

NineLogix Merchant Application and Integration Guide

Version 1.0 · August 2026

For digital-product merchants, operators, and developers applying to integrate with
NineLogix

NineLogix

Contents

How to use this guide	3
10-minute quick start	3
1. Before you begin	4
2. Submit your application	4
3. Confirm business information	4
4. Access Merchant Portal	4
5. Choose an integration method	5
6. Payment Links	5
7. Hosted Checkout	5
8. Developer preparation	5
9. Payment API quick start	6
10. Webhooks	6
11. Complete integration testing	6
12. Request production launch	7
13. Daily operations	7
14. Security	7
15. FAQ	7
Appendix A Integration checklist	8
Appendix B Status and event quick reference	8
Appendix C Before contacting support	8

How to use this guide

This guide explains how a merchant applies, confirms business information, receives a test integration package, completes testing, and requests production capabilities. It does not mean that live payments are automatically enabled.

Capability labels

Available	Usable in the current system as described
Approval required	Available only after merchant-specific administrator approval
Not available	Must not be treated as a delivered capability

10-minute quick start

1. Submit an application with a precise product, billing model, and target-market description.
2. After business review, activate the invited Merchant Portal account.
3. Get the Merchant ID, Store ID, and one-time test API key from the developer workspace.
4. Test a Payment Link, Hosted Checkout or Payment API flow and verify the signed test webhook.
5. Submit test evidence; request production API, payment methods, and live collection separately.

1. Before you begin

Confirm fit and prepare verifiable information.

- NineLogix supports digital-product commercialization and payment integration readiness. Applying does not guarantee approval or immediate collection.
- Prepare the product website, product description, delivery flow, pricing, refund approach, and support channel.
- Restricted, higher-risk, or insufficiently documented products may need enhanced review or may not be eligible.
- The merchant journey has four parts: apply, confirm business information, configure and verify, and prepare to launch.

2. Submit your application

Specific applications can be reviewed more efficiently.

- Provide the legal entity, contact, product name, website, and a working email address.
- Describe the customer problem, what is delivered, and who uses the product in plain language.
- State whether billing is one-time, subscription, usage-based, or mixed. Do not promise an unavailable capability.
- List target markets, expected currencies, typical prices, and estimated transaction volume.
- Before submitting, verify the site, policies, business facts, and contact details.

3. Confirm business information

NineLogix may request business, delivery, or policy evidence.

- A follow-up request explains what is missing and how to provide it. Reply in the original application context.
- The website should explain the product, pricing, delivery, support, and relevant policies.
- Product names, descriptions, prices, refund terms, and delivery evidence should agree.
- A case may be awaiting information, under review, ready for configuration, or not currently eligible. These are not payment or production statuses.

4. Access Merchant Portal

Approved merchants enter their own NineLogix organization through a controlled invitation.

- Activate only through the official invitation and secure the account with a strong password and protected email.
- Merchant ID identifies the organization. Store ID identifies a store. Use only approved combinations.
- A Merchant may contain multiple Stores; orders, sessions, and credentials remain tenant-isolated.
- Test and production are separate. Test credentials or successful tests never enable production automatically.
- Give each teammate the minimum access required and revoke access immediately when roles change.

5. Choose an integration method

Start with the simplest method that meets the business need.

Method	Best for	Effort	Current status
Payment Link	Manual sales and early validation	Low	Test available
Hosted Checkout	Standard website checkout	Medium	Test available
Payment API	Server-side orders and automation	Higher	Test available; production approval required

- Choose Payment Links without a development team.
- Choose Hosted Checkout for a stable website checkout entry.
- Choose Payment API for server-created orders, idempotency, and status queries.
- An embedded JavaScript SDK is not a currently delivered capability.

6. Payment Links

Create a test link on a NineLogix domain in Merchant Portal.

- Enter a clear product name, amount, USD currency, and expiry before creating the link.
- Share a test link only with controlled testers; never describe it as a live payment link.
- View link and session status and disable links that are no longer needed.
- Live Payment Links require separate merchant, payment-method, and collection approval.

7. Hosted Checkout

Customers first arrive on a NineLogix-hosted page.

- Create a Checkout Session server-side and send the customer to the returned NineLogix checkout_url.
- Checkout shows only currently available methods approved for that merchant.
- Success, cancel, and browser Return pages are experience signals and cannot set final payment state.
- Final state must come from a verified payment webhook or an explicitly enabled trusted server-side source.
- Sessions expire. Complete checkout within the displayed window and never reuse an expired session.

8. Developer preparation

Keep all credentials server-side.

- The developer workspace shows Merchant ID, Store ID, approved scopes, and environment status.
- A test API key is shown once and only its hash is retained. Store it immediately in a secret manager.

- Never place an API key or webhook secret in browser code, mobile apps, public repositories, logs, or chat.
- Use minimum scopes. Revoke and rotate immediately if exposure is suspected.
- A production key is available only after production API approval and still does not enable live collection.

9. Payment API quick start

Use server-side Bearer authentication and a stable idempotency key.

- Base URL: `https://www.ninelogix.com/api/v1`
- Authentication: Authorization: Bearer `nlx_test_...`
- Creation requests require Idempotency-Key. Reuse the same key when retrying the same business action.
- POST `/checkout-sessions` creates a test session; GET `/checkout-sessions/{session_id}` returns status.
- `checkout_url` is always a NineLogix URL. Do not parse or replace it with any upstream address.
- Common errors: 401 credential; 403 scope, environment or approval; 409 idempotency conflict; 422 invalid request; 429 rate limited.

```
curl -X POST https://www.ninelogix.com/api/v1/checkout-sessions \
  -H "Authorization: Bearer nlx_test_REPLACE" \
  -H "Idempotency-Key: order-demo-001" \
  -H "Content-Type: application/json" \
  -d '{"store_id":"store_REPLACE","amount_minor":1000,"currency":"USD","product_name":"Demo product"}'
```

10. Webhooks

Webhooks deliver server-side state changes.

- Register an HTTPS endpoint. Private networks, localhost, and unsafe addresses are rejected.
- The signing secret is shown once. Verify HMAC-SHA256 over the raw body and timestamp using constant-time comparison.
- Deduplicate by `event_id` and reject stale timestamps. Repeated delivery must not deliver a product twice.
- Send a signed test event and record reception time, `event_id`, and verification result.
- Signed test events are available. Automated production delivery, durable retry, dead-letter handling, and manual replay are not currently open; launch approval will state the available boundary.

11. Complete integration testing

Produce repeatable evidence that the integration behaves correctly.

- Create a test order with a clear product and amount and open Hosted Checkout.
- Check desktop and mobile behavior, including success, cancel, and expiry experiences.
- Receive and verify a test webhook; repeated `event_id` values must remain idempotent.
- Replay creation with the same Idempotency-Key and confirm that no second business order is created.

- Submit test order IDs, time, environment, result, and safe screenshots. Never submit credentials.

12. Request production launch

Completed testing is only a prerequisite for production approval.

- An administrator can approve or pause production API access. Production API approval does not enable live collection.
- Payment methods, currencies, markets, and one-time or subscription capabilities require merchant-specific approval.
- Create the production key only after approval and verify origin, scopes, and secret custody before launch.
- Run the first live transaction only under the approved controlled plan and observe webhook and order state afterward.
- One-time Alipay+ payment has controlled production evidence. This does not promise every wallet, country, currency, or subscription capability.

13. Daily operations

Keep orders, payments, and customer support aligned.

- Use NineLogix Order, Checkout Session, and Payment status when handling customer questions.
- For failures or timeouts, first check identifiers, creation time, session status, and customer-facing message. Do not blindly recreate.
- Refunds and disputes are not merchant self-service capabilities. Submit a support request with delivery evidence.
- Automated reconciliation, fees, and settlement are not currently open merchant features; rely on separately confirmed operational records.
- Provide NineLogix identifiers, environment, time, and safe screenshots to support. Never send credentials or full payment tokens.

14. Security

Treat credential custody, finality, and access control as launch requirements.

- Keep API keys and webhook secrets only in server-side secret management.
- Never send secrets through chat, ticket bodies, screenshots, or client-side errors.
- Do not rely on browser Return for payment success. Trust only a verified webhook or approved server-side final state.
- Review team access regularly and revoke or rotate after departures, device loss, or suspected exposure.
- Pause fulfillment and contact NineLogix when suspicious transactions, replay, or impersonation is detected.

15. FAQ

Short answers to common questions.

- Can I collect immediately after approval? No. Complete testing, then obtain separate production API, payment-method, and live-collection approvals.
- Can a test key be used in production? No. Environments and credentials are strictly separate.
- Can I fulfill when the Return page says success? No. Wait for a verified webhook or trusted final server state.
- Are subscriptions, refunds, or automated settlement available? Do not treat them as currently open capabilities.
- Can I promise every Alipay+ wallet or country? No. Coverage is limited to the merchant-specific approved capability.
- How do I get help? Provide Merchant ID, Store ID, NineLogix order/session ID, time, and environment - never a secret.

Appendix A Integration checklist

- Application facts complete
- Policies and delivery details accessible
- Test key and webhook secret stored securely
- Payment Link or Hosted Checkout test passed
- API idempotency and query passed
- Webhook signature and event_id deduplication passed
- Mobile review passed
- Production API and live payment separately approved

Appendix B Status and event quick reference

- pending / checkout_created: waiting for customer action
- paid / succeeded: complete only after a trusted final event
- failed: this payment failed; automatic retry is not necessarily safe
- expired: session expired; decide whether to create a new session
- payment_succeeded: normalized NineLogix success event, subject to the approved event contract

Appendix C Before contacting support

- Merchant ID and Store ID
- NineLogix Order ID or Checkout Session ID
- Test or production environment
- Time and timezone
- Expected and actual result
- Screenshots without credentials, full payment links, or unnecessary personal data