HOTEL & RESTAURANT POINT OF SALE

HYDRA POS

INTEGRATION GUIDE

Integration Guide

Step-by-step bring-up of every external connection: PMS, NTAK, printers & KDS, inventory, accounting, e-mail, the Manager and Waiter apps, multi-terminal sync and the Hydra Admin cloud.

PMS · NTAK · Printers/KDS · Inventory · Apps · Cloud

Version 5.0 · 2026

Contents

1 Introduction

This guide describes how to **bring up the integrations of Hydra POS** — the outside systems the till connects to. Day-to-day operation (taking orders, payment, day close) is covered in the **User Manual**; here we focus solely on getting each connection — the **PMS**, **NTAK**, **printers & KDS**, **inventory (INV)**, **accounting (ACT)**, the **daily-close e-mail**, the **Manager** and **Waiter** apps, **multi-terminal sync** and the **Hydra Admin** cloud — to come up correctly the first time.

Some steps below require an external account or an official registration (NTAK, your SMTP provider). Each chapter states exactly which prerequisites must be in place before you start.

> **IMPORTANT**
> Order matters. Do the common setup in the next chapter (outlet identity, licence, terminal role) **first** — every integration builds on it. The Manager/Waiter pairing codes only appear after a successful Hydra Admin check-in, and a client terminal cannot sync until the server terminal is configured.

1.1 Who this is for

This guide is written for the venue's **system administrator** or the **Hydra installer** — the person who has access to the Settings screens and to the venue's official registrations. You do not need to be a developer, but you should be comfortable finding a PC's IP address and opening a firewall port.

1.2 Conventions

The following conventions are used consistently so that on-screen labels, values and warnings are always distinguishable at a glance.

- **Bold** — an English term or a key idea to remember.
- On-screen menus, buttons and field labels are quoted in **Hungarian** (the app's default language) exactly as they appear, with an English gloss in parentheses — e.g. *Beállítások* (Settings).
- `code` — a value to type, a URL, a port, a file name or a field id.
- Block-quoted call-outs (below) mark notes, tips and warnings.

> **NOTE**
> A plain note: useful context that isn't a warning.
>
> **Tip.** A shortcut or a recommended practice.
>
> **Warning.** Something that can lose data or money if you get it wrong.

1.3 Two golden rules

These two principles underpin every integration in this guide. Learn them before you touch a single setting.

TIP

Test first, then go live. NTAK offers a separate sandbox, and the PMS link can be exercised against a test charge before real money moves. Always confirm a connection in test mode, then switch to production.

Warning. Secrets are per-device. The NTAK certificate password, the SMTP password, the licence key, the PMS API keys and the multi-terminal sync secret are **not** synchronised between terminals — set them on each machine (see *Prerequisites → Per-device secrets*). They are stored encrypted at rest and never travel in a backup or a sync message.

2 Prerequisites & common settings

Before you connect a single outside system, the till itself has to know **who it is**, **that it is licensed**, and **what role it plays** on the network. This chapter covers that common groundwork — the outlet identity, the licence, the terminal role, and the per-device secrets — that every later chapter builds on. Do it once, get it right, and the rest of the guide falls into place.

> **IMPORTANT**
> Order matters. Set the outlet identity, the licence and the terminal role here **first**. The Manager and Waiter pairing codes only appear after a successful Hydra Admin check-in, and a client terminal cannot sync until the server terminal is configured. See the *Hydra Admin* and *Multi-terminal sync* chapters.

2.1 Where the settings live

The integration settings open from **Settings** on the left sidebar. Opening Settings is **manager-gated** — it may ask for a manager PIN — so have a manager account ready before you start.

Most of the work in this guide happens on one tab: *Beállítások* (Settings) → *Kapcsolatok* (Connections). That tab gathers the four connection blocks you will return to again and again:

Block	What it configures
Hydra Admin & Licensz	Cloud licence, off-site backups, the live Manager dashboard
Hydra PMS kapcsolat	Charging a sale to a guest room or service account
Hydra ACT	Forwarding the day-close to accounting
Leltár szinkronizáció (Inventory)	Stock decrement and reversal

A few integrations live on their own tabs rather than under *Kapcsolatok*:

- *NTAK* and *Validáció* (Validation) — Hungarian tax reporting and its test suite.
- *Nyomtatás & KDS* (Print & KDS) — the local printer and kitchen stations.
- *Email riport* — the daily-close e-mail report.

Multi-terminal operation is configured from the *Terminál szinkron* (Terminal sync) panel, together with the **terminal number** (*Terminál szám*) covered below.

> **TIP**
> Two codes you will need for the companion apps — the **Manager access code** and the **Waiter pairing code** — are **not** on these tabs. They live in *Rendszerinfó* (System Info), which you open by tapping the **Hydra POS** logo in the top bar. They only appear once Hydra Admin has checked in successfully — see the *Hydra Admin* chapter.

2.2 Outlet identity

Every venue has a public face and a cloud identity, and they are not the same thing. The display names are what guests and staff see; the **hotel id** is what the cloud uses to find this venue's record. Set them on *Kapcsolatok* → *Hydra Admin & Licensz*.

Field	Key	What it does
Szálloda neve (Hotel name)	`hotelName`	The venue's display name, shown in the header and on reports
Outlet name	`outletName`	The outlet's display name, printed on receipts and reports
Szálloda ID (Hotel ID)	`hotelId`	The venue's cloud identifier in Hydra Admin — the key the cloud uses to find this record
Licenc kulcs (Licence key)	`licenseKey`	This venue's `POS-…` licence key (a secret — see *Per-device secrets*)

The `hotelId` is the single most consequential value on this screen. It is what binds this physical till to one record in the Hydra Admin cloud — the licence, the backups and the Manager dashboard all hang off it.

> **WARNING**
> Set `hotelId` **once** at first bring-up, then leave it alone. Changing it after go-live points the venue at a **different** cloud record — its licence, its backup history and its derived Manager/Waiter codes — and the till will appear to belong to another venue.

2.3 Licence & activation

On start-up, and periodically thereafter, the POS validates its `licenseKey` with Hydra Admin. As long as the cloud confirms the licence, the till trades normally and you never see a thing.

When the cloud cannot be reached, the POS does **not** stop trading on the spot. It keeps going on a **7-day grace period**, measured from the **last successful check-in**. Within that window everything works as usual; only when the grace period runs out does the till show a **licence wall** that blocks trading until a successful check-in restores it.

A licence that has **never** been checked, or one that the cloud reports as **expired**, shows the same wall straight away — there is no grace for a licence that was never valid.

> **NOTE**
> A licence wall is almost always a connectivity or configuration problem, not an expiry. The usual culprits are a wrong `adminUrl` or `licenseKey`, or a clock/network issue that keeps the check-in from completing. The *Hydra Admin* chapter covers how to read the check-in state in System Info.

2.4 Terminal role

Each till runs in one of three roles, and the role decides how it shares data with the others:

- **server** — owns the shared state and serves the clients.
- **client** — mirrors the server in real time.
- **standalone** — a single, self-contained till that syncs with no one.

Whatever the role, every terminal also carries a unique **2-digit terminal number** — `01`, `02`, `03`, and so on — set on the *Terminál szinkron* panel. The terminal number is stamped into receipt numbers so two tills can never issue the same one.

> **IMPORTANT**
> Configure the **server** terminal before any client. A client has nothing to connect to until the server is up, named and numbered. Full step-by-step setup is in the *Multi-terminal sync* chapter; this is only the overview you need to plan the bring-up order.

2.5 Per-device secrets

This is one of the two golden rules of the whole guide, so it is worth stating plainly: **secrets are per-device**. The values below are credentials, and Hydra POS treats them accordingly. They are **never synchronised** between terminals, they **never** travel in a cloud backup or a sync message, and they are stored **encrypted at rest** using the operating system's keychain (OS keychain on macOS, **DPAPI** on Windows, **libsecret** on Linux). Only if no keyring is available does the POS fall back to plaintext.

The practical consequence: each secret must be entered **on every terminal that needs it**. Copying a backup or pairing a client will not bring them across.

Secret	Key	Where it is set
Licence key	`licenseKey`	*Kapcsolatok* → *Hydra Admin & Licensz*
NTAK certificate password	`ntakCertPassword`	*NTAK* tab
SMTP / e-mail password	`emailPass`	*Email riport* tab
PMS API key	`pmsApiKey`	*Kapcsolatok* → *Hydra PMS kapcsolat*
POS inbound API key	`posInboundApiKey`	*Kapcsolatok* → *Hydra PMS kapcsolat*
Multi-terminal sync secret	—	*Terminál szinkron* panel

> **WARNING**
> Because these secrets never sync, a freshly paired client terminal will look configured but **fail silently** on anything that needs one of them — NTAK reporting, the day-close e-mail, the PMS link. After adding a terminal, walk this table and set each secret it needs.

3 Hydra Admin — cloud backup & licensing

Hydra Admin is the cloud service that the till checks in with. It does three jobs at once: it **validates the licence** so the POS may trade, it keeps periodic **off-site backups** of the venue's POS state, and it feeds the live **Manager dashboard**. Because the licence and the companion-app pairing codes both flow through it, Hydra Admin needs to be working before most of the rest of the guide will come together.

3.1 What it does

Job	What it means in practice
Licensing	The POS validates `licenseKey` against the cloud on start-up and periodically. A valid licence lets the till trade; a failed check counts down the grace period (see the *Prerequisites* chapter)
Off-site backup	The cloud stores snapshots of the POS state — slip history, closed days and counters — so the venue's records survive a lost or replaced terminal
Manager dashboard	While a manager is watching, Hydra Admin feeds the live dashboard with the venue's trading data

> **NOTE**
> A backup captures the **operational record** — slips, closed days and counters. It does **not** carry the per-device secrets (the licence key, the NTAK certificate password, the SMTP password, the PMS keys, the sync secret); those are encrypted at rest on each machine and never leave it. See the *Prerequisites* chapter.

3.2 Configure

Open *Beállítások* (Settings) → *Kapcsolatok* (Connections) → *Hydra Admin & Licensz*. These fields are manager-gated.

Field	Key	What it does
Admin URL	`adminUrl`	Base URL of the Hydra Admin service. Leave at the default unless your supplier gives you another
Licenc kulcs (Licence key)	`licenseKey`	This venue's `POS-…` licence key. secret — encrypted on this machine, never synced
Szálloda ID (Hotel ID)	`hotelId`	The venue's cloud identifier — the record the cloud opens for this till
Szálloda neve (Hotel name)	`hotelName`	The venue's display name, shown in the header and on reports
Auto-backup	`autoBackupEnabled` / interval	How often to back up: `Off` / `1` / `2` / `4` / `8` / `24` h. Default **4 h**

To bring Hydra Admin up on a terminal:

1. Enter the `adminUrl` your supplier provided, or leave the default in place.
2. Enter the venue's `licenseKey` (the `POS-…` value).
3. Enter the `hotelId` and `hotelName` for this venue.
4. Pick an **Auto-backup** interval. **4 h** is a sensible default; busy venues may prefer `1` or `2` h.

5. Restart or wait for the next check-in, then confirm the connection in System Info (below).

> **WARNING**
> The `hotelId` decides **which** cloud record this till reads and writes — its licence, its backups, its Manager codes. Set it once at first bring-up and do not change it after go-live, or the venue will point at a different record. See the *Pre-requisites* chapter.

3.3 How it works

The POS talks to Hydra Admin on two separate rhythms — a **licence check-in** and a **backup** — and they are not the same thing.

- **Check-in.** On start-up and periodically thereafter, the POS contacts Hydra Admin to verify the licence. A successful check-in resets the grace clock and refreshes the venue's derived codes.
- **Backup.** On the chosen **Auto-backup** interval and at **each day close**, the POS POSTs a snapshot of its state to `…/api/hotels/backup`. The day-close backup means a closed day is always pushed off-site promptly, regardless of the interval.
- **Grace.** If the cloud is unreachable, the POS keeps trading on a **7-day grace period** measured from the last successful check-in. Once that runs out it shows a licence wall until a check-in succeeds again. See the *Prerequisites* chapter for the full licence-wall behaviour.

> **NOTE**
> A backup reflects the state **as of the last snapshot**, so it can lag by up to the Auto-backup interval. A slip voided after a snapshot — a *stornó* — rides the **next** backup rather than appearing immediately. Pick a shorter interval if your venue voids frequently.

3.4 Where to watch it

System Info is the single place to confirm Hydra Admin is healthy. Open it by tapping the **Hydra POS** logo in the top bar. It shows:

Indicator	What it tells you
Admin URL	The `adminUrl` the till is actually using
Connection status	Whether the cloud is currently reachable
Last check-in	When the licence was last verified — the start of the grace clock
Backup state	Last success or failure, what **triggered** it, and the configured interval

System Info also carries a **manual backup** button, so you can push a fresh snapshot on demand — handy right after a day close, or before swapping hardware.

> **TIP**
> After first configuring a terminal, open System Info and confirm a successful check-in and a green connection status **before** moving on to the companion apps. If the licence is failing, fix it here first.

3.5 Manager & Waiter codes depend on this

The **Manager access code** and the **Waiter pairing code** are derived **server-side** from the venue's `hotel_id`, and they are delivered to the till **only after a successful check-in**. Until Hydra Admin is reachable and the licence validates, those codes are simply not present — which means the Manager and Waiter apps **cannot pair**.

In other words, Hydra Admin is a hard prerequisite for the companion apps. Get the check-in green here first; the codes then appear in System Info, ready for the *Manager app* and *Waiter app* chapters.

3.6 Common errors

Symptom	Likely cause	What to do
Licence wall on start-up	Wrong `adminUrl` or `licenseKey`	Re-check both against the values from your supplier; a single typo blocks the licence
Check-in keeps failing, grace counting down	Clock skew or no network path to the cloud	Fix the terminal's time and confirm it can reach the `adminUrl`; the grace clock keeps running until a check-in succeeds
Manager/Waiter apps will not pair	Hydra Admin has not checked in successfully	The codes only exist after a good check-in — fix the connection first, then read the codes in System Info
A voided slip is missing from the cloud	Backups lag by up to the interval	The void rides the next backup; trigger a manual backup from System Info if you need it off-site now

> **IMPORTANT**
> Almost every Hydra Admin problem is a licensing or connectivity problem, and System Info will tell you which. Start there: read the connection status, the last check-in, and the backup state before you touch any settings.

4 PMS — room & service charging

Hydra POS can charge a sale straight to a guest's **room folio** or to a **service account** held in the hotel's **PMS** (Property Management System), instead of taking cash or card at the till. The PMS exposes a small REST API on port `3000`, and the POS reaches it over the venue's LAN — typically from each till to the server-terminal that runs the PMS bridge.

The link runs in **two directions**, and each direction has its **own key**. Keeping them straight is the single most important thing in this chapter.

Direction	Who calls whom	Port	Header	Key
Outbound	POS → PMS (post a charge)	`3000`	`X-Api-Key`	`pmsApiKey`
Inbound	PMS → POS (notify a storno/void)	`3001`	`X-POS-Api-Key`	`posInboundApiKey`

> **WARNING**
> The two keys are not interchangeable. The POS presents `pmsApiKey` when it posts a charge; the PMS presents `posInboundApiKey` when it calls back to void one. Sending the wrong key on either leg returns `401` and the operation fails.

4.1 Preparation

Before you open the settings, gather the following so the bridge can come up on the first try.

Item	What it is for
PMS host & port	The LAN address of the machine running the PMS API, and its port (`3000` by default).
`pmsApiKey`	The outbound key the PMS expects from the POS in `X-Api-Key`. Obtain it from the PMS configuration.
`posInboundApiKey`	The inbound key the POS will require from the PMS on storno callbacks. You generate this — see below.
Advertised LAN IP	The IP the POS publishes to the PMS as its callback origin. The PMS must be able to reach the POS at this IP on port `3001`.

Generate the inbound key with a single command and keep it somewhere safe — you will paste the same value into the PMS so it can authenticate its callbacks:

```
openssl rand -hex
```

> **IMPORTANT**
> If `posInboundApiKey` is left empty, the POS rejects **all** inbound mutations with `401` — every storno callback from the PMS bounces. The key must be set on the POS *and* configured in the PMS before reversals can flow. The inbound key is compared in **constant time**, so a wrong key never leaks its correct length or contents through response timing.
>
> **Warning.** `pmsApiKey` and `posInboundApiKey` are **per-device secrets**. They are stored encrypted at rest and are **not** synced between terminals — set them on each machine that talks to the PMS.

4.2 Setup

Open *Beállítások* (Settings) → *Kapcsolatok* (Connections) → *Hydra PMS kapcsolat* (Hydra PMS connection) and fill in:

Field	What it does
`pmsEnabled`	Turns the PMS link on. Leave off if the venue has no PMS.
`pmsHost`	The PMS machine's hostname or LAN IP.
`pmsPort`	The PMS API port. Default `3000`.
`pmsApiKey`	Outbound key, sent in `X-Api-Key` on every charge.
`posInboundApiKey`	Inbound key the PMS must present in `X-POS-Api-Key` to void a charge.

The inbound status server and binding

When enabled, the POS runs a small **status/callback server** on port `3001`. This is what the PMS queries for the day's open/closed state and calls to notify a void. By design it does **not** listen on every interface — it binds to the POS's **advertised LAN IP**, the same address the POS publishes to the PMS as its callback origin, so the endpoint isn't exposed on unrelated networks.

For a PMS running on the **same host** as the POS, set the optional `bindAddress` override to `127.0.0.1` so the server binds to loopback only.

> **IMPORTANT**
> The PMS must be able to reach the POS at its **advertised IP** on port `3001`, or storno callbacks never arrive. If the till's IP is assigned by DHCP, give it a reservation so the callback address stays stable.

Price codes (Árkód)

Open *Beállítások* (Settings) → **PMS Árkódok** (PMS price codes). This maps each **VAT rate** to the PMS **GL account codes** (consumption and service-charge codes) so charges land on the correct ledger accounts in the hotel's books.

4.3 Testing

Charging to a room

1. At payment, choose **Szoba** (Room).
2. Enter the **room number** → the guest is looked up in the PMS and validated.
3. Pick the items to charge → the charge **posts to the folio**, identified by the POS **receipt number** (this is the key used for any later reversal).
4. A detailed **VAT receipt** prints, plus a **kitchen summary**.

Szerviz (Service account) works the same way, without the guest lookup — useful for in-house accounts that aren't tied to a specific room.

Reversals (PMS → POS)

A room or service charge is **reversed in the PMS**, never from the till. When that happens:

1. The PMS calls the POS inbound server on port `3001` with the **receipt number**, authenticating with `posInbound-ApiKey`.
2. The POS marks the payment **voided** (it shows as *"PMS sztornó"*).
3. Inventory is reversed if stock tracking is enabled.
4. The day's **stornó list** is updated.

> **IMPORTANT**
> Room and service charges **cannot** be voided from the POS — reverse them in the PMS, which then calls back. The "void" action you'd use for a normal cash payment does not apply here.

Reliability — no double-bills, no lost stornos

The bridge is built to survive a dropped network without double-charging a guest or stranding a void:

- **Idempotent charging.** The POS sends an `idempotencyKey` of the form `pos-room-<receipt>`. The PMS dedups on the receipt number, so a charge retried after a lost response cannot double-bill — the POS treats the duplicate reply as a successful delivery.
- **Idempotent stornos.** The POS storno endpoint is idempotent: a re-delivered void returns `200` (flagged as already-voided) and does not reverse inventory a second time.
- **Offline reconcile.** If the POS was offline when a storno was sent, it **reconciles missed stornos** on reconnect by polling `GET /api/pos-stornos?since=<ISO>` (with `X-Api-Key`) and applying any it missed — idempotently, so nothing is voided twice.

> **TIP**
> **Test before you trust it.** Post a small test charge to a known room, confirm it appears on the folio in the PMS, then reverse it in the PMS and watch the till mark the payment *"PMS sztornó"*. Both legs working end to end means both keys and both ports are correct.

4.4 Common errors

Symptom	Likely cause	What to do
Anyone on the LAN can hit the PMS API	`pmsApiKey` is empty	Always set `pmsApiKey` in production.
Storno callbacks rejected with `401`	`posInboundApiKey` is unset, or the PMS sends the wrong value	Set `posInboundApiKey` on the POS and configure the **same** key in the PMS.
Charges return `401`	Wrong `pmsApiKey`	Re-check the outbound key against the PMS configuration.
Charges or reversals never go through	Port `3000` or `3001` blocked by a firewall	Open `3000` (POS → PMS) and `3001` (PMS → POS) on the LAN.
Callbacks never arrive even with the right key	PMS cannot reach the POS's advertised IP on `3001`	Verify the advertised IP and routing; give the till a stable address.
"PMS nem elérhető" at charge time	The PMS is unreachable	The charge errors out — retry once the PMS is back.

> **NOTE**
> A guest who has already left, or a charge already reversed in the PMS, is not an error on the POS side — the inbound endpoint simply returns *not found* or treats the repeat as already-voided. Trust the PMS folio as the record of what was billed.

5 NTAK — tax data reporting

NTAK — *Nemzeti Turisztikai Adatszolgáltató Központ* (the National Tourism Data Supply Centre, run by MTÜ) — is the Hungarian authority that every hospitality venue must report to. Hydra POS submits two kinds of message for you, automatically: a **rendelésösszesítő** (order summary) when an order closes, and a **napi zárás** (day close) once per business day. Your job is to load the certificate and the registration numbers correctly, then watch the status dot.

> **IMPORTANT**
>
> NTAK reporting is **mandatory in Hungary**. If the venue operates anywhere else, leave NTAK disabled and skip this chapter — none of it applies.
>
> **Note.** The printed slip is **not** the legal record — the NTAK submission is. That is why receipts carry a "not a tax document" note by default.

5.1 What you need from the NTAK portal

Gather these from the **NTAK / MTÜ** registration portal (and your accountant) before you open the settings.

Item	What it is
Technical certificate	A `.pfx` or `.p12` file plus its **password**, generated from a CSR in the NTAK portal. Optionally a separate `.key` file.
Szoftver regszám	The software registration number for the POS.
Szoftver verzió	The software version **exactly as registered**, e.g. `5.0`.
Üzlet regszám	The venue registration number — 2 letters + 8 digits, e.g. `ET26004227`.
Adószám	The 11-digit tax number (digits only).
API base URL	Sandbox `https://rms.tesztntak.hu`, or production `https://rms.ntak.hu`.

You will also need every **product** to carry an NTAK **főkategória** (main category) and **alkategória** (sub-category) and the correct **VAT rate**, set in product management. Without these, an order cannot be categorised and NTAK will reject it.

> **WARNING**
>
> The certificate password is a **per-device secret**. It is stored encrypted and is **not** synced between terminals — if several tills submit to NTAK, load the certificate on **each** one.

5.2 Setup

Open *Beállítások* (Settings) → *NTAK* (manager-gated). Work through three steps.

1. **Alap beállítások** (Basic settings) — enter the **API base URL**, *Szoftver regszám* (software reg. number), *Szoftver verzió* (software version), *Üzlet regszám* (venue reg. number) and *Adószám* (tax number).
2. **mTLS tanúsítvány** (mTLS certificate) — pick the `.p12` / `.pfx` file and enter its **password** (stored **encrypted** on this machine). Add the optional `.key` file only if your private key is separate.
3. **Enable** — turn on *NTAK adatszolgáltatás bekapcsolva* (NTAK data reporting enabled).

> **TIP**
> **Validate before you trust production.** Don't switch the base URL to production until the sandbox tests pass — see *Validation & go-live* below.

5.3 How reporting works

You normally do nothing — submission is automatic.

1. **An order closes** → a *rendelésösszesítő* (order summary) is **queued** (not sent instantly).
2. About every **15 minutes** the queue is **flushed** to NTAK in a batch.
3. NTAK accepts each item (status *befogadva*, accepted) and returns a **process id**.
4. The POS **polls** the status (~every 15 minutes) until each item is *sikeres* (success) or *sikertelen* (failed).
5. At **day close**, the *napi zárás* (day close) message is queued and sent the same way.

Messages travel over **mutual-TLS** (the client certificate identifies the venue) and are **RS256 detached-JWS** signed (the signature proves integrity) — all handled for you.

> **NOTE**
> NTAK rejects timestamps that are in the **future** relative to its own clock. The POS measures the server's time on the first response and **auto-adjusts** all timestamps to stay within tolerance, so a slightly wrong PC clock won't cause rejections. Keep the terminal's clock roughly correct anyway — a large skew can still fail.

The status dot

A small **status dot** in the top bar reflects the health of the NTAK queue.

Dot	Meaning
Green	Healthy — small queue, nothing stuck.
Amber	Pending items, or a few failures — watch it.
Red	Backlog or repeated failures — open the panel and act.

Tap the dot for a per-item panel showing each item's status, age, attempt count and last error.

5.4 Validation & go-live

Open *Beállítások* (Settings) → *Validáció* (Validation). It runs the official NTAK **T-tests** (order summaries), the **Z007** series (day close) and the **R006** idempotency check. Work through it in order:

1. Point the **API base URL** at the **sandbox** (`https://rms.tesztntak.hu`).
2. Load your **test certificate**.
3. Run the tests and confirm each reaches **success**.
4. **Export the protocol** (*Jegyzőkönyv exportálás*) — MTÜ requires it for certification.
5. Switch the **API base URL** to **production** (`https://rms.ntak.hu`) and load your **live certificate**.

> **IMPORTANT**
> The *Szoftver verzió* (software version) must match **exactly** what was registered with MTÜ — not the app's build number. A mismatch fails validation.

5.5 Common errors

Symptom	Likely cause	What to do
Items *sikertelen* (failed)	A mis-categorised product or invalid data	Tap the dot, read each item's error, fix the product's NTAK category/VAT, then re-send.
"Privát kulcs nem található" / *"Cert nem olvasható"*	Certificate/key not loaded or wrong password	Re-select the `.p12` / `.key` and re-enter the password.
Items stuck **pending** for hours	NTAK unreachable, or the certificate rejected at the TLS layer	Check the internet connection; verify the certificate and base URL.
Future-date / time errors	The PC clock is far off	Sync the terminal clock (NTP). The POS auto-adjusts within tolerance, but large skews still fail.
Validation test fails	Wrong base URL, test certificate, version or category data	Confirm the **sandbox** URL, the **test** certificate, the **registered** version and the product categories.

TIP

Mis-categorised products are the most common cause of *sikertelen* items. Review each product's **főkategória/alkategória** and **VAT rate** before go-live, and the queue will stay green.

6 Printers & KDS

Most integrations in this guide reach out across the network to a single, named service. Printing is the exception: a busy outlet has **several** output destinations at once — the slip printer at the till, a thermal printer behind the bar, the kitchen screens on the pass — and Hydra POS has to send the right ticket to the right place every time. This chapter explains the three kinds of destination, how to set each one up, and how to read a print failure when one slips through.

6.1 The three destinations

Hydra POS prints to three kinds of destination, and it helps to keep them straight from the start because each is configured differently:

Destination	What it is	How it's reached
System printers	Any printer installed on the terminal's operating system	The OS print pipeline (by printer name)
Network printers	Thermal printers speaking **ESC/POS**	Raw TCP to an `IP:port`, on port `9100`
KDS stations	Kitchen screens or printers, fed by product category	A station record matched to a terminal

A **system printer** is anything the till's OS already knows about — a USB slip printer, a shared office printer. A **network printer** is a thermal printer with an Ethernet port that listens for ESC/POS commands directly. A **KDS station** is a kitchen destination — a screen or a printer — that receives only the products whose category is routed to it.

6.2 Receipt (slip) printing

The slip the guest receives is printed by **this terminal's** local system printer. Two settings control it, and they live on different tabs.

The printer itself is chosen on *Beállítások* (Settings) → *Nyomtatás & KDS* (Print & KDS) → *Helyi nyomtató* (Local printer), which writes the `localPrinterName` value. This is the system printer that slips go to on this machine.

The slip's **shape** — its paper width and font size — is set separately on *Beállítások* → *Bizonylat & Fizetés* (Receipt & Payment):

Setting	Key	Options
Slip width	`slipWidth`	`58` / `76` / `80` mm
Font size	`slipFontSize`	*Kis* / *Közepes* / *Nagy* (small / medium / large)

The width affects how text wraps, so set it to match the paper actually loaded in the printer. The two settings are deliberately on different tabs: the printer is a property of **this device**, while the slip layout is a property of the **outlet**.

6.3 Printer profiles

A single till can drive several printers at once, and **printer profiles** are how you map products to them. A profile assigns one or more **product categories** to a specific destination — either a **system printer** or an `IP:port` — so that, for example, drinks print at the bar while food prints in the kitchen.

Each profile carries:

- a **name** (so staff can recognise it),
- the **destination** — a system printer **or** an `IP:port`,
- the **product categories** it serves, and
- a **default** flag (the fallback for anything not otherwise routed).

Routing by category is what lets one order split cleanly across the venue: the guest's slip prints at the till, the cocktail line prints at the bar, and the food line prints in the kitchen — all from the same payment.

> **TIP**
> Give every venue at least one **default** profile. Anything whose category is not explicitly routed falls through to the default, so a missing default means a silently un-printed line.

6.4 KDS stations

Kitchen Display System (**KDS**) stations are configured on *Beállítások → KDS állomások* (KDS stations). A station is a kitchen destination — a screen or a printer — that receives only the tickets routed to it. Each station has:

- a **name** (e.g. *Meleg konyha* — hot kitchen, or *Desszert* — dessert),
- a **category filter** (which products route to it),
- an **active** toggle, and
- an optional **printer**.

The key idea that makes a multi-terminal kitchen work is **machine matching**: a station only prints on the machine it is assigned to, matched by the terminal's **machine name** / terminal number. So the hot-kitchen station prints on the hot-kitchen screen, and the dessert station prints on the dessert screen, even though every till sees the same order. (The machine name is set on *Beállítások → Általános → Gép azonosítója* — Machine ID.)

A **sound** alerts the kitchen to new tickets so a busy line never has to watch the screen. Configure it on *Beállítások → KDS hangjelzés* (KDS sound): an on/off toggle plus a volume control.

> **NOTE**
> A station's **category filter** and a **printer profile**'s categories are independent. A product can appear on a kitchen screen (via its KDS station) and also print a paper ticket (via a printer profile) — they are two separate routes, not one.

6.5 Previewing and testing a slip

Before a shift you can confirm the slip looks right without printing a live order. On *Beállítások → Bizonylat & Fizetés → 👁 Slip előnézet* (Slip preview), the POS renders a **sample slip** on screen using the current width and font settings. The preview's **Nyomtat** (Print) button then sends a **test print** to the configured printer — the quickest way to prove the paper, the width and the cut are all correct.

6.6 How a print is delivered

How a slip actually reaches paper depends on where the POS is running.

On **Electron** (the desktop app), receipts are sent to the OS print pipeline **silently** — no dialog, the slip just prints. In a **browser preview**, the POS has no silent path, so it falls back to the browser's print dialog via a hidden frame.

Network ticket and receipt printing is different again: it speaks **raw ESC/POS over TCP** directly to the printer, with no OS driver in between. The byte stream includes the printer init, the **Hungarian code page** (so accented characters render correctly), alignment, **bold** and double-size text, the paper **cut**, and a **cash-drawer kick**. The connection has a **5-second timeout**, so an unreachable printer fails fast rather than hanging the till.

6.7 When printing fails

A failed print is not silent. A **toast** appears on screen, and **System Info** keeps a record of the **last 10 print failures** — each entry showing the **kind** of print, the **printer**, and the **error**. Reviewing that list is the fastest way to see a pattern (one printer always failing, say, versus an intermittent network issue).

The common causes are short:

Symptom	Likely cause
Network printer never prints	Wrong `IP` in the profile, or the printer is offline
Times out after 5 seconds	Port `9100` blocked between the till and the printer
Nothing prints, no error	No **default** printer profile for that category

> **WARNING**
> Network printing needs an unobstructed path to port `9100`. A firewall, a VLAN boundary or a guest-network segment between the till and the printer will make every ESC/POS job time out — even though the printer is powered on and reachable by ping from elsewhere.

7 Inventory — Hydra INV

A till that takes orders but never tells anyone what it sold leaves the stock count to guesswork. The **inventory** integration closes that loop: when an item sells, its stock is **decremented** at an external inventory service, and when a sale is voided, the stock is put back. Done right, the bar and the kitchen always know what is actually on the shelf — and the kitchen screen warns the moment a product runs low.

7.1 What it does

The inventory link does two things, automatically and in real time:

- On a **sale**, the items sold are **decremented** at the inventory service.
- On a **void** (*stornó*), a reversing call puts that stock **back**.

Everything else — the daily reconciliation, the low-stock warnings, the retry behaviour when the server is down — builds on those two movements.

7.2 Configure

Inventory sync is set up on *Beállítások* (Settings) → *Kapcsolatok* (Connections) → *Leltár szinkronizáció* (Inventory sync):

Field	Key	What it does
Enable inventory sync	`inventoryEnabled`	Master switch for stock decrement and reversal
Server URL	`inventoryApiUrl`	The inventory service endpoint, e.g. `http://localhost:3002`
API key	`inventoryApiKey`	Optional `X-Api-Key` header sent with each call
Low-stock alert	`inventoryLowStockAlert`	Surface a warning when a product hits a critical level

The `inventoryApiKey` is **optional** — include it only if your inventory service requires authentication. Like other credentials in this guide, it is treated as sensitive and is not synced between terminals, so enter it on each till that needs it.

7.3 How it works

The two basic movements — decrement on sale, reverse on void — sit inside a larger flow that keeps the numbers honest even when the network does not cooperate.

On a sale. The items on the closed order are decremented at the inventory service.

On a void or *stornó*. A **storno** call reverses the stock for those items. This includes a **PMS reversal**: when a room charge is voided in the hotel PMS and the PMS calls back to void the POS payment, the inventory is reversed at the same time, so a reversed charge never leaves phantom stock missing.

At day close. A **daily package** is sent to the inventory service, reconciling the day's movements in one batch.

When the server is unreachable. The package is not lost — it is **queued and retried** on a back-off schedule:

Stage	Behaviour
First attempts	Fast retry, roughly every **30 seconds**

Stage	Behaviour
Then	Slower retry, roughly every **30 minutes**
After **48 hours**	Flagged for **manual retry**

Because the schedule starts aggressive and only then backs off, a brief network blip clears itself within a minute, while a genuinely-down server escalates visibly rather than failing in silence.

Low-stock warnings. When `inventoryLowStockAlert` is on, a **low-stock** warning can surface on the **KDS** and in the **Low-stock panel**, so the kitchen sees a product running out before it sells the last one.

> **NOTE**
> The reversing **storno** call is what keeps voids honest. A void that only cancelled the POS payment — without telling inventory — would slowly drift the stock count low. The reversal makes the inventory mirror what was actually served, voids included.

7.4 Recipes

Many products are not sold as whole stock items — a cocktail draws on several ingredients, a plate on several components. **Recipes** map a POS product to its **ingredients**, and they are defined in **product management** (not on the Settings tabs).

Once a product has a recipe, selling it automatically draws down the **right ingredients** at the inventory service rather than a single notional unit of the finished item. Sell a gin and tonic and the gin, the tonic and the garnish each decrement by the recipe amount.

> **TIP**
> Set recipes for anything assembled from parts — cocktails, mixed drinks, composed plates. A product **without** a recipe decrements only itself, which is correct for a bottled beer but wrong for a drink poured from several bottles.

7.5 Common errors

Inventory problems are rarely dramatic; they tend to be a queue quietly growing. The two to watch for:

Symptom	Cause	What happens
Queue backing up	Inventory server slow or down	You are **warned well before** any data could be lost
Items stay queued and never clear	Wrong `inventoryApiUrl` or wrong `inventoryApiKey`	Calls keep failing; nothing reaches the server

The reassuring part is the first row: a backing-up queue is surfaced **in good time**, well ahead of the 48-hour manual-retry threshold, so you have a wide window to fix the connection before anything needs hand-holding.

> **WARNING**
> A wrong `inventoryApiUrl` or a wrong `inventoryApiKey` does not raise an immediate hard error — the packages simply **queue** and retry against a destination that will never accept them. If you see the queue growing right after enabling inventory, check the URL and key **first**, before assuming the inventory server itself is at fault.

8 Accounting & daily-close e-mail

When a trading day ends, its numbers need to go two places: to the books, and to whoever signs off on the day. Hydra POS handles both at **day close** — it can forward the closing data to an **accounting service** (Hydra ACT) and e-mail a full **Z-report** with a PDF attachment to the owner or accountant. Both are optional, both fire automatically when the day closes, and this chapter sets up each one.

8.1 Accounting (Hydra ACT)

The accounting link forwards each **day-close** to an external accounting service so the books are fed straight from the till, with no re-keying.

Configure it on *Beállítások* (Settings) → *Kapcsolatok* (Connections) → *Hydra ACT*:

Field	Key	What it does
ACT URL	`actUrl`	The accounting endpoint that receives day-close data. Empty = disabled

There is a single value to set: `actUrl`. Leave it **empty** to disable the integration entirely. Once a URL is set, the POS forwards each day-close to the accounting service **automatically** — there is nothing to trigger by hand and no schedule to manage.

> **NOTE**
> Accounting forwarding rides the **day close**, not each sale. A figure only reaches the accounting service when the day is closed, so a day still open has not yet been forwarded — by design.

8.2 Daily-close e-mail report

The e-mail report sends the day-close **Z-report** — an HTML summary plus a PDF attachment — to the owner, the accountant, or anyone else who needs the day's numbers. It is configured on *Beállítások* → *Email riport* (Email report).

Setting it up

Setting up the report is mostly a matter of describing your mail account. Work through it in order:

1. **Enable** the report (`emailEnabled`).
2. Pick a **provider preset** — *Gmail* / *Outlook* / *Yahoo* / *Egyéni* (custom). The preset fills in the host, port and SSL for you.
3. Confirm or enter the **SMTP host / port / SSL** (the preset pre-fills these; *Egyéni* lets you type your own).
4. Enter the **login** (your e-mail address) and an **app password**.
5. Set a **from name** — the display name on the "From" line.
6. List the **recipients**, separated by `;` or `,`.
7. Choose how it sends: toggle *Automata küldés napzáráskor* (auto-send at day close) on for automatic delivery, or leave it off to send manually.

The connection fields gather as follows:

Field	Key	Notes
Provider preset	`emailProvider`	Gmail / Outlook / Yahoo / *Egyéni* (custom)
SMTP host	`emailSmtpHost`	Pre-filled by the preset
SMTP port	`emailSmtpPort`	`587` (STARTTLS) or `465` (SSL)
SSL	`emailSmtpSecure`	On for port `465`, off for `587`
Login	`emailUser`	The SMTP username (your e-mail address)
App password	`emailPass`	A **2-factor app password** — see the warning below
From name	`emailFromName`	The "From" display name
Recipients	`emailRecipients`	Separated by `;` or `,`
Auto-send at day close	`emailAutoSendOnDayClose`	*Automata küldés napzáráskor*

What the e-mail contains

The report is a full picture of the day, not just a total. Each e-mail carries:

- **gross revenue**, **order count**, **covers**, **average check** and **tips**;
- a **payment-method breakdown**;
- **waiters by revenue**;
- the **top 10 items**; and
- a **VAT summary**.

A matching **PDF** is attached, so the same figures can be filed or forwarded without the HTML body.

Testing

Two actions let you prove the setup without waiting for day close:

- *Teszt email küldése* (Send test e-mail) — validates the **SMTP** connection without building a full report. Use this to confirm the host, port and app password are right.
- *Mai napi riport küldése most* (Send today's report now) — sends **today's** report on demand, full build and all.

> **TIP**
> Run *Teszt email küldése* the moment you finish entering the credentials. It checks the SMTP login in isolation, so if it fails you know the problem is the host, port or app password — not the report itself.

Credentials and security

The e-mail account's credentials need a little care, mostly because the big providers no longer accept a plain account password over SMTP.

> **WARNING**
> For **Gmail**, **Outlook** and **Yahoo** you must use a **2-factor app password**, not the normal account password. Generate it in the provider's security settings with two-factor authentication enabled, and paste **that** into the *App password* field. The normal account password will be rejected.

Two ports cover almost every provider, and they pair with the SSL toggle in a fixed way:

Port	SSL toggle	Encryption
`587`	**off**	STARTTLS
`465`	**on**	SSL

WARNING

The e-mail password is stored **encrypted** on this machine and is **not synced** to other terminals. Because it never travels in a backup or a sync message, you must enter it on **each** terminal that sends the report — otherwise that terminal's day close will look configured but fail to send.

9 Companion apps — Manager & Waiter

Two companion apps extend the till beyond the counter. The **Manager app** is an owner's dashboard that watches a venue's live trading from anywhere, over the **Hydra Admin cloud**. The **Waiter app** lets staff take orders table-side from a phone or tablet, over the **local network** — no internet needed once paired. Each is paired with a short code, and both codes live in one place on the till.

> **IMPORTANT**
> The pairing codes only appear after a successful **Hydra Admin check-in**. Set up Hydra Admin first — see the *Hydra Admin* chapter — or the code blocks in System Info will be blank.

9.1 Where the codes come from: System Info

Everything you need to pair either app is on one screen of the POS.

Open it: tap the **Hydra POS** logo in the top bar. The *Rendszerinfó* (System Info) window opens. It shows the venue/property id, the current connection status, and the two pairing codes — each with a **copy** button.

Block	Field	Used by
Manager access	property id + *Hozzáférési kód* (access code, e.g. `WHSP-VQYL`)	The owner, in the **Manager** app
Waiter pairing	property id + *Párosítási kód* (pairing code, e.g. `T2U9-E7TE`)	A waiter, in the **Waiter** launcher

The two codes are **different** and not interchangeable: an access code cannot pair a waiter, and a pairing code cannot add a venue to the Manager app. Both are derived **server-side** from the venue's `hotel_id`, so they are **stable** — a terminal swap does not change them.

> **NOTE**
> If a code block is blank, the till has not completed its first Hydra Admin check-in yet. Connect it to the internet, wait for the next check-in, and reopen System Info.

9.2 The Manager app — owner dashboard

The Manager app is the owner's window onto the venue. It runs over the **Hydra Admin cloud**, so the owner can watch **live revenue, orders and stornós** (voids) from anywhere — not just on the venue's LAN. One phone can hold several venues at once.

Connect, step by step

1. Open *Rendszerinfó* (System Info) on the till and **copy** the *Hozzáférési kód* (access code, e.g. `WHSP-VQYL`).
2. Note the **property id** shown in the same window.
3. In the Manager app, choose to **add a venue**.
4. Paste the **property id** and the **access code**, then confirm.

On success the venue appears in the app, streaming the owner's live figures.

"Live" without wasted traffic

The till does **not** stream stats around the clock. Live data flows only while a manager is **actually watching** the dashboard — a behaviour called **demand-gating**. Open the app and figures come alive within about a minute; close it and the till quietly stops pushing. The result is genuinely live data that costs almost nothing when nobody is looking.

Push notifications are optional: a manager can opt in to receive alerts (for example a large void, or the day close being ready) without keeping the dashboard open.

> **TIP**
> A blank dashboard is usually **not** a fault. Because of demand-gating, figures only stream while someone is watching — give it a minute after opening the app. If it stays empty, check the licence and that Hydra Admin has checked in (see the *Hydra Admin* chapter).

9.3 The Waiter app — table-side ordering

The Waiter app turns a phone or tablet into a roving order pad. Ordering happens entirely over the **local network** with a chosen till — **no internet is needed once paired**. Each waiter also needs a **PIN** to log in.

Before you start

Field	Set to
Pairing code	The *Párosítási kód* (e.g. `T2U9-E7TE`) from System Info
Property id	The property id shown in System Info
Waiter PIN	A PIN created in *Beállítások* (Settings) → *Pincérek & Menedzserek* (Waiters & Managers)
Network	The tablet on the **same LAN** as the till; port `3005` open between them

Connect, step by step

1. Open *Rendszerinfó* (System Info) on the till and **copy** the *Párosítási kód* (pairing code, e.g. `T2U9-E7TE`).
2. In the **Waiter launcher** on the phone, enter the pairing code.
3. The phone **discovers** the venue's terminals; pick the one you want from the list.
4. The launcher hands off to that terminal over the **LAN on port** `3005`.
5. Log in with the **waiter PIN**, then start taking orders.

> **NOTE**
> Discovery uses the cloud once, only to learn which terminals exist. The orders themselves travel **directly** to the chosen till over the LAN, so the kitchen keeps printing even if the internet drops after pairing.

9.4 Networking

Keep the tablets, phones and tills on the **same LAN** and make sure these ports are not blocked between devices:

Port	Between	Used for
`3005`	Tablet ↔ terminal	Waiter app + ordering
`3003`	Terminal ↔ terminal	Multi-terminal sync (see the *Multi-terminal sync* chapter)

9.5 Common problems

Symptom	Likely cause	Fix
Both codes blank in System Info	Hydra Admin has not checked in yet	Connect the till to the internet; wait for check-in (see the *Hydra Admin* chapter)
Waiter can't find a terminal	Phone and till on a different LAN/subnet, or `3005` blocked	Put them on the same network; open port `3005` between them
Waiter app pairs but won't log in	No waiter PIN, or wrong PIN	Create a PIN in *Pincérek & Menedzserek*; re-enter it
Manager dashboard empty	No manager currently watching (demand-gating), or a licence / check-in issue	Wait a minute with the app open; otherwise check the licence and Hydra Admin status

> **WARNING**
> The Manager and Waiter codes identify your venue. Treat them like any other credential — share them only with the owner and the waiting staff who need them, and re-issue if a device is lost.

10 Multi-terminal sync

A venue with more than one till can run them as a single trading floor: every terminal shares the same **orders, tables and KDS queue** in real time. You do that by joining the tills into one **sync group** over the LAN. One terminal acts as the **server** (*Szerver*) and the rest as **clients** (*Kliens*); a single till left on its own is **standalone** and needs none of this.

The terminals in a group share a *Csoport ID* (Group ID) and a *Sync titok* (sync secret), and talk over **WebSocket port** `3003`. Each terminal keeps its own *Terminál szám* (terminal number) — `01`, `02`, … — so receipt numbers never collide, and **only one terminal performs the day close** for the whole group.

> **IMPORTANT**
> Bring the **server up first**. A client cannot join, pull state or trade in the group until its server is configured and listening on port `3003`.

10.1 Pick the roles

Role	*Üzemmód* (Mode)	Which till
Server	*Szerver*	Your main till — it relays messages and holds the group state
Client	*Kliens*	Every other till — it connects to the server and mirrors the group
Standalone	*Önálló*	A lone till that does not sync

All of this lives in *Beállítások* (Settings) → *Terminál szinkron* (Terminal sync) on each machine.

10.2 Configure the server

On your main till, open *Beállítások* → *Terminál szinkron* and set:

Field	Set to
Üzemmód (Mode)	**Szerver**
Csoport ID (Group ID)	A name shared by **all** tills, e.g. `ETTEREM`
Sync titok (sync secret)	A shared secret — generate one with `openssl rand -hex 16`
Terminál szám (terminal number)	`01`

Tap **Mentés & újraindítás** (Save & restart). The server then listens on port `3003` and waits for clients to join.

10.3 Configure each client

On every other till, open the **same** panel and set:

Field	Set to
Üzemmód (Mode)	**Kliens**
Csoport ID (Group ID)	The **same** name as the server, e.g. `ETTEREM`
Sync titok (sync secret)	The **same** secret as the server
Server IP	The server till's LAN address, e.g. `192.168.1.100`

Field	Set to
Terminál szám (terminal number)	A **unique** number — `02`, `03`, …

Tap **Mentés & újraindítás**. The client connects to the server on port `3003`, joins the group, and pulls the current state.

> **WARNING**
> Give every terminal a **unique** *Terminál szám*. Two tills sharing a number will mint colliding receipt numbers. The server is `01`; number the clients `02`, `03`, and so on.

10.4 The sync secret

The *Sync titok* is a **per-device secret** — set the **same value** on every terminal in the group. It is what proves a joining device belongs to your venue.

> **WARNING**
> Without a sync secret, **any** device on the LAN can join the group and push order and state changes. Always set a secret, and set the identical value on every till. Like other per-device secrets, it is never carried in a backup or a sync message — type it in on each machine.

10.5 Reconnection and catch-up

Brief network drops are handled automatically and do not lose data.

- Every message the server relays carries a **sequence number**, and the server keeps a short **replay buffer** of recent messages.
- When a client briefly disconnects and comes back, it tells the server the last sequence number it applied, and the server **replays exactly the messages it missed** — so it catches up rather than falling behind.
- A reconnecting client also pulls a **full state snapshot** (a `REQUEST_STATE`), so even after a longer outage it ends up fully in step with the group.

> **NOTE**
> Because catch-up is built in, a client that loses Wi-Fi for a moment will resync on its own when the connection returns. No manual refresh needed.

10.6 The day close

A sync group closes the day **once**, on **one terminal only**. Running the day close on a second terminal would double-count the group's takings. Decide which till owns the close (the server is the natural choice) and always run it there.

10.7 Common problems

Symptom	Likely cause	Fix
A client won't join	Wrong *Csoport ID* or *Sync titok*, or `3003` blocked on the server's firewall	Match the Group ID and secret to the server; open port `3003` inbound on the server

Symptom	Likely cause	Fix
Receipt numbers collide	Two terminals share a *Terminál szám*	Give each till a unique terminal number (`01` , `02` , …)
The day's totals look doubled	Day close was run on more than one terminal	Run the day close on **one** terminal only
A client keeps dropping out	LAN instability, or the server restarted	The client reconnects and catches up automatically; confirm the server is up and on `3003`

11 Administrator appendix

This appendix gathers the **network and security facts** an administrator needs in one place: every port the till uses and what crosses it, the firewall rules that keep those ports on the LAN where they belong, and how Hydra POS stores the handful of secrets that make the integrations work. The other chapters tell you *how* to set each connection up; this one is the reference you keep open while you do.

11.1 Ports

Every port below runs on the relevant **terminal** — the till itself — and all of it is meant to stay on the venue's **LAN**. Nothing here should ever face the public internet except the single outbound `443` to the cloud.

Port	Direction	Service
`3005`	Tablet ↔ terminal	Waiter app + ordering API
`3003`	Terminal ↔ terminal	Multi-terminal sync (WebSocket)
`3001`	PMS → terminal	Inbound storno/void callbacks (POS status API)
`3000` *(default)*	Terminal → PMS	Room/service charge posting
`3002` *(example)*	Terminal → inventory	Stock decrement / storno
`9100`	Terminal → printer	ESC/POS network printing
`443`	Terminal → cloud	Hydra Admin (HTTPS) & NTAK (HTTPS + mTLS)

A quick way to read the table: the **first half** (`3005` , `3003` , `3001` , `3000` , `3002` , `9100`) is all **LAN traffic** between devices in the venue; the **last row** (`443`) is the only connection that leaves the building, and it is always **outbound** from the terminal.

> **NOTE**
> `3000` and `3002` are defaults/examples. The PMS port is whatever the hotel's PMS exposes (set as `pmsPort`); the inventory service URL carries its own port. Confirm both against the system you are connecting to rather than assuming the numbers above.

11.2 Firewall

The rules are simple to state and worth getting exactly right, because a blocked port shows up later as a connection that "just doesn't work" with no obvious cause.

Rule	Why
Tablets and tills on the **same LAN**	The Waiter app and sync both depend on devices being able to reach each other directly. A guest VLAN or a second Wi-Fi network breaks them.
`3005` and `3003` open **between devices**	`3005` carries the Waiter app and ordering API; `3003` carries multi-terminal sync. Both must be reachable across the LAN.
The PMS must reach the POS on `3001`	Storno/void callbacks arrive here. If the PMS cannot open `3001` to the till, reversals never land.
The POS must reach the PMS host on `3000`	Room and service charges post outbound here.

> **WARNING**
>
> The LAN ports (`3005` , `3003` , `3001` , `3000` , `3002` , `9100`) must **never** be exposed to the public internet. They authenticate against other devices on a trusted network, not against the open web. Forwarding any of them through a router or putting a till on a public IP is a security hole — keep them inside the LAN.
>
> **Tip.** When a connection fails for no visible reason, the firewall is the first thing to rule out. From a second device on the same network, try to reach the port directly (for example with a `ping` to the host and a connection attempt to the port). A clean failure there means the firewall, not the app.

11.3 Secret storage

Hydra POS holds only a small set of secrets, and it treats every one of them the same way: **encrypted at rest** and **per-device**. They are never written into a backup or a sync message in cleartext, and they never travel from one terminal to another. That is deliberate — see the *Prerequisites* chapter's two golden rules — but it means each secret has to be entered **on every terminal that needs it**.

Where the encryption comes from depends on the operating system:

OS	Keychain used
macOS	OS keychain
Windows	**DPAPI**
Linux	**libsecret / kwallet**

Only when **no keyring is available** does the POS fall back to storing a value in plaintext. On a properly provisioned terminal that fallback should never be reached.

The secrets, and where each one is set, are:

Secret	Where set
`licenseKey`	*Kapcsolatok* (Connections) → *Hydra Admin & Licensz* (Hydra Admin & Licence)
NTAK certificate password	*NTAK* → *mTLS tanúsítvány* (mTLS certificate)
`emailPass` (SMTP)	*Email riport* (E-mail report)
`pmsApiKey` , `posInboundApiKey`	*Kapcsolatok* (Connections) → *Hydra PMS kapcsolat* (Hydra PMS connection)
sync secret (*Sync titok* — Sync secret)	*Terminál szinkron* (Terminal sync)

> **IMPORTANT**
>
> Because these values are per-device, a freshly paired client terminal will look fully configured but **fail silently** on anything that needs a secret it does not yet have — NTAK reporting, the day-close e-mail, the PMS link. After adding or re-imaging any terminal, walk this table and set each secret that terminal needs.

11.4 The inbound status server and binding

The POS status/callback server on `3001` — the endpoint the PMS uses to query the day's state and to notify a storno — does **not** listen on every interface. It binds to the till's **advertised LAN IP**, the same address the POS publishes to the PMS as its callback origin, so the endpoint is not exposed on unrelated networks the terminal might also be attached to.

For a PMS that runs on the **same host** as the POS, set the optional `bindAddress` override (for example `127.0.0.1`) to isolate the server to **loopback** only. Nothing off-box can then reach it.

> **NOTE**
> If the till takes its address from DHCP, give it a **reservation** so the advertised IP — and therefore the callback address the PMS was told to use — stays stable across reboots. A changed IP silently breaks inbound callbacks until the PMS is updated.

12 Troubleshooting & checklists

When a bring-up goes wrong, it is almost always because something was done **out of order**, a **port** is blocked, or a **per-device secret** was never set on the terminal in front of you. This chapter gives you the order to work in, a short checklist for each integration so you can confirm it is genuinely connected, and the general checks that resolve the largest share of problems.

12.1 Recommended bring-up order

Do these in sequence. Each step depends on the ones before it — most notably, the Manager and Waiter codes do not exist until Hydra Admin has checked in, and a client terminal has nothing to sync to until the server is configured.

1. **Outlet identity + licence (Hydra Admin)** — set the hotel identity and licence key, then confirm a **green check-in**. Everything else hangs off this.
2. **Multi-terminal** — configure the **server** terminal first, then the clients. A client cannot join a group that has no server.
3. **NTAK** — go **sandbox → validate → production**. Never point a fresh setup straight at production.
4. **PMS** — set the **inbound key** (`posInboundApiKey`) on the POS, give the **outbound key** (`pmsApiKey`) to the PMS, **test a charge**, then **test a storno**. Both legs working end to end means both keys and both ports are right.
5. **Printers & KDS** — add printer profiles and kitchen stations, and print a test slip and a test ticket.
6. **Inventory, Accounting, E-mail** — enable each service and run its test send where one exists.
7. **Manager & Waiter apps** — pair them last. Their **codes appear only after the Admin check-in** in step 1 has succeeded.

> **IMPORTANT**
> If a later step misbehaves, suspect an earlier one. A red NTAK dot, a Manager app that "won't pair", or a client that won't sync are very often a missing Admin check-in or an unconfigured server terminal — not a fault in the integration you are looking at.

12.2 Per-integration checklists

Each table lists the **2-4 things that must be true** for that integration to work. If all the rows are satisfied and it still fails, capture a **Bug report** (see *General checks*) and move on — the fault is rarely where it appears to be.

Hydra Admin

Must be true	How to confirm
`hotelId` and `licenseKey` set correctly	*Kapcsolatok* (Connections) → *Hydra Admin & Licensz*
Terminal is **online** and `443` outbound is open	A browser on the same machine can load any site
A **green check-in** has happened	*Rendszerinfó* (System Info) → Admin status, last check-in

PMS

Must be true	How to confirm
`pmsApiKey` set on the POS; same key configured in the PMS	*Kapcsolatok* → *Hydra PMS kapcsolat*

Must be true	How to confirm
`posInboundApiKey` set on the POS; same key given to the PMS	Same panel — a wrong/empty key returns `401` on callbacks
Ports `3000` (POS → PMS) and `3001` (PMS → POS) open on the LAN	See the *Administrator appendix* ports table
Test charge posts and a **test storno** clears it	Post to a known room, reverse it in the PMS, watch *"PMS sztornó"* on the till

NTAK

Must be true	How to confirm
`.p12` / `.key` certificate loaded with the correct **password**	*NTAK* → *mTLS tanúsítvány* (mTLS certificate)
Validated in **sandbox** before switching to production	*Validáció* (Validation) suite passes
PC **clock** roughly correct (NTAK rejects future timestamps)	Sync the clock via NTP
Status dot **green** and the queue draining	Top-bar status dot; *Rendszerinfó* → NTAK queue

Printers & KDS

Must be true	How to confirm
Local printer name set, or an **IP:port** on the profile	*Nyomtatás & KDS* (Print & KDS) / printer profile
Network printers reachable on `9100` over the same LAN	Ping the printer IP from another device
Each KDS station assigned to the right terminal and category	*KDS állomások* (KDS stations)
A **test slip** and a **test ticket** print	*Bizonylat & Fizetés* → *Slip előnézet* → *Nyomtat*

Inventory

Must be true	How to confirm
`inventoryEnabled` on, `inventoryApiUrl` set (and `inventoryApiKey` if required)	*Kapcsolatok* → *Leltár szinkronizáció* (Inventory sync)
The terminal can reach the inventory service URL/port	A sale decrements stock; a void reverses it
The daily package sends (or queues and drains) at day close	*Rendszerinfó* → inventory queue size

Accounting & E-mail

Must be true	How to confirm
Accounting: `actUrl` set (leave empty to disable)	*Kapcsolatok* → *Hydra ACT*
E-mail: SMTP host/port/SSL, login and **app password** (`emailPass`) set	*Email riport* (E-mail report)
Test e-mail validates the SMTP settings	*Teszt email küldése* (Send test e-mail)

> **TIP**
> For Gmail / Outlook / Yahoo the e-mail **app password** is a 2-factor app password, not the normal account password. Port `587` is STARTTLS (SSL off); port `465` is SSL on.

Companion apps (Manager & Waiter)

Must be true	How to confirm
Hydra Admin has **checked in** (codes appear only afterwards)	*Rendszerinfó* → Admin status is healthy
Manager: venue id + **access code** copied correctly	*Rendszerinfó* → Manager app access (use the *access* code)
Waiter: venue id + **pairing code** copied correctly	*Rendszerinfó* → Waiter app pairing (use the *pairing* code)
Waiter tablet on the **same Wi-Fi** as the POS; `3005` reachable	Same LAN, not a guest network

Sync

Must be true	How to confirm
Server terminal configured first, with a known LAN IP	*Terminál szinkron* (Terminal sync) → role *Szerver*
Clients use the server's IP, same **Group ID** and **sync secret**	Each client's *Terminál szinkron* panel
Port `3003` open between all terminals on the same LAN	See the *Administrator appendix* ports table
Every terminal has a **unique 2-digit number** (`01` , `02` , …)	A conflict modal warns if two share a number

12.3 General checks

Most failures across every integration come down to the same handful of things. Run through these before digging into any one connection:

- **Same LAN / subnet.** Tablets and tills must share one network, not a guest VLAN or a second Wi-Fi. This single mistake breaks the Waiter app and sync at once.
- **Firewall ports open.** Confirm the LAN ports are reachable between devices — see the **ports table in the *Administrator appendix***.
- **PC clock roughly correct.** NTAK **rejects future timestamps**, so a clock skewed ahead fails reporting even when everything else is right. Keep terminals on NTP.
- **Every per-device secret set on every terminal.** A freshly paired client looks configured but fails silently on anything needing a secret it never received. Walk the secrets table in the *Administrator appendix* after adding a terminal.

> **IMPORTANT**
>
> Make **Rendszerinfó** (System Info) your **first diagnostic**. It shows every service's status, the last check-in, the backup state, NTAK queue health, recent print failures, and the Manager / Waiter codes — in one screen. Almost any "is X connected?" question is answered there before you change a single setting.
>
> **Tip.** When you do need support, send a **Hibajelentés** (Bug report) — the bug button captures a snapshot of the system state plus the recent log, so the support team sees exactly what you saw without a round of back-and-forth.

13 Glossary

The terms below appear throughout this guide. Each definition is deliberately short — one line of *what it is and why it matters* — so the glossary stays a quick reference rather than a second manual. Where a term is a Hungarian on-screen label, the English gloss follows in parentheses.

Term	Definition
PMS	**Property Management System** — the hotel's guest/folio system that Hydra POS charges room and service sales to.
NTAK	The Hungarian tax authority's reporting system that the POS reports turnover to over HTTPS with **mTLS**.
KDS	**Kitchen Display System** — the kitchen screen (or printer) that receives order tickets routed by product category.
ESC/POS	The raw thermal-printer command language Hydra POS speaks over TCP (port `9100`) to network receipt and ticket printers.
Hydra Admin	The cloud service that verifies the licence, stores off-site backups, and feeds the live Manager dashboard.
`hotel_id`	The venue's cloud identifier (`hotelId`) that binds this till to one record in Hydra Admin — its licence, backups and codes.
Licence key	The venue's `POS-…` secret (`licenseKey`) that Hydra Admin checks on start-up to authorise trading.
Group ID	The shared identifier that ties a server terminal and its clients into one multi-terminal **sync** group.
Sync secret	(*Sync titok*) — the shared secret a terminal must present to join a sync group; a wrong value rejects the join.
Terminal number	The unique 2-digit number (`01`, `02`, …) stamped into each till's receipt numbers so two tills never collide.
Access code	The code (with the venue id) that pairs the **Manager** app to a venue; copied from System Info.
Pairing code	The code (with the venue id) that pairs the **Waiter** app to a venue; copied from System Info — distinct from the access code.
Storno	(*Sztornó*) — a void/reversal of a sale or charge; room/service charges are reversed in the PMS, which calls the POS back.
Idempotency key	A stable identifier (e.g. `pos-room-<receipt>`) that lets the POS retry a charge or storno without double-billing.
mTLS	**Mutual TLS** — both client and server present certificates; how the POS authenticates to NTAK over `443`.
Folio	A guest's running room account in the PMS; a room charge **posts to the folio** against the POS receipt number.
Árkód	(Price code) — the mapping of each VAT rate to the PMS GL account codes so charges land on the right ledger accounts.
Demand-gating	The behaviour where the POS only streams live data (e.g. Manager stats) while someone is actually watching.
Grace period	The **7-day** window — measured from the last successful check-in — during which the till keeps trading if Hydra Admin is unreachable.

Term	Definition
PWA / companion app	A **Progressive Web App** installed to a device's home screen — how the Manager and Waiter apps run.
Server / Client / Standalone	The three **sync roles**: the server owns shared state, clients mirror it in real time, a standalone till syncs with no one.