

KYRO API · V1

Pre-transaction counterparty decisions for wallets

Kyro is a pre-transaction counterparty decision layer powered by wallet intelligence and reputation evidence.

Before funds move, a caller checks a wallet and receives an **allow** / **caution** / **block** verdict with a recommended USDC limit, machine-readable reason codes, the evidence the verdict used (and what was missing), data-coverage notes and an optional immutable, shareable decision receipt. Supporting endpoints expose the underlying wallet score, Kyro profile summaries, the verified-relationship trust graph, batch screening and on-demand intake for wallets Kyro has not seen yet.

BASE URL

https://www.thekyro.co
Production · HTTPS only

MACHINE-READABLE SPEC

/kyro-openapi.yaml
OpenAPI 3.1.0 · 8 operations

FORMAT

JSON envelope
Versioned ok / data /
error wrapper on every
response

AUTHENTICATION

Optional bearer key
Anonymous access on
every endpoint

Authentication

Every endpoint works anonymously. Omit the `Authorization` header entirely. Sending a Kyro API key raises the rate budget:

```
Authorization: Bearer kyro_live_...
```

The header is only validated when present. A malformed, unknown or revoked key is rejected with

```
401 INVALID_KEY
```

Rate limits

CALLER	BUDGET
Anonymous	20 units / minute per IP
With API key	120 units / minute per key
Unit costs	Most requests 1 · batch of N rows N · intake 5 Receipt creation: extra cap of 5 per minute

Limits use a fixed 60-second window. Responses carry `X-RateLimit-Limit` and `X-RateLimit-Remaining`; a `429` adds `Retry-After` in seconds.

SCOPE & INTENDED USE

Kyro provides wallet intelligence and counterparty decision infrastructure. It does **not** perform AML or sanctions screening and does **not** provide legal, regulatory or compliance determinations of any kind.

Verdicts, scores and receipts are advisory signals derived from observable on-chain evidence and Kyro's reputation graph. They are inputs to the caller's own decision process, not guarantees of counterparty behavior or transaction outcomes. When evidence is missing, Kyro returns a conservative baseline and flags it instead of guessing.

Endpoint reference

GET `/api/v1/score/{wallet}`

SCORES

Get a wallet's committed reputation score

0-100 score with component breakdown, chain coverage and freshness. Valid wallets always answer 200. A wallet without a committed snapshot returns a conservative baseline payload instead of a 404.

RETURNS 200 · 400 · 401 · 429 **COST** 1 unit

GET `/api/v1/profile/{username}`

PROFILES

Get a Kyro profile summary by username

Public summary of a registered Kyro identity: display name, linked wallet, score snapshot and profile metadata. Unregistered usernames answer 404 NOT_FOUND.

RETURNS 200 · 400 · 401 · 404 · 429 **COST** 1 unit

GET `/api/v1/trust/{wallet}`

TRUST

Get a wallet's verified trust graph

Transaction-backed relationship edges with a snapshot and graph metrics computed at read time, plus anomaly flags. Sanitized for public consumption.

RETURNS 200 · 400 · 401 · 429 **COST** 1 unit

GET `/api/v1/decision/{wallet}`

DECISIONS

Get an allow / caution / block decision for a wallet

Verdict with recommended USDC limit, reason codes, evidence used and coverage notes. Optional ?useCase= payment | escrow | lending | marketplace (default payment). Valid-but-unknown wallets answer 200 with a conservative baseline.

RETURNS 200 · 400 · 401 · 404 · 429 **COST** 1 unit

POST `/api/v1/decision/batch`

DECISIONS

Screen many counterparties in one call

Up to 200 raw entries; after case-insensitive dedupe, at most **10 unique rows anonymously** or **50 with an API key**. Per-row problems become row statuses (ok · no_score · invalid · error) and never fail the batch. Committed reads only; no receipts.

RETURNS 200 · 400 · 401 · 429 **COST** N units (N = unique rows)

POST `/api/v1/decision-receipts`

RECEIPTS

Mint an immutable, shareable receipt of a decision

Runs a live decision and freezes the verdict into an immutable receipt (201). An identical decision for the same wallet, use case and UTC day dedupes to the existing receipt (200).

RETURNS 201 · 200 · 400 · 401 · 404 · 429 **COST** 1 unit + 5 creations/min cap

GET `/api/v1/decision-receipts/{id}`

RECEIPTS

Read a decision receipt by id

Fetch a frozen receipt (rcp_...) exactly as minted. Receipts never change after creation; successful reads are served with immutable cache headers.

RETURNS 200 · 400 · 401 · 404 · 429 **COST** 1 unit

POST `/api/v1/intake/{wallet}`

INTAKE

Index a wallet Kyro has not seen yet

Starts on-demand indexing. Idempotent by state: already-indexed wallets answer 200 already_indexed; new or in-flight work answers 202 started / indexing. A 10-minute per-wallet cooldown limits re-triggering.

RETURNS 200 · 202 · 400 · 401 · 429 **COST** 5 units

Response envelope

Every response, success or error, is JSON in the same versioned envelope.

SUCCESS

```
{
  "ok": true,
  "version": "v1",
  "data": {
    "wallet": "0xd8dA...6045",
    "score": 79,
    "cacheStatus": "cached"
  }
}
```

ERROR

```
{
  "ok": false,
  "version": "v1",
  "error": {
    "code": "RATE_LIMITED",
    "message": "Rate limit exceeded (20
      requests per minute). Retry in 17s."
  }
}
```

Error codes

CODE	STATUS	MEANING
INVALID_WALLET	400	Malformed wallet address
INVALID_USERNAME	400	Malformed profile username
INVALID_REQUEST	400	Bad body or size limits exceeded
INVALID_KEY	401	Malformed, unknown or revoked API key
NOT_FOUND	404	Profile or receipt does not exist
RATE_LIMITED	429	Rate budget or creation cap exhausted
SCHEMA_MISSING	5XX	Backing store unavailable
INTERNAL	5XX	Unexpected server error

Error `code` values are stable and machine-readable; `message` text is human-readable and may change. Any operation can also answer `5XX` in the same envelope.

No-score semantics

A valid wallet Kyro has not committed a score for still answers HTTP 200: the score read returns a conservative baseline payload, the decision read returns a conservative baseline verdict (freshness `indexing_required`) and batch rows report `no_score` instead of a guessed verdict. To bring an unseen wallet online:

1

Request intake

POST `/api/v1/intake/{wallet}` answers 202 and indexes the wallet in the background.

2

Poll the score

GET `/api/v1/score/{wallet}` until `cacheStatus` is "cached". The snapshot is now committed.

3

Re-run the decision

GET `/api/v1/decision/{wallet}`. The verdict now uses full evidence and can be frozen into a receipt.

DOCUMENT NOTES

This document summarizes the Kyro API v1 for review purposes. The authoritative, machine-readable contract is the OpenAPI 3.1.0 specification served at <https://www.thekyro.co/kyro-openapi.yaml>.

Kyro does not perform AML or sanctions screening and does not provide legal, regulatory or compliance determinations. All verdicts and scores are advisory signals for the caller's own decision process.