Overview

Slinky is a revolutionary enshrined oracle built for the highest performance DeFi. It leverages Neutron chain security to provide guaranteed per-block price updates with millisecond refresh rates.

Note: you can find more info about Slinky and how it works in the official Skip's Slinky docs:https://docs.skip.money/slinky/overview

F.A.Q

I'm a validator, how should i run Slinky's Sidecar?

Please usethe official Slinky's documentation how to get the Sidecar up and running.

How can i access prices from the Slinky?

Slinky prices are stored within thex/oracle module by default.

These prices are updated on a per-block basis, when there is sufficient difference from the last block's price. They can be accessed natively byCosmWasm smart contracts , by other modules, or by anyone else who has access to chain state.

Slinky market configuration is stored in thex/marketmap . This module, unlikex/oracle , doesnot store price data. Instead, it stores which currency pairs are supported and how they are configured.

Getting Supported Assets

You can find out which assets are supported on Neutron by either running:

1. (via a local running chain):curl http://:1317/slinky/marketmap/v1/marketmap
2. (via chain app CLI):neutrond q marketmap marketmap
3. (via gRPC):grpcurl -plaintext :9090 slinky.marketmap.v1.Query/MarketMap

This will return a JSON list of supported assets with associated metadata.

Accessing Slinky Prices over node APIs and RPC

To accessall Slinky prices (as of the last committed block), you can run:

1. (via a local running chain):curl http://:1317/slinky/oracle/v1/get_prices?currency_pair_ids=ADA%2FUSD&currency_pair_ids=ADA%2FUSD
2. (via gRPC):grpcurl -plaintext :9090 slinky.oracle.v1.Query/GetPrices

To get aspecific currency pair, you can call:

1. (Get all currency pairs request)neutrond q oracle currency-pairs
2. (Get price request)neutrond q oracle price [base] [quote]

Price Metadata within Slinky

When callinggetPrices via the above methods, you are returned an array ofGetPriceResponse , each of which contains the following metadata about individual prices:

1. QuotePrice
2. nonce
3. decimals
4. ID

GetPriceResponse looks like this:

// GetPriceResponse is the response from the GetPrice grpc method exposed from // the x/oracle query service.

message GetPriceResponse { // QuotePrice represents the quote-price for the CurrencyPair given in // GetPriceRequest (possibly nil if no update has been made) QuotePrice price = 1 [(gogoproto.nullable) = true]; // nonce represents the nonce for the CurrencyPair if it exists in state uint64 nonce = 2; // decimals represents the number of decimals that the quote-price is // represented in. For Pairs where ETHEREUM is the quote this will be 18, // otherwise it will be 8. uint64 decimals = 3; // ID represents the identifier for the CurrencyPair. uint64 id = 4; } InsideQuotePrice , you can fetch for the currency-pair:

1. price

2. timestamp of last update
3. blockheight of last update

QuotePrice looks like this:

// QuotePrice is the representation of the aggregated prices for a CurrencyPair, // where price represents the price of Base in terms of Quote message QuotePrice { string price = 1 [(cosmos_proto.scalar) = "cosmos.Int", (gogoproto.customtype) = "cosmossdk.io/math.Int", (gogoproto.nullable) = false];

// BlockTimestamp tracks the block height associated with this price update. // We include block timestamp alongside the price to ensure that smart // contracts and applications are not utilizing stale oracle prices google.protobuf.Timestamp block_timestamp = 2 [(gogoproto.nullable) = false, (gogoproto.stdtime) = true];

// BlockHeight is height of block mentioned above uint64 block_height = 3; }

Slinky Price Update Timing

Prices within Slinky committed on a one-block delay, since validators use the vote extensions from blockn-1 to securely submit their price data for blockn .

Most of the time, Slinky feeds will update every single block. This happens when there is a non-insignificant update to the price, and over 66% of validators are correctly running their price-fetching sidecar.

However, an individual Slinky feed will not update on any given block if:

1. (More common) There was not a sufficient enough change from last block's price to surpass the Slinkymin_price_change
2. parameter. To save network bandwidth, the price won't update.
3. (Less common) The market is disabled (i.e. not updating) withinx/marketmap
4. (Less common) Less than 2/3s of validators (by stake weight) contributed to a price update. This can happen if not enough validators run the Slinky sidecar, or there is a massive, widespread outage across providers.

Q: Can I get historical prices in Slinky?

A: No, thex/oracle module only stores the most recently posted price. However, you can use blockchain indexers or inspect past blocks to see the prices committed on previous heights. Previous Overview Next Overview