Connect Wallet



Hunch Cup

HHunch

MarketsSwipeStackAccount

Start swiping

Docs

# PlayHunch docs.

Everything for the swipe feed, the markets, the wallet, and the agent platform — in one place. Search it in depth, browse by section, or hand the whole thing to an LLM as Markdown.

Archived copy · cryp2nova.com

Building with an AI agent? Copy this entire hub as clean Markdown.

Copy page

StartMarketsFeaturesBuildResources



Search docs — try “parimu

⌘K

StartMarketsFeaturesBuildResources

Getting started

- What is Hunch?
- How a bet works
- Place your first bet

Archived copy · cryp2nova.com

- The swipe feed
- Markets explorer
- Wallet & USDC

Markets & resolution

- Anatomy of a market
- Every market type
- Binary milestones
- Range ladders
- Head-to-heads & flips
- Up/Down rounds
- Count-the-days markets
- Crossing-date windows
- Combined-cap baskets
- Manual real-world events
- How resolution works
- Payouts & fees

Platform features

- Portfolio & Stack
- Proof & receipts
- Leaderboard & streaks
- Referrals
- Social entrypoints
- Base App Mini App
- Settlement rails
- Safety & trust

Build on Hunch

- Agent platform
- MCP for agents
- CLI & SDKs
- Agent API reference
- Hunch Intelligence on x402 Cloud
- Base MCP
- Architecture
- Machine-readable docs

Resources

- Future roadmap
- Hunch Catalysts

CRYP2NOVA

Archived copy · cryp2nova.com

- Community & links

# Getting started

What Hunch is, how a bet works end to end, and where you bet from.

## What is Hunch?

Start here

Hunch turns prediction markets into a feed you can swipe. Back YES or NO on live markets — token milestones, up/down rounds, launchpad races, head-to-heads and more — with tickets small enough for fast discovery.

Every bet settles in USDC on Base with gas sponsored, so you sign once and a relayer pays the network fee. An agent rides under each market to route the best odds and keep proof attached to the trade. The same markets are bettable by humans and by autonomous agents over a keyless x402 rail.

Swipe feed

A Tinder-style deck of live markets. Swipe right for YES, left for NO.

Open the feed→

Markets explorer

Search and filter every live market by token, chain, type, or status.

Browse markets→

Portfolio (Stack)

Your open positions, realizable payouts, history, and receipts.

View your stack→

Agent platform

Discover, simulate, and place real USDC bets from code or an AI assistant.

Build an agent→

## How a bet works

A bet on Hunch is a single signed USDC payment on Base. You never need ETH for gas — a relayer pays the network fee and moves your USDC with an EIP-3009 authorized transfer, so the whole thing is one signature.

CRYP2NOVA

Archived copy · cryp2nova.com

Markets are parimutuel: your stake joins a pool, and when the market resolves the whole pool (net of a small fee) is split among the winning side. There is no order book and no house taking the other side.




From a swipe to a payout — one signature, no gas, settled on Base.

1. 1

   Pick a market — From the swipe feed, the explorer, or a shared link.

2. 2

   Sign once — Authorize the USDC transfer. Gas is sponsored by the relayer.

3. 3

   Settle on Base — The relayer broadcasts the EIP-3009 transfer; your stake joins the pool.

4. 4

   Market resolves — An automated resolver (or a human, for manual markets) decides the outcome from real data.

5. 5

   Auto-payout — Winners are paid straight to their wallet — no claim step.

Gasless by design

You hold and bet in USDC. The relayer covers ETH gas on every bet and every payout, so you never top up a gas balance.

## Place your first bet

You can browse every market signed out. To bet, you need a wallet and a little USDC on Base — both take seconds to set up.

1. 1

   Sign in — Connect a wallet or use the embedded wallet — an account is created for you on first sign-in.

2. 2

Archived copy · cryp2nova.com

Add USDC — Add funds to the same Base address you bet from. Tickets start small, so a few dollars is plenty to explore.

3. 3

   Find a market — Swipe the feed or search the explorer for a token or theme you have a view on.

4. 4

   Size and confirm — Choose YES or NO, set a ticket, and sign once. Your bet is live the moment it settles.

Small tickets on purpose

Markets are tuned for breadth, not whales — small entries keep discovery fast and the feed lively.

Open the feed→

## The swipe feed

The swipe feed is a Tinder-style deck of the markets we built. Swipe right for YES, left for NO, or open a card for the full chart, snapshot, and a sized bet.

Tickets stay small by design so you can explore breadth quickly. Closing-soonest markets lead the deck, then milestones and flips by deadline.

- Swipe right → YES · swipe left → NO · tap a card to open the full detail page.
- Inline $1 bets settle over the same audited trade route the explorer and agents use.
- The deck is ranked by how soon each market closes, so time-sensitive calls surface first.

Open the swipe feed→

## Markets explorer

The explorer lists every live market with probability-forward cards. Search instantly by token ($BNKR, $HUNCH), chain (Base, Solana), or any keyword, and filter by status and market type.

Open any card for the detail page: live odds, the total bet so far, charts, a market snapshot, and related markets for the same token and type.

- Search by cashtag, chain, or free text — results update as you type.
- Filter by status (open, locked, resolved) and by market type.
- Cards show live odds and trailing-24h volume; lists paginate with a cursor so large catalogues stay fast.

Browse all markets→

Archived copy · cryp2nova.com

### Wallet & USDC

Your balance, deposits, withdrawals, and receipts are built around USDC on Base so the money path is always legible. Add funds to the same address you bet from, and withdraw to any Base address.

Bets and payouts are relayed — you never need ETH for gas — and every settlement carries an on-chain proof link you can verify on BaseScan. If you use the embedded wallet, you can export your own key at any time.

1. 1

   Add funds — Top up USDC on Base to the address you bet from.

2. 2

   Withdraw — Send USDC to any Base address; the same relayer covers gas.

3. 3

   Export key — Own the embedded wallet outright — export the private key whenever you want.

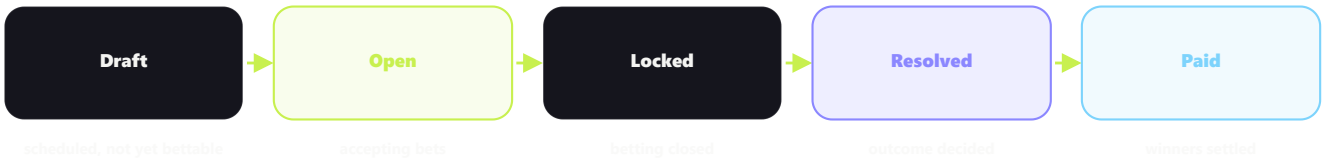
Manage your wallet→

## Markets & resolution

Every market type Hunch runs, how each one resolves, and how winners are paid.

### Anatomy of a market

Every market — whatever it predicts — shares one shape. It has a subject (a token, a chain, an event), a set of outcomes you bet on, a deadline, a parimutuel fee, and a settlement rail. It moves through the same lifecycle from draft to paid.




Every market walks the same path. Threshold markets can jump straight to Resolved the instant the outcome locks.

| Status | Meaning | Bettable? |
|---|---|---|
| Draft | Scheduled but its betting window hasn't opened (e.g. a future-anchored round). | No |
| Open | Accepting bets. | Yes |
| Locked | Betting window closed; awaiting resolution. | No |

Archived copy · cryp2nova.com

| Status | Meaning | Bettable? |
|---|---|---|
| Resolved | Outcome decided and recorded. | No |
| Paid | Winners settled on-chain. | No |

Market status

- Outcomes are binary (YES/NO, or UP/DOWN for direction markets) or N-way (a ladder of ranges, or a field of named candidates).
- A small parimutuel fee (2% by default) is the only cut; the rest of the pool flows to winners.
- Markets settle on a rail — Base USDC by default, with Sui and an Arbitrum vault as parallel rails.

## Every market type

Hunch runs more than twenty distinct market types. They look different on the surface — a price ladder, a head-to-head, an up/down round — but they reduce to a handful of families that share resolution and payout machinery.

The chart groups every type by family; the table below is the complete reference. "Early-lock" means the market can resolve the instant the outcome is locked, rather than waiting for the deadline.




Market types by family. Most families share one resolver and the same parimutuel payout.

| Type | Family | You predict | Resolves on | Early-lock |
|---|---|---|---|---|
| market_cap | Binary milestone | A token reaches $X market cap | DexScreener mcap | Yes |
| token_mcap_close | Binary milestone | A token closes above $X on a date | Mcap at the deadline | No |
| token_rank_milestone | Binary milestone | A token reaches top-N (ex- | CoinGecko rank | Yes |

Archived copy · cryp2nova.com

| Type | Family | You predict | Resolves on | Early-lock |
|---|---|---|---|---|
| | | stablecoins) by a date | | |
| chain_tps | Binary milestone | A chain hits a TPS milestone | On-chain metric | Yes |
| chain_tvl | Binary milestone | A chain hits a TVL milestone | On-chain metric | Yes |
| chain_stablecoin_mcap | Binary milestone | A chain's stablecoin supply hits $X | On-chain metric | Yes |
| dune_metric | Binary milestone | An on-chain usage metric clears a line | Dune query | Yes |
| external_metric_milestone | Binary milestone | A partner-API metric reaches a target by a date | Partner HTTP API | Yes |
| token_outperform | Head-to-head | Which token returns most over a window | Price returns | No |
| token_return_compare | Head-to-head | Token A beats token B on return | Price returns | No |
| token_mcap_flip | Head-to-head | Token A passes token B in market cap | DexScreener mcap (by address) | Yes |
| chain_dex_volume_compare | Head-to-head | Which chain does more DEX volume | Dune volume | Yes |
| token_rank_race | Head-to-head | Which token is first into the top-N | CoinGecko rank | Yes |
| token_mcap_range | Ladder | Which band a token's closing mcap lands in | Mcap at the deadline | No |
| token_price_range | Ladder | Which band a token's closing price lands in | Price at the deadline | No |
| price_direction | Up/Down round | Whether a price closes up or down over a round | Price stream (open vs close) | No |
| dex_volume_days | Count-the-days | How many days a chain's DEX | Dune daily volume | Yes |

Archived copy · cryp2nova.com

| Type | Family | You predict | Resolves on | Early-lock |
|---|---|---|---|---|
| | | volume clears a line | | |
| launchpad_rank_days | Count-the-days | Days a launchpad token is #1 by volume | Observed rank | Yes |
| launchpad_volume_winning_days | Count-the-days | Launchpad daily-volume winning days | Dune daily volume | Yes |
| launchpad_token_mcap_count | Count-the-days | How many launches clear a market-cap bar | Observed mcap | Yes |
| volume_crossing_date | Crossing window | The window a metric first crosses a line | Dune / CoinGecko / DeFiLlama | Yes |
| token_basket_mcap | Combined basket | Whether a basket's summed cap hits $X | Summed DexScreener mcaps | Yes |
| event_outcome | Manual event | The outcome of a real-world event | Human resolution + proof | No |
| event_versus | Manual event | The winner of a named head-to-head contest | Human resolution + proof | No |

Every market type

CRYP2NOVA

## Binary milestones

Milestone markets ask a single yes-or-no question: will a token, a chain, or a metric cross a line by a date? They settle YES the instant the line is crossed (early-lock) and NO if the deadline arrives untouched.

This family covers token market-cap targets, top-N rank milestones (excluding stablecoins), chain TPS / TVL / stablecoin-supply milestones, and arbitrary on-chain usage metrics read from Dune.

Peak vs close

A milestone locks YES on touch — any reading at or above the line settles it. Its sibling token_mcap_close is the opposite: it reads the value at the deadline, so a spike that falls back resolves NO.

Archived copy · cryp2nova.com

## Range ladders

A ladder splits a range into N bands and asks which one a token will close in. Market-cap ladders (token_mcap_range) and price-strike ladders (token_price_range) share one engine; only the metric differs.

Ladders are end-state markets: they read the closing value at the deadline, so they don't early-lock. Each band is its own outcome, and the parimutuel pool is split among everyone who picked the winning band.

- Bands are usually centred on the live value so the current reading sits mid-ladder.
- Exactly one band wins; it's an N-way market, not YES/NO.
- Resolution reads the latest value at or before the deadline.

## Head-to-heads & flips

Head-to-heads pit two or more subjects against each other. Will token A out-return token B? Will A flip B in market cap? Which chain does more DEX volume? Which token reaches a top-N rank first?

Flips and rank races early-lock the moment the cross is confirmed; return comparisons are end-state and settle at the deadline. Flips resolve by mint address via a chain-agnostic search, so any two tokens — including Solana pump.fun launches — work with no resolver changes.

- token_outperform — best performer among a field over a window (N-way).
- token_return_compare — A vs B on return (binary, end-state).
- token_mcap_flip — A passes B in market cap (binary, early-lock by address).
- chain_dex_volume_compare — chain vs chain on DEX volume.
- token_rank_race — first token into the top-N by market cap (N-way, no deadline).

## Up/Down rounds

Up/Down markets (price_direction) are recurring rounds on whether a token closes a window up or down. They run on cadences from every few minutes to weekly and monthly, and a tick resolves and pays each round as its window closes.

These markets label their outcomes UP and DOWN — never YES/NO. A round where the open and close are exactly equal is a flat tie: it resolves to a VOID sentinel and refunds everyone in full, closing the one-sided-book trap.

Flat ties refund

If a round closes exactly where it opened, nobody was right. The round goes VOID and every participant is refunded their full gross stake.

## Count-the-days markets

Archived copy · cryp2nova.com

Count-the-days markets ask whether a threshold number of qualifying days will occur inside a window: days a chain's DEX volume closes over a line (dex_volume_days), days a launchpad token leads by volume, or how many launches clear a market-cap bar.

They're early-lockable count-ups: as soon as the required number of days banks, the market resolves YES — it doesn't need to wait for the window to end.

## Crossing-date windows

A volume_crossing_date market asks not whether but when: in which window does a metric first cross a line? Outcomes are date windows, and the market resolves to the window containing the first confirmed crossing.

The source is pluggable — a Dune cumulative total, a Dune level series, a daily-peak count, a CoinGecko supply or price, or a DeFiLlama total. Non-monotonic sources (price, on-chain supply) are safe because a first crossing needs two readings at least 30 minutes apart, so a single flash wick can't settle it.

## Combined-cap baskets

A token_basket_mcap market sums the market caps of a set of tokens and asks whether the combined figure reaches a target. It's a milestone over many subjects at once — handy for an ecosystem or a launchpad cohort.

Members without a tradable pair are excluded up front so a missing price can't brick settlement. The combined cap is read from DexScreener for each member and summed.

## Manual real-world events

Some questions have no automated feed: the outcome of a real-world event (event_outcome) or the winner of a named head-to-head contest (event_versus). These stay pending until a human records the resolution out-of-band, with proof links attached.

The trading and payout machinery is identical to the automated markets — event_versus reuses the same parimutuel book and payout authority as the on-chain head-to-heads. Only the resolution step is manual.

Never an LLM in the money path

Manual resolution is a recorded human decision with evidence — never a model's guess. LLM output never settles a market.

## How resolution works

Resolution is deterministic. A resolver is a pure function: given a market and a set of timestamped observations, it returns an outcome (or stays pending). Nothing about the result depends on who runs

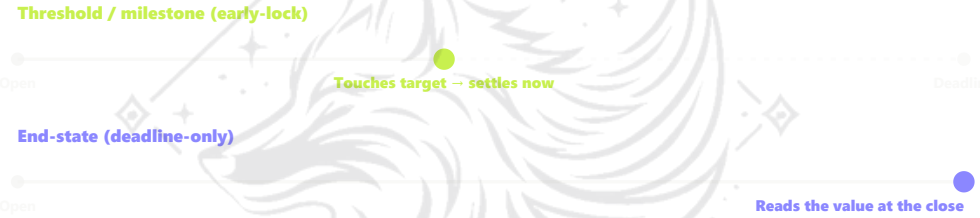
Archived copy · cryp2nova.com

it or when, beyond the clock it compares the deadline against.

Readings come from a small set of sources, land in an observation store, and the resolver reads from there. Threshold markets resolve the instant the outcome is locked; end-state markets refuse to resolve until the deadline.




Sources feed a timestamped observation store; a pure resolver turns it into an outcome.




Threshold markets early-lock on touch; end-state markets read the value at the close.

| Source | Feeds |
| --- | --- |
| observation_store | DexScreener / CoinGecko / on-chain readings, snapshotted on a schedule. |
| dune_window | Dune Analytics queries, refreshed daily so /results never freezes. |
| price_direction_stream | The price feed that opens and closes up/down rounds. |
| manual | A recorded human resolution for real-world events. |

Resolution sources

Confirmation guards

Early-lockable resolvers can be fooled by one transient outlier tick. Flip-style resolvers require multiple confirming readings (and a crossing needs two reads ≥30 min apart) before they lock, so a single wick can't false-settle a market.

## Payouts & fees

Hunch is parimutuel: with two or more participants, the entire net pool — every stake on both sides, net of the entry fee — is split among the winning side pro-rata by net stake. A payout can never

Archived copy · cryp2nova.com

exceed the pool the market collected, so the house carries zero exposure.

All of this comes from one pure function, computeMarketPayouts, which is the single source of truth for every rail. The lib runner is the only thing that touches the database or the chain.




A worked parimutuel settlement: the net pool splits among winners pro-rata by stake.

Entry fee

2%

House exposure

$0

Claim step

None

- Two or more participants → parimutuel split, pro-rata by net stake, minus the 2% fee.
- Single participant → full gross refund (the fee is returned), whichever way it resolved — there's no counterparty to win from.
- Flat-tie VOID round → every participant refunded in full gross.
- Payouts are pushed automatically on resolution; there is no claim transaction.

One payout authority

Base, Sui, and the Arbitrum vault all settle through the same computeMarketPayouts logic, differentially tested against the on-chain vault. The money math lives in exactly one place.

# Platform features

Everything around the bet — your stack, proof, social reach, and the rails underneath.

### Portfolio & Stack

Archived copy · cryp2nova.com

Your Stack is every position you hold and everything you've settled. Open positions show the realizable payout if the market resolved now — computed from the same parimutuel logic that pays you — so a sole $9 bet shows $9, not an imaginary multiple.

Settled markets show what you actually won or got back, with a link to the on-chain receipt.

Open your stack→

## Proof & receipts

There's no off-chain ledger to trust. Every bet and every payout is a USDC transfer on Base, and each one carries a proof link you can open on BaseScan. Shared proof pages let anyone verify a specific result.

Because settlement is on-chain, the money path is legible end to end: stake in, pool, payout out.

## Leaderboard & streaks

The leaderboard ranks the sharpest callers, and streaks reward consistency. It's a lightweight social layer on top of real, settled results — not vanity metrics.

See the leaderboard→

## Referrals

Referrals turn a shared link into attribution: invite friends and earn from the volume they bring. Referral links route into the same one-tap market pages as any other entrypoint, with attribution preserved.

Get your referral link→

## Social entrypoints

Links shared on X and Telegram route straight into one-tap market pages with attribution preserved, so a tag becomes a bet in a single step.

Tag @bankrbot on a token and Hunch surfaces the related markets to back — the same markets you'd find in the explorer, discoverable from the conversation.

X

Follow @playhunchxyz and bet straight from a shared link.

Open X↗

Telegram

Join the channel; market links open one-tap pages.

CRYP2NOVA

Archived copy · cryp2nova.com

Open Telegram↗

Bankr

Tag @bankrbot on a token to discover and back its markets.

## Base App Mini App

Hunch ships as a Mini App inside the Base App using the standard web-app model. Inside the Base App your wallet auto-connects, so discovery and betting work without a separate connect step.

It's the same app and the same markets — just embedded where Base users already are.

## Settlement rails

Markets settle on a rail. Base USDC is the default and the live money path. Sui runs a live mainnet Move vault with user-pays-gas betting. An Arbitrum parimutuel vault runs as a parallel rail behind a flag.

Whatever the rail, payouts are computed by the same pure logic — the rail only changes where value moves, never how winners are decided.




Three rails, one payout authority. The rail changes where value settles, not how winners are computed.

| Rail | Asset | Status |
| --- | --- | --- |
| Base | USDC (gasless, relayed) | Live — default |
| Sui | Mainnet Move vault (user pays gas) | Live |
| Arbitrum | Parimutuel vault | Dark (flag-gated) |

Rails

## Safety & trust

Two rules anchor the system. First, pricing and resolution are deterministic — the agent never invents a market or a price, and an LLM's output never settles a bet or moves money. Second, the payout math lives in one audited function shared by every rail.

Archived copy · cryp2nova.com

On the surface, the app respects reduced-motion preferences and is built mobile-first with accessible controls.

- LLM use is confined behind a client port and never trusted into a money path.
- Pricing is deterministic; the agent only routes within markets Hunch already lists.
- Animations back off for users who ask for reduced motion (WCAG 2.3.3).

# Build on Hunch

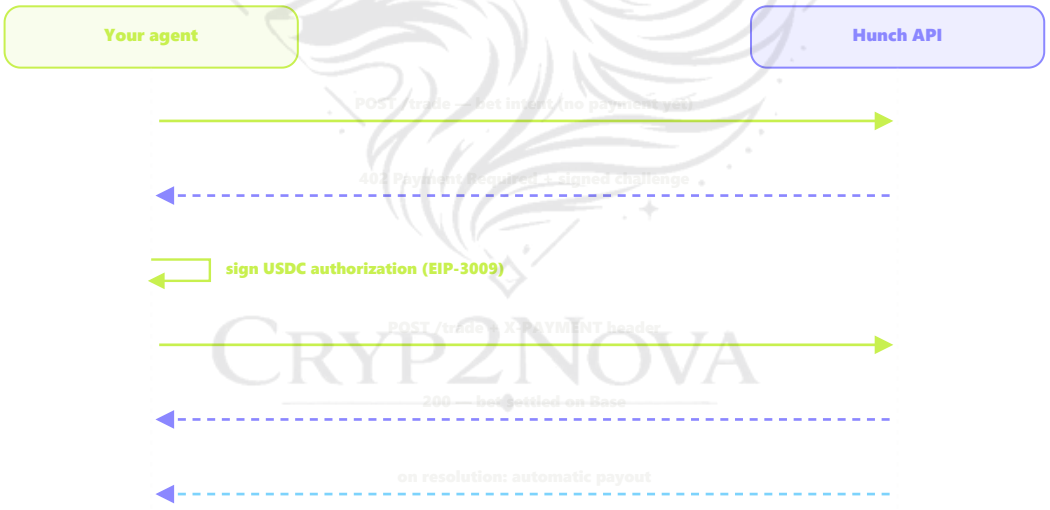
Ship autonomous agents and integrations against the keyless x402 rail.

### Agent platform

Live

The agent platform lets any wallet-holding agent discover a market, research it, simulate a bet for $0, place a real one with a single x402 USDC payment on Base, and get paid automatically on resolution — with no API key, no signup, and no claim step.

It's exposed two ways over the same audited core: a REST API at /api/agent/v1 and a remote MCP server at /api/mcp.




The x402 handshake: an unpaid intent gets a 402 challenge; the agent signs USDC and resubmits to settle.

- Keyless: identity is the wallet that signs the x402 payment — no key, no signup.
- Simulate for $0 before risking funds; place_bet defaults to a simulation.
- No cap on size; bets over $10 carry a 60s quote lock plus a slippage bound.
- Winners are paid automatically — there is no claim step.

Open the agent platform→

### MCP for agents

Archived copy · cryp2nova.com

1-line install

Add the remote MCP server and Hunch shows up as native tools inside Claude, Cursor, or Codex — discover and research markets, simulate a bet for free, and place real USDC bets, all from your assistant.

The tools map one-to-one onto the REST routes; only place_bet moves money, and it defaults to a $0 simulation so an agent has to opt into real funds.

Add the remote MCP server

```
claude mcp add --transport http hunch https://www.playhunch.xyz/api/mcp
```

Install the MCP server→

## CLI & SDKs

Any runtime

You can drive the whole platform from a terminal: npx hunch-agent lists markets, researches them, simulates bets for $0, settles real x402 USDC bets, and even runs the tested continuous strategy — every command takes --json so shell scripts can pipe it into jq.

Prefer a library? The TypeScript and Python clients run the whole x402 handshake for you, and the scaffolder writes a runnable agent project that ships its own Dockerfile and scheduled GitHub Actions workflow, so it hosts anywhere.

Pick your one-liner

```
npx hunch-agent markets # look around — keyless, free npx hunch-agent bet <id> yes 5 # $0 simulation (add --live for real USDC) npx create-hunch-agent my-bot --loop # scaffold a deployable 24×7 agent
```

- npm i @hunchxyz/agent-sdk — typed TS client: x402 loop, webhook verifier, SSE, crowd-conviction signal.
- pip install hunch-agent — the Python equivalent (httpx + eth-account).
- Scaffolded agents host on GitHub Actions (free cron), Railway, Fly.io, Render, or any VPS via the bundled Dockerfile.

See install commands & docs→

## Agent API reference

The complete reference: every REST route and MCP tool, the x402 payment handshake, sizing tiers and fees, the events system (webhooks, SSE, and polling), and the full error catalogue.

It's generated from the same modules that produce the live manifests, so the docs can never drift from the running API.

CRYP2NOVA

Archived copy · cryp2nova.com

Read the API reference→

## Hunch Intelligence on x402 Cloud

New

Hunch Intelligence is a pool-weighted prediction-market sentiment score for any token, synthesised from every live Hunch market's odds and real on-chain betting depth. It's free on Hunch at /api/partner/intel and packaged as a paid, agent-discoverable endpoint on Bankr's x402 Cloud, settling USDC on Base.

Discover and quote stay free — they feed the bet funnel; only the standalone signal is metered.

Read the integration docs→

## Base MCP

Base MCP exposes public read endpoints so agent clients can search Hunch markets, fetch quotes, and open quick-trade links — no payment and no key required.

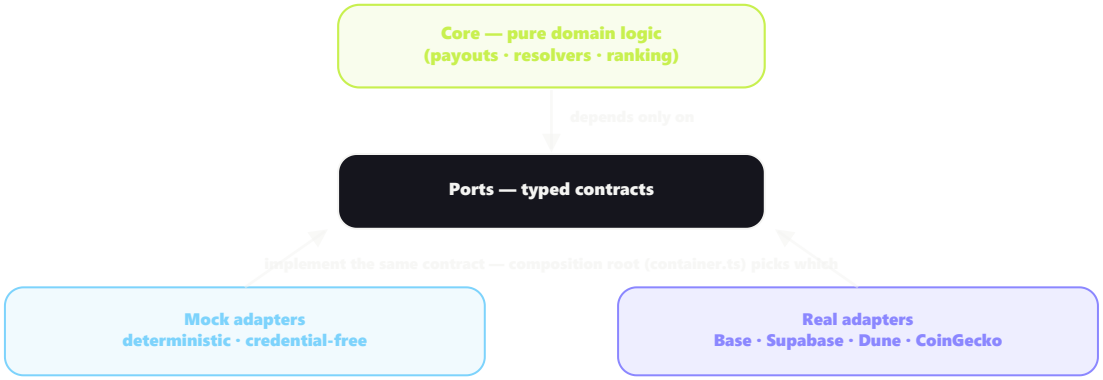
It's the read-only companion to the full agent platform: discover here, settle there.

Open Base MCP→

## Architecture

Hunch is built ports-and-adapters. The core is pure domain logic — payouts, resolvers, ranking — and depends only on typed ports. Adapters implement those ports, and a single composition root picks which adapter is wired in.

Everything ships mock-first: deterministic, credential-free mock adapters drive the same contract tests as the real ones, so a new chain or data source lands behind the same contract with zero core refactor.




Core depends only on ports; adapters implement them; the composition root picks which.

## Machine-readable docs

Archived copy · cryp2nova.com

Everything here is also available in machine-readable form for agents and assistants. Hand an assistant the Markdown of this page, point a client at the OpenAPI spec, or fetch the manifests directly.

- /docs.md — this entire hub as clean Markdown (use the Copy page button, or fetch it).
- /llms.txt and /llms-full.txt — the agent-oriented machine guide.
- /openapi.json — the full OpenAPI spec for the agent API.
- /.well-known/* — the agent, MCP, and x402 discovery manifests.

Fetch this page as Markdown

```
curl https://www.playhunch.xyz/docs.md
```

View this page as Markdown→

# Resources

Where Hunch is heading and how to take part.

### Future roadmap

The long-term path: X-native markets, token maturity pages, creator-minted markets, and the HUNCH token utility loop that ties fees back to the people who drive volume.

See the roadmap→

### Hunch Catalysts

Hunch Catalysts are open community challenges. Challenge #01 is the 10M Manifesto: make the case for HUNCH at $10M and tell the community how we get there.

Read the manifesto→

### Community & links

Hunch is built in the open. Follow the build on X, join the conversation on Telegram, and read the code on GitHub.

X — @playhunchxyz

Product updates and new markets.

Follow↗

Telegram

Join the community chat.

Archived copy · cryp2nova.com

Join↗

GitHub

Read the source.

Open GitHub↗

On this page


- What is Hunch?
- How a bet works
- Place your first bet
- The swipe feed
- Markets explorer
- Wallet & USDC


HHunch

Swipe prediction markets with an agent under every bet. Built on Base, settled in USDC.

## Product

MarketsSwipeStackCrewReferRanks

## Resources

DocsBase MCPCROOBankr skillRoadmapHome

## Social

X@playhunchxyzTelegram@playhunch

PlayHunchBack your hunch

Live · Powered by HunchSwipe & bet native marketsSwipe

CRYP2NOVA

Archived copy · cryp2nova.com