

Stripe EU Integration & Historical Enrichment

An overview of the public transactions API, and the project to give Stripe EU Legacy the same donor identity, project attribution, and subscription tracking PayPal already has — for anyone who needs the shape of this work without reading the code.

Prepared 17 Sep 2026 | Channel: Stripe EU Legacy | Live at stats.apopo.org | Endpoint /api/v1/transactions

01 · OVERVIEW

Where Stripe EU data stood, and where it stands now

Unlike PayPal, Stripe EU's own API has no 3-year history limit — every transaction since the account opened in 2016 was already being synced. What was missing was everything NAV already tracks that Stripe's API doesn't: which project/animal a gift funded, and (for Stripe's own direct-billing subscriptions) whether that subscription is still active and when it started. This report covers closing that gap.

34,402	2016–2026	31,020	24,805
Stripe EU transactions in the dashboard	Continuous coverage, full account history	Rows matched to a NAV product (90%)	Distinct donors identified by email

02 · DATA COVERAGE

Transactions per year

No recovery needed here — Stripe's own Balance Transactions API already returns the account's full history, charges and refunds alike, back to when the account opened.

Year	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026*
Txns	448	2,426	3,510	3,484	4,432	5,233	4,817	3,513	2,815	2,350	1,374

*2026 is partial, through Sep 17.

03 · HOW IT WAS BUILT

A hidden ID mismatch, then the same NAV enrichment PayPal has

Stripe EU was already syncing live. Getting it enriched meant fixing one structural problem first, then running the same NAV-matching approach already proven on PayPal.

1. Live sync Stripe Balance Transactions API, charges + refunds. No history limit, unlike PayPal's 3-year cap.	2. Charge ID fix Root-caused a Stripe ID mismatch (see 04), then a full historical re-sync backfilled charge/invoice/subscription ids onto all 34,402 rows.	3. NAV enrichment Stripe-EU.xlsx, case-insensitive charge-id match. Product, subscription no., donor identity. 31,020 rows matched.	4. Subscription status Direct-Stripe-subscription charges get real status + start date from Stripe's own API. 1,553 checked.
--	---	---	--

04 · THE ID MISMATCH

Two different Stripe id namespaces, silently blocking every NAV match

The dashboard had been storing each row under Stripe's own **balance-transaction id** (txn_ . . .) since day one. NAV's own export references the **charge id** instead (ch_ . . .) — a completely different Stripe object, in a different id namespace. No NAV row could ever have matched against this table, at any point, until that was fixed.



Found and fixed, not a data loss
The charge id was already available in the data Stripe returns on every sync (via the charge object's own expansion) — it just wasn't being stored. A new stripe_charge_id column plus a one-time full historical re-sync (re-fetching all 34,402 transactions, no new Stripe data, purely backfilling the missing id) closed the gap with nothing lost. The same expanded charge object also carries the invoice and, from that, the Stripe subscription id — both captured in the same pass, no extra API calls needed.

05 · MATCH QUALITY

What matched, and what genuinely couldn't

NAV's own export (40,808 rows, 34,656 distinct charge references) mixes in some entries that were never going to match a real transaction, verified rather than assumed.

Situation found in NAV	Decision	Rows
Charge id no longer exists in Stripe — confirmed live against Stripe's own API (resource_missing)	Excluded	3,246
PaymentIntent (pi_...) or payout-related (py_...) id — a different Stripe object, never a charge	Excluded	385
Plain-text manual batch/correction label, e.g. "SHOPIFY 06/2018 STRIPE-EU" — not an individual transaction	Excluded	5
Real charge id, found in the database	Matched	31,020

OK

Subscription status - real, not inferred

1,553 distinct Stripe subscriptions were found linked to a direct-billing charge (charge → invoice → subscription, all from data already captured during sync, no extra API calls). Every one was checked live against Stripe's own Subscriptions API: **229 active, 1,328 canceled**. Donorbox- and Tiltify-routed charges have no Stripe Subscription object behind them at all (those platforms bill separately), so they never get a `subscription_status` — that's a real absence, not a gap in checking.

06 · KNOWN LIMITS

What this integration still can't tell you

- **Stripe US Legacy has none of this yet** — no NAV export has been provided for that account. The same pipeline (charge-id backfill already shipped for both accounts; NAV matching + subscription status) runs the moment a `Stripe-US.xlsx` export exists.
- **project_code** stays UNDESIGNATED, same as PayPal — NAV's product codes map to `project_name` only, since they're free-text, not a stable coded identifier agreed with Odoo.
- **Donor identity from Stripe itself is sparse** — only 297 of 34,402 rows had `billing_details` on the charge (older API versions/checkout flows didn't always collect it). The other 24,508 donor emails come from NAV as a fallback, applied only where Stripe's own live data didn't already have one.
- **subscription_status only ever applies to Stripe's own direct-billing subscriptions** — a Donorbox or Tiltify charge is a genuine one-off payment from Stripe's perspective, even if the donor's underlying commitment on that platform is recurring.

Data: Stripe Balance Transactions/Subscriptions API + APOPO NAV export (`Stripe-EU.xlsx`) | Verified live against `stats.apopo.org` and the production database