a check tender.
Split payment	<code>POS_PAYMENT_SPLIT_ENABLED</code> (off)	Multiple tenders on one sale.
Manual card entry	<code>POS_PAYMENT_MANUAL_CARD_ENABLED</code> (off)	Key a card by hand — see below.
Saved cards (card-on-file)	<code>POS_SAVED_CARDS_ENABLED</code> (off)	Charge a customer's stored card.

6.1 Manual (keyed) card entry

Turn on **Manual Card Entry** (`POS_PAYMENT_MANUAL_CARD_ENABLED`) to let a cashier key a card number when a reader isn't available or the card won't read. Manual entry uses the Stripe **publishable key**, so make sure that's set (section 8). Two guardrails are available:

- **Require a customer** (`POS_MANUAL_CARD_REQUIRE_CUSTOMER` , default ON) — the sale must be attached to a customer before a keyed card can be charged.
- **Require ZIP (AVS)** (`POS_MANUAL_CARD_REQUIRE_ZIP`) — prompts for the billing ZIP so Stripe can run an address-verification (AVS) check.

6.2 Tipping & card processing fee

- **Tipping** (`POS_TIPPING_ENABLED`) shows tip presets on card payments; set the presets in `POS_TIP_PRESETS` (default 15, 18, 20).
- **Card processing fee / surcharge** (`POS_CARD_FEE_ENABLED` , off by default): when on, a fee of `POS_CARD_FEE_PCT` (default 2.9%) plus `POS_CARD_FEE_FIXED` (default \$0.30) is added to card sales, recorded in its own column and shown to the cashier. Cash and check are never charged the fee.

Surcharge rules: Card surcharging is regulated and prohibited in some jurisdictions and by some card-network rules. Confirm it's permitted for your business before enabling, and set the percent to your actual cost of acceptance.

7. Receipts: print, email & text

Every completed sale can print, email, and/or text a receipt. Completed receipts carry a **PAID** badge across print, email, and PDF. Receipt settings live under **POS → Settings → Receipts**.

7.1 Header & footer

Set **Receipt Header** (`POS_RECEIPT_HEADER`) and **Receipt Footer** (`POS_RECEIPT_F00TER`), default "Thank you for your business!"). The store name, logo, and address come from the store record.

7.2 Email receipts

Turn on **Email Receipt** (`POS_EMAIL_RECEIPT_ENABLED`) to let cashiers email a receipt with the PDF attached (rendered via mPDF). Configure:

Setting	Purpose
<code>POS_RECEIPT_EMAIL_TEMPLATE</code>	Which email template to use.
<code>POS_RECEIPT_EMAIL_SUBJECT</code>	Subject line of the receipt email.
<code>POS_RECEIPT_EMAIL_FROM_NAME</code>	Sender display name.
<code>POS_RECEIPT_EMAIL_FROM_ADDRESS</code>	Sender email address.
<code>POS_RECEIPT_EMAIL_BCC</code>	Optional BCC address (e.g. a store mailbox).
<code>POS_RECEIPT_EMAIL_BODY</code>	Body text accompanying the receipt.
<code>POS_RECEIPT_EMAIL_ATTACH_PDF</code>	Attach the receipt PDF (on by default).

7.3 Text (SMS) receipts

Turn on **SMS Receipt** (`POS_SMS_RECEIPT_ENABLED`) to let cashiers text a receipt to the customer's mobile number. SMS delivery uses the platform's existing messaging integration; confirm that's configured for your site.

8. Stripe account & keys

Work in **test mode** first (toggle in the top-right of the Stripe Dashboard).

8.1 Create a Terminal Location

- 1 Stripe Dashboard → **Terminal** → **Locations** → **+ New**.
- 2 Give it a name and address. Save.
- 3 Copy the `tml_...` ID and paste it into the store's **Stripe Terminal Location ID** field (section 4.1), then save the store.

8.2 Copy the API keys

- 1 Stripe Dashboard → **Developers** → **API keys**. Copy the **test secret key** (`sk_test_...`) and the matching **publishable key** (`pk_test_...`).
- 2 In **POS** → **Settings**, paste them into **Stripe Secret Key** and **Stripe Publishable Key**. (The publishable key is only needed for the keyed-entry fallback, but it's harmless to set.)

Security: The secret key is stored server-side only and is never sent back to the browser. To change it, type a new value; to remove it, use the field's **Clear** control.

8.3 Set up the webhook

- 1 Stripe Dashboard → **Developers** → **Webhooks** → **+ Add endpoint**.
- 2 URL: your site address followed by `/StripeTerminal/Webhook/receive` (for example `https://www.your-store.com/StripeTerminal/Webhook/receive`). If you're unsure of the exact address, Indition support will confirm it for your site.
- 3 Subscribe to these events:
 - `terminal.reader.action_succeeded`
 - `terminal.reader.action_failed`
 - `terminal.reader.action_updated`
 - `payment_intent.succeeded`
 - `payment_intent.payment_failed`
 - `payment_intent.canceled`
- 4 If you route stores through Stripe Connect, also enable "**Events on connected accounts**".
- 5 Save, copy the **signing secret** (`whsec_...`), and paste it into the POS **Stripe Terminal Webhook Secret** setting.

Firewall: The webhook URL must be reachable from the public internet, and whatever proxy/load balancer fronts your site must *not* strip the `Stripe-Signature` header — the webhook is rejected without it.

9. Card readers (Stripe Terminal)

Indition works with two different families of Stripe reader, and they are set up in **completely different ways** — follow the section for the reader you actually have:

- **Smart readers (countertop, Wi-Fi)** — the **Stripe Reader S700** (recommended) and the **S710**. They join your Wi-Fi and are added from the Indition web admin with a pairing code. **Use sections 9.1–9.3.**
- **Mobile / Bluetooth readers** — the **Stripe Reader M2** (and BBPOS Chipper 2X / WisePad 3). These are battery readers that pair over **Bluetooth** to the **Indition companion app** on a phone or tablet. There is **no Wi-Fi setup and no pairing code**, and they are **not** added from the web admin. **Use section 9.4.**

The browser register can only drive **smart** readers. The **M2 and other Bluetooth readers work only through the Indition companion app** — they cannot be paired or used from the web POS.

9.1 Smart reader — power on & connect Wi-Fi (S700 / S710)

- 1 Plug the USB-C cable into the **left side** of the reader, into the included 12 W power brick, into a wall outlet.
- 2 Hold the **power button** on the right edge until the screen lights up (first boot ~30 seconds).
- 3 Swipe in from the **left edge** → **Settings**. Enter the admin passcode **07139**.
- 4 Tap **Wi-Fi**, choose the store network, enter the password. Wait until the status bar shows the Wi-Fi icon and **Online**.

Captive-portal Wi-Fi (hotel/guest networks with a login page) will block the reader. Use the store's main network.

9.2 Smart reader — generate a pairing code (S700 / S710)

The pairing code is three words that prove the physical reader is yours. It expires after about **30 seconds**, so do the next step right away.

- 1 Still in **Settings**, tap **Generate pairing code**.
- 2 Note the three words (e.g. `apple-pear-mango`).

9.3 Smart reader — add it in Indition (S700 / S710)

- 1 Go to **POS → Card Readers → + Add new**.
- 2 **Store / dealer** — pick the store this reader lives at (this can't be changed after creating it).
- 3 **Reader name** — a label you'll recognise in a list (e.g. "Front Counter S700").
- 4 (Optional) **Where in the store** — an internal note like "Lane 1".
- 5 **Pairing code** — type the three words exactly (order matters; case doesn't).
- 6 Tick **Set as default reader** if it's the store's only/main reader.
- 7 Click **Save**. The system registers the reader with Stripe and fills in its reader ID (`tmr_...`), serial number, and device type (e.g. `stripe_s700`). The reader screen switches to a "Ready" view and shows a green **Active** badge in the list.

No physical test card yet? Stripe offers a **simulated reader** in test mode that lets you run the whole flow without tapping a card. Ask Indition support to enable a simulated reader for your account, then pair it as a non-default reader and run a test sale through it.

9.4 Mobile reader — pair a Stripe M2 in the companion app (M2 / Chipper 2X / WisePad 3)

Bluetooth readers are paired **inside the Indition companion app**, on the phone or tablet that will run the register — not from the web admin, and with no pairing code. Before you start: the app is installed and signed in, the device's store has its **Stripe Terminal Location ID** (`tml_...`) set (section 8.1), and the M2 is charged.

- 1 In the app, complete **Station Setup** so the device is tied to the store (this pulls the store's Stripe Terminal Location, which the reader connects to).
- 2 When prompted, grant the app **Bluetooth** and **Location** permissions. **Both are required** — on iOS the reader will not connect (and the app may crash) without Location permission.
- 3 Power on the M2 and put it in pairing mode (hold its button until the light blinks).
- 4 Open the app's **reader picker** and start a scan. The app discovers nearby readers over Bluetooth and lists them by **serial number**; tap yours.
- 5 The app connects the reader to the store's Stripe Terminal Location and registers it to the station. It then appears under **POS → Card Readers** with device type `stripe_m2`, using its **serial number** as its ID — Bluetooth readers have no `tmr_...` ID.
- 6 After this first pairing the app **auto-reconnects** the reader whenever the cashier signs in, so there is nothing to repeat day to day.

A station has **one reader at a time** — pairing an M2 to a station replaces whatever reader was on it. Keep the M2 charged; it is battery-powered.

10. Multi-store & Stripe Connect

For a single store on one Stripe account, skip this section. For multi-tenant setups where each store is paid out to its own Stripe account, the system routes each charge to the right place automatically. Routing is now consistent across **every** card rail — the on-screen card flow, the S700/S710 smart readers, the M2 Bluetooth reader (mobile app), and keyed / manual entry — through one canonical resolver, keyed by the store on the cashier's open session:

- When a store has an approved **Connected Stripe Account**, every Stripe call for that store (connection token, payment intent, reader processing, refunds) is made on that account (`Stripe-Account: acct_...`) so the money lands there. Reads and refunds ride the same account as the original charge.
- When a store has no connected account, its **own per-store keys** are used (or the platform key when nothing store-specific is configured).
- If a store has a connected account that **isn't approved yet** to accept card payments, card sales are **cleanly declined** with a clear message until approval completes — the sale is never charged against the wrong (platform) account. Cash and check are unaffected.

Verify routing in Stripe Dashboard → **Developers** → **Logs** — confirm each call ran against the expected account.

Resolved this release: every POS card leg — including the PaymentIntent creation, not just the reader processing — now runs on the store's connected account via the canonical resolver. The M2 Bluetooth reader in particular now creates its PaymentIntent server-side on the connected account, fixing a prior "invalid payment method" failure on Connect stores. Still worth a per-store end-to-end test sale in test mode before go-live to confirm the expected account in the Stripe logs.

11. Cash drawer & receipt printer

Most cash drawers connect to the receipt printer by an RJ-12 ("DK-Port") cable and open when the printer fires an ESC/POS pulse. Because a browser can't drive USB hardware directly, a small **drawer agent** runs on the register computer. The register sends it a small HTTP request; it pulses the drawer.

11.1 Install a drawer agent

Any of these work — pick what suits the computer:

Option	Good for
Node helper (CUPS / direct USB)	macOS & Linux registers. A tiny Node script prints the raw pulse bytes through the printer.
QZ Tray (<code>qz.io</code>)	The most common browser-to-printer bridge across all platforms.
Windows POS utility (OPOS / StarPRNT)	Windows registers with Star or OPOS-class printers.

Whatever you use, the agent must listen for an HTTP `POST` and answer with permissive CORS headers (`Access-Control-Allow-Origin: *`, allow `POST`, `OPTIONS`, allow `Content-Type`; answer preflight `OPTIONS` with 204). Localhost (e.g. `http://127.0.0.1:9100`) is recommended.

Reference implementations (Node, QZ Tray, Windows) and the exact ESC/POS pulse bytes are documented in the module's `docs/POS_DRAWER_AGENT.md`. Pre-built helper projects also exist alongside the site as `indition-pos-drawer-agent` and `indition-pos-receipt-agent`.

11.2 Point POS at the agent

- 1 In **POS → Settings → Cash Drawer**, turn on **Enable Drawer Kick**.
- 2 Set **Drawer Agent URL** to the agent's address (e.g. `http://127.0.0.1:9100`).
- 3 Leave **Auto-kick on cash sales** on so the drawer opens automatically on cash payment. (Pay-in/out, drops and pickups always kick.)

If the agent is unreachable, the register simply shows a warning toast — it never blocks a sale.

11.3 Receipt printer

Receipts print through the browser's standard print dialog (the register hides its own UI and prints only the receipt). Set the receipt printer as the computer's default, or as the chosen printer in the browser dialog, and load receipt paper. The header/footer text comes from POS Settings; the store name, logo, and address come from the store record.

12. Touch POS & the register computer

- **Browser** — any current Chrome, Edge, Safari, or Firefox. Run it full-screen/kiosk for a clean register.
- **Barcode scanner** — any USB "keyboard-wedge" scanner works; no driver needed. The register listens for scans automatically.
- **Bookmark** `/pos/` and have cashiers tick "Keep me signed in on this terminal" on trusted machines.
- **Network** — the register needs to reach your Indition site; the card reader needs to reach Stripe. Both must be online for a card sale to finish (if the register drops offline mid-charge, the sale resolves as soon as it reconnects — nothing is lost).
- **Accessibility** — text size and light/dark theme are set per computer from the top bar and remembered.

12.1 Touch POS UI (`/web/pos`)

A touch-optimised register is available at `/web/pos` for tablets and touch terminals. Enable it under **POS → Settings → Touch UI**:

- **Enable Touch UI** (`POS_TOUCH_UI_ENABLED`) — switches on the touch-optimised layout.
- **Auto-enable for device classes** (`POS_TOUCH_AUTO_DEVICES`) — automatically serve the touch UI to matching devices (e.g. tablets), so a tablet lands on the touch layout while a desktop keeps the standard register.

The touch UI shares the same catalogue, customers, inventory, payments, and settings as the standard register — it's a layout switch, not a separate system. Pair Bluetooth readers (Stripe M2) with the touch UI on tablets; remember the M2 needs location permission on the device.

13. Test sale & go-live

13.1 Run a test sale (test mode)

- 1 Sign in to **/pos/**, open a shift, and add any **\$1 item**.
- 2 Click **Card → Charge**. The screen shows "Sending to reader..." then "Follow prompts on reader...".
- 3 On the reader, tap a Stripe **test card** (e.g. `4242 4242 4242 4242`) — or drive the simulated reader.
- 4 The register lands on **Approved** and creates the sale.
- 5 Confirm it recorded: the sale appears under **POS → Recent Sales**, and in **POS → Reports → Daily Summary** (and **Sales Analytics**) for today, showing the card payment and the correct total.
- 6 Check your Stripe Dashboard → **Developers → Webhooks** → your endpoint shows recent deliveries all **200 OK**.
- 7 Refund the test sale from **POS → Recent Sales** (or at the register via **F9**) to clean up.

13.2 Switch to live

- 1 In Stripe, toggle to **Live mode** and create a live Terminal Location, live API keys, and a live webhook endpoint (with the same events). Copy the live `tml_...`, `sk_live_...`, `pk_live_...`, and `whsec_...`.
- 2 Update the POS settings and the store's Terminal Location ID with the **live** values.
- 3 Re-pair the physical reader (a reader paired in test mode must be re-registered against live keys; the simulated reader does not exist in live mode).
- 4 Run one real **\$1 sale** on a card you can refund immediately, then refund it.

13.3 Go-live checklist

- ☐ POS module enabled and at least one store selected
- ☐ Store flagged *Is POS Location* with its live Stripe Terminal Location ID

- ❑ Live Stripe secret, publishable, and webhook-signing keys saved
- ❑ Webhook endpoint live, reachable, all deliveries 200 OK
- ❑ Reader paired against live keys, showing Active, set as default
- ❑ Staff assigned POS roles (and stores, if multi-store)
- ❑ Cash drawer agent running and Drawer Agent URL set (if using a drawer)
- ❑ Receipt printer default-set with paper loaded; header/footer configured
- ❑ Card processing fee, tax, and payment methods set as intended
- ❑ Inventory options reviewed (block oversell, override + cost basis) and per-store warehouse/locations set, if using Inventory
- ❑ Email/SMS receipt sender details set, if using emailed or texted receipts
- ❑ Screen lock & PIN reviewed (idle timeout, roles) if using it on shared terminals
- ❑ Multi-store: each store's card routing confirmed against the expected Stripe account in the logs (payment-intent creation, not just reader processing)
- ❑ One live \$1 test sale completed and refunded

14. Troubleshooting

Symptom	Cause & fix
Reader never shows "Online"	Wi-Fi unreachable or captive-portal network. Use the store's main network; power-cycle the reader if stuck "Connecting" over a minute.
"Pairing code expired" on Save	The code is valid ~30 seconds. Tap Generate pairing code again and save the admin form immediately.
Reader is Active but the register can't reach it	Reload <code>/pos/</code> . Confirm the reader is ticked as default for the store under POS → Card Readers.
Reader says "No Location" / "Setup incomplete"	The store record is missing its Stripe Terminal Location ID. Paste the <code>tml_...</code> into the store and save (section 4.1 / 8.1).
"Store is not flagged as a POS Location" on first sale	Set <i>Is POS Location</i> = ON on the store record.
Sale completes on the reader but the register hangs on "Follow prompts..."	The register lost internet. It resolves automatically when the connection returns; nothing is lost.
Device type isn't <code>stripe_s700</code> after pairing	Older firmware or a different SKU. Note the returned device type for support; the server-driven path keys off the expected reader types.
Webhook deliveries failing at Stripe	Public URL not resolving, or a proxy stripping the <code>Stripe-Signature</code> header. Confirm the URL and proxy config.
"Card payment is not configured" on the payment screen	Stripe keys not set. Add the secret + publishable keys in POS Settings, then connect a reader or enable Manual Card Entry.
Connect store: money on the wrong account	Verify the connected account is approved and check Stripe → Developers → Logs that calls carry <code>Stripe-Account</code> (see section 10).
Drawer doesn't open	Confirm the drawer agent is running, the Drawer Agent URL is correct, and Enable Drawer Kick is on. A warning toast (not a blocked sale) means the agent is unreachable.

15. Appendix: settings reference

All settings live under **POS → Settings**. Defaults shown are the out-of-the-box values.

Setting key	Default	Purpose
<code>POS_MODULE_ENABLED</code>	0	Master on/off switch.
<code>POS_ENABLED_STORE_IDS</code>	[]	Stores where the register is active.
<code>POS_TAX_OVERRIDE_ENABLED</code>	0	Use a flat rate instead of live tax quoting.
<code>POS_DEFAULT_TAX_RATE</code>	0	Flat tax rate (%) when override is on.
<code>POS_STRIPE_SECRET_KEY</code>	(empty)	Stripe secret key (server-only). Retired for charge routing — each store's card charges now route via its connected Stripe account or its own per-store keys (section 10). Retained for backward compatibility.
<code>POS_STRIPE_PUBLISHABLE_KEY</code>	(empty)	Stripe publishable key (keyed-entry fallback).
<code>POS_STRIPE_TERMINAL_WEBHOOK_SECRET</code>	(empty)	Webhook signing secret (<code>whsec_...</code>).
<code>POS_CARD_FEE_ENABLED</code>	0	Add a card processing fee.
<code>POS_CARD_FEE_PCT</code>	2.9	Card fee percent.
<code>POS_CARD_FEE_FIXED</code>	0.30	Card fee fixed amount (\$).
<code>POS_PAYMENT_CASH_ENABLED</code>	1	Show the Cash button.
<code>POS_PAYMENT_CARD_ENABLED</code>	1	Show the Credit/Debit button.
<code>POS_PAYMENT_CHECK_ENABLED</code>	0	Show the Check button.
<code>POS_PAYMENT_SPLIT_ENABLED</code>	0	Allow split tenders.
<code>POS_PAYMENT_MANUAL_CARD_ENABLED</code>	0	Allow keyed card entry.
<code>POS_MANUAL_CARD_REQUIRE_CUSTOMER</code>	1	Require a customer for keyed cards.
<code>POS_MANUAL_CARD_REQUIRE_ZIP</code>	0	Require billing ZIP (AVS) on keyed cards.
<code>POS_SAVED_CARDS_ENABLED</code>	0	Show saved-card buttons / "save card" option.
<code>POS_USE_SHIFTS</code>	1	Require opening a shift before selling.
<code>POS_REQUIRE_DRAWER_COUNT</code>	1	Require an opening float at shift open.
<code>POS_ALLOW_CASHIER_SELF_VOID</code>	0	Cashiers may void own sales in their open shift.

POS_BLOCK_OVERSELL	1	Prevent selling out-of-stock items (manager override).
POS_ALLOW_INVENTORY_OVERRIDE	0	Allow selling out-of-stock tracked items; receive the shortfall on completion.
POS_INVENTORY_OVERRIDE_COST_BASIS	last	Cost basis for override receives: last / average / zero .
POS_STORE_GATING_ENABLED	0	Require store assignment to bind a register.
POS_DRAWER_KICK_ENABLED	0	Master switch for the cash-drawer agent.
POS_DRAWER_AGENT_URL	(empty)	Where the register POSTs drawer-kick requests.
POS_DRAWER_KICK_ON_CASH	1	Auto-open the drawer on cash sales.
POS_TIPPING_ENABLED	0	Show the tip prompt on card payments.
POS_TIP_PRESETS	15,18,20	Tip preset percentages.
POS_PRICE_LEVEL_ID	0	Price level (0 = use Shop default).
POS_PRODUCT_STATUS_FILTER	sellable, active	Which product statuses cashiers can find.
POS_ENABLED_CATEGORY_IDS	[]	Limit search to certain categories (empty = all).
POS_SEARCH_LIMIT	200	Max search results (0 = all; server caps at 5,000).
POS_PRODUCT_IMAGE_TAG	[]	Gallery image tags shown in product detail.
POS_QUICK_KEY_SKUS	[]	One-tap product buttons (up to 24).
POS_RECEIPT_HEADER	(empty)	Text at the top of receipts.
POS_RECEIPT_FOOTER	Thank you for your business!	Text at the bottom of receipts.
POS_EMAIL_RECEIPT_ENABLED	0	Offer emailing the receipt (PDF attached).
POS_RECEIPT_EMAIL_TEMPLATE	(default)	Email template for receipts.
POS_RECEIPT_EMAIL_SUBJECT	(default)	Receipt email subject line.

POS_RECEIPT_EMAIL_FROM_NAME	(empty)	Receipt email sender name.
POS_RECEIPT_EMAIL_FROM_ADDRESS	(empty)	Receipt email sender address.
POS_RECEIPT_EMAIL_BCC	(empty)	Optional BCC on receipt emails.
POS_RECEIPT_EMAIL_BODY	(default)	Body text of the receipt email.
POS_RECEIPT_EMAIL_ATTACH_PDF	1	Attach the receipt PDF to the email.
POS_SMS_RECEIPT_ENABLED	0	Offer texting the receipt to the customer.
POS_FRACTION_QUICKPICK_ENABLED	0	Show fraction quick-pick buttons for unit-sold items.
POS_FRACTION_PRESETS	(e.g. 1/4, 1/2, 3/4)	Fraction quick-pick presets.
POS_FRACTION_ALLOW_DECIMAL	1	Allow typing a decimal quantity.
POS_TOUCH_UI_ENABLED	0	Enable the touch-optimised register at <code>/web/pos</code> .
POS_TOUCH_AUTO_DEVICES	(empty)	Device classes auto-served the touch UI.
POS_PIN_UNLOCK_ENABLED	0	Master switch for the screen lock / PIN unlock (section 4.3).
POS_LOCK_IDLE_TIMEOUT_SECS	300	Auto-lock after N seconds idle (0 = off).
POS_LOCK_INTERVAL_SECS	0	Force re-lock N seconds after unlock (0 = off; deferred mid-sale).
POS_PIN_MIN_LENGTH	4	Minimum PIN length.
POS_PIN_ROTATION_DAYS	90	Days before a PIN must be reset (0 = never).
POS_PIN_MAX_ATTEMPTS	5	Wrong PINs before a forced full re-login.
POS_PIN_LOCK_ROLES	cashier, manager, admin	POS roles the screen lock applies to.

The **per-store warehouse & inventory location selection** (section 5.3) is not a single setting key — it is configured under each enabled store in **POS → Settings → Stores** and stored per store/location.

