Price Feeds

Price feeds enabled by an oracle serve as a bridge between real-world data feed and the blockchain. They provide real time pricing data that is aggregated from multiple trusted external sources (e.g. crypto exchanges, financial data providers, etc.) to the blockchain network.

For the example in this book section, we will use Pragma Oracle to read the price feed for `ETH/USD` asset pair and also showcase a mini application that utilizes this feed.

[Pragma Oracle](https://www.pragma.build/) is a leading zero knowledge oracle that provides access to off-chain data on Starknet blockchain in a verifiable way.

Add Pragma as a Project Dependency

To get started with integrating Pragma on your Cairo smart contract for price feed data, edit your project's `Scarb.toml` file to include the path to use Pragma.

```toml
[dependencies]
pragma_lib = { git = "https://github.com/astraly-labs/pragma-lib" }
```

Creating a Price Feed Contract

After adding the required dependencies for your project, you'll need to define a contract interface that includes the required pragma price feed entry point.

Contract Interface

```cairo,noplayground
{{#include ../listings/ch16-building-advanced-starknet-smart-contracts/listing_08_price_feed/src/lib.cairo:price_interface}}
```

Of the two public functions exposed in the `IPriceFeedExample`, the one necessary to interact with the pragma price feed oracle is the `get_asset_price` function, a view function that takes in the `asset_id` argument and returns a `u128` value.

Pragma Dependency Import to Contract

```cairo,noplayground
{{#include ../listings/ch16-building-advanced-starknet-smart-contracts/listing_08_price_feed/src/lib.cairo:pragma_lib}}
```

The snippet above shows the necessary imports you need to add to your contract module in order to interact with the Pragma oracle.

Required Price Feed Function Impl in Contract

```cairo,noplayground
{{#include ../listings/ch16-building-advanced-starknet-smart-contracts/listing_08_price_feed/src/lib.cairo:price_feed_impl}}
```

The `get_asset_price` function is responsible for retrieving the price of the asset specified by the `asset_id` argument from Pragma Oracle. The `get_data_median` method is called from the `IPragmaDispatcher` instance by passing the `DataType::SpotEntry(asset_id)` as an argument and its output is assigned to a variable named `output` of type `PragmaPricesResponse`. Finally, the function returns the price of the requested asset as a `u128`.

Example Application Using Pragma Price Feed

```cairo,noplayground
{{#include ../listings/ch16-building-advanced-starknet-smart-contracts/listing_08_price_feed/src/lib.cairo:here}}
```

> Note: Pragma returns the value of different token pairs using the decimal factor of 6 or 8. You can convert the value to the required decimal factor by dividing the value by \\({10^{n}} \\), where `n` is the decimal factor.

The code above is an example implementation of an applications consuming a price feed from the Pragma oracle. The contract imports necessary modules and interfaces, including the `IPragmaABIDispatcher` for interacting with the Pragma oracle contract and the `ERC20ABIDispatcher` for interacting with the ETH ERC20 token contract.

The contract has a `const` that stores the token pair ID of `ETH/USD`, and a `Storage` struct that holds two fields `pragma_contract` and `product_price_in_usd`. The constructor function initializes the `pragma_contract` address and sets the `product_price_in_usd` to 100.

The `buy_item` function is the main entry point for a user to purchase an item. It retrieves the caller's address. It calls the `get_asset_price` function to get the current price of ETH in USD using the `ETH_USD` asset ID. It calculates the amount of ETH needed to buy the product based on the product price in USD at the corresponding ETH price. It then checks if the caller has enough ETH by calling the `balance_of` method on the ERC20 ETH contract. If the caller has enough ETH, it calls the `transfer_from` method of the `eth_dispatcher` instance to transfer the required amount of ETH from the caller to another contract address.

The `get_asset_price` function is the entry point to interact with the Pragma oracle and has been explained in the section above.

You can get a detailed guide on consuming data using Pragma price feeds on their [documentation](https://docs.pragma.build/Resources/Starknet/data-feeds/consuming-data).