

# Via One

## Live Transaction Console — User Manual

Version 1.0 · May 2026

Audience: internal operations, on-call engineers, partner-facing support

# Via One — Live Transaction Console

---

## User Manual

---

**Version:** 1.0

**Document date:** May 2026

**Audience:** Internal operations, on-call engineers, partner-facing support

---

## 1. What is the Live Transaction Console?

---

The **Live Transaction Console** (the "cockpit") is a single web page that shows everything moving through the Via One platform in real time. Think of it as Wireshark for business transactions, but business-friendly: no protocol jargon, no command line, no SSH.

It answers three questions in one place:

Question	Tab
<i>Who is calling <b>us</b> right now?</i> (every API hit, every login, every topup attempt)	<b>Inbound</b>
<i>Who are <b>we</b> calling right now?</i> (every TEMM, Altamira, PPN, CellPay, Pockyt, etc. call we make)	<b>Outbound (Providers)</b>
<i>Are our <b>webhook deliveries</b> to partners landing?</i>	<b>Webhooks</b>

The page auto-refreshes every 2 seconds, so you watch traffic happen as it happens. Click any row to expand and see the full request body, headers, and response — with sensitive fields automatically masked.

### When to use it

- **A partner says "my call failed"** → open the console, filter by their tenant, find their request in seconds, see exactly what they sent and what we returned.
- **A provider integration acts up** → open the Outbound tab, filter by provider name, see the response codes and latency profile.

- **You want a sanity check that traffic is flowing** → open it during a launch window, watch the heartbeat.
- **You're auditing what happened in a 30-day window** → use the time filter and the search box to scope down to the events that matter.

---

## 2. How to access it

---

### URL

Environment	URL
Production	<code>https://app.via.one/console/</code> <i>(when live — check with engineering before relying on this)</i>
Staging	<code>https://latcom-fix-staging-staging.up.railway.app/console/</code>

### Authentication

The console requires an **admin JWT token** (the same admin gate as `/api/admin/*`). Tokens are short-lived (1 hour) for security.

To get a fresh staging token, an engineer can run:

```
bash /tmp/staging-jwt-gen.sh
```

...and paste the output into the **Admin JWT** field at the top of the page, then click **Save**. The token is stored in your browser's local storage and used automatically for every subsequent call until you click **Clear** or it expires.

If you see "Auth needed" or 401 errors, your token has expired — get a fresh one.

---

## 3. Tour of the interface

### Header strip

```
Live Transaction Console  Live  Auto-refresh 2s
Inbound API hits, outbound provider calls, and webhook deliveries.
```

- **● Live indicator** — pulses green when polling is active, grey when paused, amber when authentication is needed, red when there's a backend error.
- **Auto-refresh checkbox** — toggle the 2-second polling. Turn off if you want the view to freeze (e.g., taking a screenshot or studying a specific row).

### Auth bar

```
Admin JWT: [.....] [Save] [Clear]
```

Paste, save, done. The dots are the masked token (it's a password input, not a security risk).

### Summary cards

```
Inbound (5 min) | Outbound providers (5 min) | Webhooks (5 min)
142 120 ok 22 err | 47 41 ok 6 err | 18 15 ok 3 err
```

Three live counters showing **the last 5 minutes** of activity in each stream. Updates every 2 seconds.

### Tab navigation

Three tabs along the top — **Inbound**, **Outbound (Providers)**, **Webhooks**. Click to switch. The currently-selected tab is highlighted blue with an underline.

## Filter row

[Search path, partner, provider, event...] [Any status ▼] [Last 15 min ▼] [Refresh now] 142 rows

Control	What it does
<b>Search box</b>	Free-text filter. Matches the path ( /api/dislogin ), partner code ( latcomdigital ), provider name ( TELEFONICA ), or webhook event name. Case-insensitive.
<b>Status dropdown</b>	"Any" / "OK" (2xx-equivalent) / "Errors only" (anything that failed).
<b>Time window</b>	"Last 15 min" / "Last 1 hour" / "Last 6 hours" / "Last 24 hours". The summary cards always show 5 minutes regardless.
<b>Refresh now</b>	Force-fetch even if auto-refresh is paused.
<b>Row counter</b>	Shows how many rows match your current filters.

## Table

The columns differ per tab; details below. Rows are listed newest-first. **Click any row to expand it** — an inline panel opens showing the full request and response as formatted JSON.

## 4. The three tabs in detail

### 4.1 Inbound tab — every API request hitting us

Column	Example	What it means
<b>Time</b>	21:12:47.280	Request timestamp (UTC, sub-second precision)
<b>Partner</b>	latcomdigital	Tenant code if the request was authenticated; — if anonymous (e.g., login attempts or attacks)
<b>Method</b>	POST	HTTP verb (GET, POST, etc.)
<b>Path</b>	/api/dislogin	Request path (no query string)
<b>Status</b>	200 (green) 401 (yellow) 500 (red)	HTTP response code
<b>Latency</b>	17 ms	How long the request took, end-to-end
<b>IP</b>	5.78.104.144	Best-effort client IP (handles Fastly + Railway forwarding chain)

Click a row to expand — you'll see:

- Full request headers (with `Authorization`, `x-api-key`, etc. redacted to `[REDACTED]` or first-8-chars + `...`)
- Full request body (with card numbers, CVVs, PINs, passwords masked)
- Full response summary (top-level fields, nested objects shown as `[object]` to bound size)
- Request ID and correlation ID if the client sent one

#### Useful patterns:

- "Show me only Latcom Digital's traffic" → type `latcomdigital` in search
- "Show me only failures" → status filter = "Errors only"
- "Show me only login attempts" → search `dislogin` or `login`
- "Show me brute-force attempts" → search `dislogin` + status = "Errors only"

## 4.2 Outbound (Providers) tab — every call we make to providers

Column	Example	What it means
<b>Time</b>	20:02:04	When we initiated the call
<b>Provider</b>	TELEFONICA , ALTAMIRA , PPN , CELLPAY , POCKYT , MUWE , etc.	Which provider we hit
<b>Type</b>	TAE , PAQUETE , BILLPAY , BALANCE , REVERSAL	What kind of operation
<b>SKU</b>	TFE_150_MXN , GL001B_6M2D_P1 , PQRI6M2DP	Product SKU sent (or wire SKU after translation)
<b>Destination</b>	5537686648	Phone number, account, or beneficiary identifier (last-4 visible only)
<b>Amount</b>	30.00 MXN , 1.75 USD	Transaction value
<b>Status</b>	SUCCESS (green) FAILED (red) TIMEOUT (red) PENDING (grey)	Outcome
<b>Latency</b>	3044 ms	Provider round-trip time

**Click a row** to see the full ISO8583 fields (STAN, RRN, auth code, MTI request/response, processing code), provider-specific metadata, and the full request and response payloads from the provider.

### Useful patterns:

- "Are TEMM cert calls succeeding?" → search TELEFONICA , time window = last hour
- "Why did Latcom Digital's last paquete fail?" → search the destination phone number, expand the row, look at the response code in the response\_payload
- "Provider X has been slow today" → search the provider name, look at the latency column trend

## 4.3 Webhooks tab — every callback we deliver to partners

Column	Example	What it means
Time	19:30:00	Delivery attempt timestamp
Partner	latcomdigital	Customer ID receiving the webhook
Endpoint	order_fulfillment_callbacks	Which configured webhook endpoint
Event	order.delivered , order.failed , cashapp.success	Event name
State	delivered (green) pending (info) retrying (yellow) dead (red)	Delivery state from the queue
Last code	200 , 404 , 500	HTTP code from the partner's last delivery attempt
Attempts	1 , 3 , 5	How many tries we've made

Click a row to see the full payload we sent, the partner's response body, and the retry history (per-attempt timestamps and error messages).

### Useful patterns:

- "Are partner webhooks landing?" → time window = last hour, status = OK only
- "Show me all dead-letter webhooks" → status = errors only
- "Did Pockyt's order callback ever reach us?" → search pockyt

## 5. Privacy & redaction

The console **never stores raw secrets**. Before any data hits the database, the request-logger middleware automatically redacts:

Field type	Treatment
<b>Authorization headers, API keys, tokens, cookies, passwords, secrets</b>	Replaced with [REDACTED] (first 8 chars kept for tracing only)
<b>Card numbers (PAN)</b> — any 13–19 digit run	** (e.g., **5832 )
<b>CVV / CVC / PIN fields</b>	[REDACTED]
<b>Phone numbers in known phone fields</b>	Last 4 digits visible only (e.g., ****5678 )
<b>Email addresses</b>	First char + domain (e.g., r***@gmail.com )
<b>Bodies larger than 32 KB</b>	Stored as a placeholder { _truncated: true, _original_size_bytes: N }

**What's still fully visible:** transaction amounts, currencies, status codes, partner/provider IDs, transaction IDs, error messages, paths, latency profiles, IP addresses.

**Retention:** Inbound rows expire after **30 days**. Outbound and webhook tables have their own retention managed by their respective subsystems.

## 6. Common workflows / playbooks

### "Partner says my call failed at 14:32"

1. Set time window to **Last 1 hour**.
2. Switch to **Inbound** tab.
3. Type the partner's tenant code ( `latcomdigital` , `HAZ_001` , `RISE_HOLDINGS_001` , etc.) in search.
4. Find the timestamp matching their report.
5. Click the row to expand — read the request body and the response.
6. If the inbound is fine but the partner saw an error, switch to **Outbound (Providers)** for the same time window — find the matching provider call.

### "Why did this transaction not get processed?"

1. Find the transaction's `dist_transid` (from the partner's report or your own logs).
2. **Inbound** tab → search the `dist_transid` (it appears in the request body summary).
3. Click → confirm we received the request and what we returned.

4. **Outbound (Providers)** tab → find the same time window → look for the matching provider call.
5. **Webhooks** tab → did we send the partner a callback? Did it land?

## "Show me what's happening right now during this launch test"

1. Open the console.
2. Auto-refresh ON.
3. Filter by the test partner if you only want their traffic.
4. Watch the **summary cards** — total counts and OK/error split update every 2 seconds.
5. New rows appear at the top of the table as they happen.

## "Audit the last 24 hours of provider failures"

1. Switch to **Outbound (Providers)** tab.
2. Time window = **Last 24 hours**.
3. Status = **Errors only**.
4. Sort/scan by provider — anomalies stand out fast.

---

## 7. Limitations & known gaps

---

This is **v1**, internal-only. Things it deliberately doesn't do (yet):

Limitation	Why / when it'll change
<b>No partner-shareable read-only links</b>	v2 — once internal version is proven
<b>No SSE / WebSocket push</b> (uses 2-second polling)	Polling is good enough for v1 — adding SSE is ~50 lines if needed
<b>No CSV / PDF export from the UI</b>	Use the underlying SQL queries directly for now
<b>No alerting / threshold rules</b>	Console is observability only — alerting lives in NOC + WhatsApp ops bot
<b><code>/api/tn/latcom</code> doesn't fully exercise the new <code>payments-iso8583</code> path</b>	The <code>payments_iso8583_enabled</code> customer flag only short-circuits <code>/api/v1/topup</code> . <code>/api/tn/latcom</code> always routes via Optimus router. Console captures everything regardless.

## 8. Troubleshooting

### "My JWT was accepted but no rows appear"

Either:

- **Time window too narrow** — bump to "Last 1 hour" or "Last 24 hours".
- **Filter too restrictive** — clear the search box, set status to "Any".
- **No traffic in the window** — generate some yourself by hitting any endpoint.
- **Wrong environment** — staging and production are separate; you're not seeing prod data on staging or vice versa.

### "JWT keeps getting rejected"

- Tokens are 1-hour. Generate a fresh one.
- Make sure you're using the **staging** secret on the staging URL, **prod** secret on the prod URL — they're different.
- Confirm you didn't accidentally copy a trailing space or paste a wrong character.

## "Console URL returns 404"

- The console may not be deployed to that environment yet. Production deploy is gated on go-live readiness; check with engineering.

## "Auto-refresh seems stuck"

- Click the ● **Live** indicator area. If it's grey or red, polling is paused or errored.
- Click **Refresh now** to force-fetch.
- Check the browser console (F12 → Console) for any JavaScript errors.

## "I see `eyJhbGci...` showing up as if it were a partner key"

- That's the bug from May 5 2026 — the middleware was treating admin JWTs as partner API keys. **Fixed** in commit `54779cad`. If you still see it, your environment may be running an older build.
-

## 9. Glossary

Term	Meaning
<b>F39</b>	ISO8583 response code field. <code>00</code> = approved. Others = various decline / error states (e.g., <code>05</code> "Do not honor", <code>51</code> "Insufficient funds", <code>83</code> "Respuesta desconocida")
<b>STAN</b>	System Trace Audit Number — a 6-digit ID Telefónica's gateway assigns to each transaction
<b>RRN</b>	Retrieval Reference Number — 12-digit reference for tracking
<b>MSISDN</b>	Mobile Subscriber ISDN — phone number in international format (we store 10-digit MX numbers without country code)
<b>Cert mode</b>	Telefónica's certification environment (vs production). When <code>PAYMENTS_CERT_MODE=true</code> , paquete attempts get rewritten at the wire to known-good cert SKU + amount ( <code>code=21, amount=150 MXN</code> )
<b>Tenant</b>	A logical partner organization. OPTIMUS, HAZ_GROUP, RISE FINANCE, VIAONE, VIAONE_MASTER. Each tenant has its own customers, route mappings, balances
<b>Customer / distributor</b>	An individual API consumer within a tenant. <code>latcomdigital</code> is a customer under the OPTIMUS tenant
<b>Partner</b>	Used loosely as a synonym for tenant or customer, depending on context
<b>Provider</b>	An external upstream we call to fulfill transactions: TELEFONICA, ALTAMIRA, PPN, CELLPAY, POCKYT, MUWE, TAECEL, ALTAN, etc.

## 10. Technical reference (for engineers)

Component	Location
Console UI	<code>public/console/index.html</code>
Backend API	<code>routes/consoleRoutes.js</code> (mounted at <code>/api/console/*</code> )
Inbound capture	<code>middleware/request-logger.js</code>
Database table	<code>api_requests</code> (migration <code>17792000000000_api_request_log.sql</code> )
Other data sources	<code>provider_transactions</code> , <code>webhook_deliveries</code> , <code>webhook_endpoints</code>

API endpoints (all admin-auth gated):

```
GET /api/console/summary           - 5-minute counters per tab
GET /api/console/inbound?since=&q=&limit= - inbound table query
GET /api/console/outbound?since=&q=&limit= - outbound table query
GET /api/console/webhooks?since=&q=&limit= - webhooks table query
GET /api/console/{type}/:id       - single-row detail (full JSON)
```

Skip-paths excluded from logging (to avoid noise): `/health`, `/favicon`, `/console`, `/api/console/`, `/app/`, `/operator/`, `/customer/`, `/static/`, `/assets/`, `/o/`.

---

## 11. Document history

---

Version	Date	Notes
1.0	2026-05-06	Initial release. Covers staging deployment + 2 May 5 bugfixes (self-poll noise, admin JWT-as-partner-key).

---