PAYMENTS · DATA · AI · COMPLIANCE

PayPal Integration Delivery and Support Runbook

For merchants, marketplaces and fintechs, from requirements to live L1, L2 and L3 support: orders and captures, webhooks, idempotency, sandbox testing, live pilot, disputes, reconciliation, alerts, disaster recovery and audit.

SP Sankar · Payment Product Consultant

sp@paymentconsultant.top · paymentconsultant.top · linkedin.com/in/spsankar

Version 2026 · Always confirm current APIs and rules in PayPal's developer documentation and your PayPal agreement.

Contents

1. Purpose and how to use this runbook
2. Planning vs delivery alignment
3. PayPal basics for the team
4. Rules and agreements
5. The nine kinds of work, step by step
6. Exit criteria and sign-off: E2E and UAT
7. Proving it in production
8. Controls that run across every payment
9. Security, certificates and certification
10. Production support: L1, L2 and L3
11. Operational runbooks for common incidents
12. Disaster recovery
13. Internal and external audit
14. Measures to watch
15. Go-live checklist
16. Sources

Purpose and how to use this runbook

This runbook takes a PayPal integration, for example adding PayPal, Venmo or Pay Later to a website or app checkout, payouts to sellers, or moving to PayPal's current APIs, from idea to live production and support. It is written for product managers, delivery leads, engineers, testers, operations, finance and compliance teams at merchants, marketplaces and fintechs.

Unlike a bank payment rail, PayPal is a **third-party wallet and payment platform**. You don't join a network; you integrate with PayPal's APIs and follow PayPal's agreements and rules. Three things shape every PayPal project:

- **The flow is asynchronous.** The buyer approves in a PayPal window, your server captures the payment, and many final results (refunds, disputes, holds) arrive later as webhooks.
- **Money lands in your PayPal balance first**, then moves to your bank account, so reconciliation needs PayPal's own reports.
- **Disputes and holds** follow PayPal's rules and timelines, alongside card network chargebacks when the buyer paid with a card.

Golden rules for Agile PayPal delivery

- Work in sprints, but never skip the hard gates: sandbox E2E testing, webhook verification, UAT sign-off and a live low-value pilot.
- Never trust the browser alone: confirm every payment on your server and with signed webhooks.
- Make every call idempotent so a retry never charges twice.
- Keep API credentials in a secrets vault, never in code.

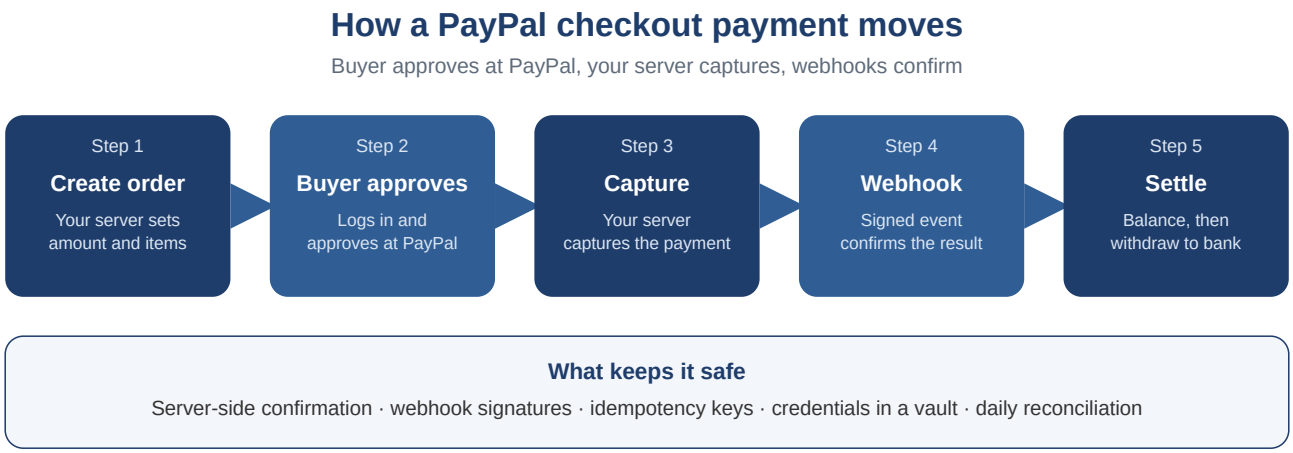
Planning vs delivery alignment

Level	What it holds	Who owns it	Reviewed
Product roadmap	Payment methods (PayPal, Venmo, Pay Later, cards), markets, PayPal API deprecation dates	Product manager	Quarterly
Release or PI plan	Features committed, dependencies, risks	Product manager, delivery lead	Every planning increment
Sprint plan	User stories for 2 weeks	Product owner and team	Every sprint
Release calendar	Deployment dates, peak-season freeze, sale events	Release manager	Weekly

- **Track outside dependencies:** PayPal account approvals and features, your e-commerce platform or payment orchestration vendor, and finance systems.
- **Avoid peak trading periods** for go-live; most merchants freeze from mid-November to early January.
- **Measure it:** say/do ratio, feature predictability, scope change and milestone status, reported weekly.

PayPal basics for the team

PayPal is one of the largest digital payment platforms. In 2025 it processed $1.79 trillion in total payment volume across 25.4 billion payment transactions, up 7% by volume (PayPal fourth quarter and full year 2025 results). Its brands include PayPal, Venmo and Braintree (for larger merchants).



How a PayPal checkout payment moves

> **How to read this picture:** Read left to right: each box is one step of a PayPal checkout payment, from your server creating the order to the money reaching your bank. The band at the bottom lists the controls that stop the two worst failures: charging a buyer twice, and taking money without recording the order.

Who is involved

Party	Role
Buyer	Pays with their PayPal or Venmo balance, linked bank account or card, or Pay Later
Merchant (you)	Shows PayPal at checkout, creates and captures orders, ships, refunds
PayPal	Authenticates the buyer, runs risk checks, moves the money, handles disputes

Party	Role
Your bank	Receives withdrawals from your PayPal balance
Platform or orchestration vendor (optional)	Connects PayPal to your store or to several payment methods

The life of a PayPal checkout payment

Step	What happens
Create order	Your server creates an order with the amount and items
Buyer approves	The buyer logs in and approves in the PayPal window or app
Capture	Your server captures the payment (at once, or later after an authorization)
Confirm	A signed webhook confirms the capture; you update the order
Settle and withdraw	Funds sit in your PayPal balance, then move to your bank
After-sale	Refunds, disputes, chargebacks and holds arrive by API and webhook

Rules and agreements

Rule	What it means for delivery
PayPal user agreement and acceptable use policy	What you may sell, how you show PayPal, and holds and reserves
PayPal developer documentation and API versions	Build to the current APIs; plan for deprecations
Card network rules and PCI DSS	Apply when you take cards; using PayPal-hosted fields keeps your PCI scope smaller
Consumer protection and UDAAP	Clear prices, fees, refund and return terms
Privacy laws	Protect buyer data you receive from PayPal
Tax and reporting rules	For payouts and marketplaces, check information reporting duties with your tax advisers

The nine kinds of work, step by step

These are not waterfall phases. In sprints they overlap; the gates are what matter.

Phase 1: Requirements

- Payment methods, markets and currencies; one-time, saved (vaulted) or recurring payments.
- Capture model: capture at once, or authorize now and capture at shipping.
- Refunds, partial refunds, disputes, payouts to sellers (for marketplaces).
- Order statuses your systems need and how finance will reconcile.

Exit: requirements signed by product, engineering, finance, operations and compliance.

Phase 2: Design

- Server-side order creation and capture; client only shows the buttons.
- Webhook handling: verify signatures, process each event once, store raw events.
- Idempotency keys on every create, capture and refund call.
- Error handling for declined funding sources, buyer cancels, timeouts and duplicate events.
- Reconciliation from PayPal reports to orders and the general ledger.

Exit: architecture and security reviews passed.

Phase 3 to 5: Development, unit and performance testing

- Build in the PayPal sandbox with sandbox buyer and seller accounts.
- Unit tests for amount calculations, currency rounding, status changes and webhook handling.
- Performance tests for sale events: checkout, capture and webhook bursts.

Phase 6: End-to-end (E2E) testing

- Every payment method and market in scope, approved and declined.
- Authorize and capture, partial capture, void, refund and partial refund.
- Disputes and holds (as far as the sandbox allows), webhook retries and duplicates.
- Reconciliation from sandbox reports to orders.

Phase 7: User acceptance testing (UAT)

- Customer service, operations and finance run real cases: refund, dispute response, reconciliation.

Phase 8 and 9: Release and deployment

- Switch to live credentials through the secrets vault; change approval; rollback plan (hide the PayPal button with a feature flag).
- Deploy, smoke test, then a live low-value pilot (see "Proving it in production"); hypercare.

Exit criteria and sign-off: E2E and UAT

E2E exit criteria

- [ ] 100% of planned test cases run in the sandbox; at least 95% passed; no open Severity 1 or 2 defects
- [ ] Every payment is captured and confirmed server side; no order is marked paid from the browser alone
- [ ] Webhook signatures verified; duplicate and out-of-order events handled
- [ ] Refunds and partial refunds correct
- [ ] Reconciliation from PayPal reports to orders balanced

UAT exit criteria

- [ ] Business scenarios signed by the business owner
- [ ] Service, dispute and finance procedures tested by the people who use them
- [ ] Customer messages and receipts checked

Sign-off matrix

Who signs	What they confirm
Product owner	Features meet acceptance criteria
QA lead	Exit criteria met, evidence stored
Finance	Reconciliation and accounting ready
Operations and service	Refund and dispute procedures ready
Information security	Credentials, webhook and data protection checked
Change manager	Release approved

Proving it in production

1. With live credentials, buy a low-value item with a real PayPal account (and Venmo and Pay Later if in scope).
2. Confirm the capture webhook and the order status.

3. Make a full and a partial refund.
4. Check the funds in the PayPal balance and the withdrawal to your bank.
5. Reconcile the pilot transactions in your finance system.
6. Release to a small share of customers first with a feature flag, then widen.

Exit: pilot payments, refunds and withdrawals reconciled; sign-off by product, finance and operations.

Controls that run across every payment

Control	What good looks like
Server-side confirmation	Orders paid only after a confirmed capture
Webhook security	Signature verified; events processed once
Idempotency	Retries never charge or refund twice
Credential management	API keys in a vault, rotated, least privilege
Fraud and risk	PayPal's risk checks plus your own rules for high-risk orders
Dispute handling	Responses with evidence within PayPal's deadlines
Reconciliation	Daily match of PayPal reports to orders and ledger
Audit trail	Every API call and webhook logged without sensitive data

Security, certificates and certification

- **Access:** least privilege for the PayPal business account; multi-factor authentication for every user; quarterly reviews; follow NIST SP 800-63B for your own systems.
- **API credentials:** client ID and secret stored in a vault; rotate on a schedule and when people leave.
- **Certificates:** TLS on your webhook endpoint; track expiry; alert 60, 30 and 7 days before.
- **PCI DSS:** if you also take cards, use hosted fields to keep scope small and complete the right self-assessment.
- **Vendors, licences and open source:** SDK versions, platform plug-ins and dependencies scanned on every build.

Production support: L1, L2 and L3

Level	Who	What they do	Target
L1	Customer service and payment operations	Buyer queries, refunds, first triage	Acknowledge within 15 minutes in trading hours
L2	Application support	Diagnose API errors, webhook backlogs, order mismatches	Respond within 30 minutes
L3	Engineering, platform vendor and PayPal merchant support	Code fixes, hotfixes, root cause	Engaged within 1 hour for Sev 1

Severity levels: Sev 1 when PayPal checkout fails for many buyers or orders are paid but not recorded (bridge within 15 minutes); Sev 2 for one method or market failing or webhook delays (30 minutes); Sev 3 for a single order (next business day).

Alerts: to Slack or Teams and paging for checkout success rate below normal, API error rate rising, webhook backlog or failures, captured-but-unfulfilled orders, and reconciliation breaks. Check PayPal's status page during incidents.

Escalation and hotfixes: L1 to L2 at 15 minutes; L2 to L3 and the incident manager at 30 minutes; senior leaders for any Sev 1. Hide the PayPal button with the feature flag if it is failing, rather than letting buyers fail at checkout. Hotfixes follow the emergency change process with a review within 5 business days.

Operational runbooks for common incidents

1. Buyers charged but orders not created

1. Use the PayPal transaction search or reports to list captures without orders.
2. Create the orders or refund the buyers; fix the webhook or capture logic.

2. Webhooks failing or delayed

1. Check your endpoint, certificate and signature verification.
2. Replay missed events from the PayPal dashboard or API; make sure each is processed once.

3. Spike in disputes

1. Find the cause (product, shipping, fraud); respond with evidence within the deadlines.
2. Tighten fraud rules or fix the customer issue.

4. Funds on hold or account limited

1. Read PayPal's notice; provide the requested information quickly.

2. Plan cash flow for the hold period.

Disaster recovery

Item	Example target
Recovery time objective (RTO)	Under 1 hour for checkout
Recovery point objective (RPO)	Zero for orders and payment records

- [ ] Backup region or site can create and capture orders and receive webhooks
- [ ] Webhook replay process tested
- [ ] DR test at least once a year, with evidence
- [ ] After failover, reconcile captures and orders to avoid duplicates

Internal and external audit

Internal audit checklist

- [ ] Payment policies and refund rules current and approved
- [ ] PayPal account users and roles reviewed quarterly; multi-factor on
- [ ] API credentials stored in a vault and rotated
- [ ] Server-side confirmation and webhook verification working (sample orders)
- [ ] Disputes answered within deadlines
- [ ] No direct database changes to payment status without approved tickets
- [ ] Daily reconciliation balanced
- [ ] DR test in the last 12 months

External audit checklist

- [ ] Evidence index linking each control to its evidence
- [ ] Order, refund and dispute population lists ready to sample
- [ ] Change records with approvals, test and pilot evidence
- [ ] Access lists and review sign-offs
- [ ] Incident records and root cause reviews
- [ ] PCI self-assessment if you take cards
- [ ] Open findings tracker with owners and dates

Measures to watch

Measure	Why it matters
Checkout success rate	Lost sales if it drops
Capture-to-order match rate	Paid but not recorded is the worst failure
Webhook latency and failures	Late events delay fulfilment
Refund time	Customer satisfaction
Dispute rate and win rate	Cost and account health
Reconciliation breaks	Must be zero at end of day

Go-live checklist

- [ ] Sandbox E2E and UAT exit criteria met and signed
- [ ] Live credentials in the vault; webhooks registered and verified
- [ ] Live pilot payments, refunds and withdrawals reconciled
- [ ] Feature flag ready to hide PayPal if needed
- [ ] Alerts, on-call and escalation tested
- [ ] DR path tested
- [ ] Service, dispute and finance teams trained

Sources

- PayPal: fourth quarter and full year 2025 results (February 2026)
- PayPal developer documentation (Orders, Payments and Webhooks APIs; check current versions)
- PCI Security Standards Council: PCI DSS v4.0.1