

PartnerPilot

Complete Setup Guide

This document walks you through every step required to take your affiliate platform from local development to production — including database, authentication, sales tracking, commission approval, and affiliate payouts.

Generated for the Affiliate Partner App codebase. Keep this PDF handy while configuring Supabase, Stripe, Shopify, and Vercel.

Overview

Your platform has four major layers to configure:

- Infrastructure — Supabase database, auth, and environment variables
- Sales tracking — Stripe / Shopify webhooks and referral links
- Commission workflow — approval rules and merchant dashboard
- Payouts — affiliate payment methods and Stripe Connect transfers

Estimated time: 2–4 hours for a first production deployment (excluding DNS propagation).

Phase 1 — Local development

Step 1: Install dependencies

1. Clone or open the project folder.
2. Run: `npm install`
3. Copy environment template: `cp .env.example .env.local`

Step 2: Create a Supabase project

1. Go to <https://supabase.com> and create a new project.
2. Wait for the database to finish provisioning.
3. In Project Settings ! API, copy:
 - Project URL !' NEXT_PUBLIC_SUPABASE_URL
 - anon public key !' NEXT_PUBLIC_SUPABASE_ANON_KEY
 - service_role key !' SUPABASE_SERVICE_ROLE_KEY (keep secret; server-only)

Step 3: Apply database migrations

1. Install Supabase CLI: <https://supabase.com/docs/guides/cli>
2. Log in: `supabase login`
3. Link your project: `supabase link --project-ref <your-project-ref>`
4. Push migrations: `npm run db:push` (or `./scripts/db-push.sh`)

This creates all tables: programs, affiliates, clicks, conversions, commissions, payouts, regions, MLM, SSO, and RLS policies.

Step 4: Configure app branding

1. Set `NEXT_PUBLIC_APP_URL=http://localhost:3000` for local dev.
2. Set `NEXT_PUBLIC_APP_NAME` to your program name (e.g. Podium Partner Program).
3. Start the dev server: `npm run dev`
4. Open `http://localhost:3000` and sign up for a merchant account.

Step 5: Complete merchant onboarding

1. After signup, complete `/onboarding` to create your first affiliate program.
2. Set commission type, cookie window, payout threshold, and MLM settings.
3. Visit Dashboard !' Settings to fine-tune approval rules and branding.

Phase 2 — Stripe (sales + payouts)

Stripe handles two jobs: (1) detecting paid checkouts/subscriptions via webhooks, and (2) sending money to affiliates via Connect transfers.

Step 6: Enable Stripe Connect on your platform account

1. Log into `https://dashboard.stripe.com`
2. Go to Connect !' Settings and complete platform profile.
3. Enable Express accounts for affiliates (recipient onboarding).
4. Copy your Secret key !' `STRIPE_SECRET_KEY` in `.env.local`
5. Optional: copy Publishable key !' `NEXT_PUBLIC_STRIPE_PUBLISHABLE_KEY`

Step 7: Register Stripe webhooks

1. Install Stripe CLI: `npm i -g @stripe/cli` && `stripe login`
2. For production, run:
`NEXT_PUBLIC_APP_URL=https://your-domain.com ./scripts/stripe-webhook-setup.sh`
3. Copy the webhook signing secret (`whsec_...`) !' `STRIPE_WEBHOOK_SECRET`
4. For local testing:

```
stripe listen --forward-to localhost:3000/api/webhooks/stripe
```

Webhooks listen for `checkout.session.completed` and `invoice.payment_succeeded` to create conversions and commissions.

Step 8: Connect merchant Stripe (Dashboard !' Integrations)

1. In the dashboard, click Connect Stripe.
2. Complete OAuth — stores the connected account in the integrations table.
3. Set `STRIPE_CONNECT_CLIENT_ID` from Stripe Connect settings if using OAuth.

Step 9: Pass affiliate referral codes at checkout

Stripe must receive the affiliate code on each checkout. Use one of:

- `client_reference_id` = affiliate referral code on Checkout Sessions
- `metadata.affiliate_ref` = referral code

- For subscriptions: metadata.affiliate_ref on the subscription or invoice

Referral links append ?pp_ref=<code> to landing URLs. Your checkout must read this and pass it to Stripe.

Phase 3 — Shopify (optional)

Skip this phase if you only sell via Stripe or custom sites.

Step 10: Create a Shopify app

1. In Shopify Partners, create a custom app.
2. Set redirect URL: {APP_URL}/api/integrations/shopify/callback
3. Copy API key !' SHOPIFY_API_KEY and secret !' SHOPIFY_API_SECRET
4. Set SHOPIFY_WEBHOOK_SECRET for HMAC validation.

Step 11: Connect Shopify in the dashboard

1. Dashboard !' Integrations !' Connect Shopify.
2. Install on your store — registers orders/paid webhook automatically.
3. Shopify matches affiliate codes from order note attributes, discount codes, or referrer URL.

Phase 4 — Affiliate program launch

Step 12: Configure commission rules

1. Dashboard !' Settings: set default commission %, payout threshold, cookie window.
2. Toggle “Manually approve commissions” OFF to auto-approve webhook sales.
3. Optional: Dashboard !' Regions for geo-based tiered rates.
4. Optional: Dashboard !' Ranks for MLM override percentages.

Step 13: Open affiliate applications

1. Share your public apply page: {APP_URL}/p/<program-slug>
2. Or org-wide apply: {APP_URL}/portal/apply
3. Dashboard !' Applications: approve applicants and assign referral codes.

Step 14: Affiliates set up payouts

1. Affiliates log into Portal !' Payouts.
2. Choose method: Manual, PayPal, or Stripe.
3. For Stripe: click Connect Stripe to complete Express onboarding.
4. Select preferred payout currency (FX stamped at commission time).

Step 15: Share referral links

1. Each affiliate gets a code (e.g. maya-ecosystem).
2. Link format: {APP_URL}/r/<code>?p=<program-slug>
3. Optional destination override: add &u=https://your-product-page.com

4. For satellite sites, embed `/pp.js` to capture `?pp_ref=` cookies cross-domain.

Phase 5 — Commission & payout workflow

Step 16: Sales create commissions automatically

When a tracked sale completes:

- Stripe/Shopify webhook or POST `/api/track/conversion` creates a conversion
- Commission rows are created (pending or approved based on settings)
- MLM overrides apply if enabled

Step 17: Approve commissions (if manual approval is on)

1. Dashboard !' Commissions
2. Review pending rows — Approve, Reject, or Refund individually
3. Or click “Approve all pending” for bulk approval

Step 18: Pay affiliates

1. Dashboard !' Payouts
2. Affiliates above the payout threshold appear in “Ready to pay out”
3. Click Mark as paid (manual), Record PayPal payout, or Pay via Stripe
4. Commissions move to paid status; payout history is recorded

Stripe transfers require platform balance and a connected affiliate Express account. PayPal records a processing payout — automated Mass Payouts API is not yet integrated.

Phase 6 — Production deployment

Step 19: Deploy to Vercel (recommended)

1. Push code to GitHub.
2. Import project in Vercel and set all environment variables from `.env.local`.
3. Set `NEXT_PUBLIC_APP_URL` to your production domain.
4. Deploy and verify the site loads.

Step 20: Update external webhook URLs

1. Stripe: point webhook to `https://your-domain.com/api/webhooks/stripe`
2. Shopify: reconnect or update webhook if domain changed
3. Re-run `./scripts/stripe-webhook-setup.sh` with production URL if needed

Step 21: Production smoke checks

1. Sign up / log in as merchant and affiliate in production.
2. Run: `node scripts/smoke-payout-flow.mjs` (validates env + payout math)
3. Send a test Stripe checkout with `client_reference_id` set to a test affiliate code

4. Verify conversion appears in Dashboard !' Commissions
5. Approve (if needed) and test a manual payout

Step 22: SSO for multi-brand sites (optional)

1. Generate OAUTH_JWT_SECRET: openssl rand -base64 48
2. Dashboard !' Integrations !' SSO & API: register OAuth clients per satellite site
3. Mint API keys and configure satellites per examples/satellite-integration.md
4. Satellites POST sales to /api/track/conversion with Bearer pp_live_... key

Go-live checklist

Use this quick checklist before announcing your program:

- Supabase project linked and migrations applied
- All env vars set in production (Supabase, Stripe, app URL)
- Stripe webhook registered and STRIPE_WEBHOOK_SECRET set
- Stripe Connect enabled; affiliates can onboard in portal
- Merchant completed onboarding and program settings
- Test sale creates a conversion + commission
- Commission approval flow tested
- Payout flow tested (manual or Stripe)
- Public apply page shared with partners
- Privacy / Terms pages reviewed (footer links)

Known limitations (roadmap)

These items are not required for launch but are planned or partial:

- PayPal Mass Payouts API — records payout only; no automated send
- Refund/chargeback webhooks — manual commission refunds in dashboard today
- Email notifications (sale approved, payout sent)
- Marketing asset library in affiliate portal
- FX rate auto-refresh cron (rates seeded in DB; update manually for now)
- Recurring subscription commission scheduler beyond per-invoice webhooks

Questions? Contact your developer or refer to README.md and examples/satellite-integration.md in the project repository.