Creating a Data Source

In this section, we will take a look at how to create a data source.

Writing the data source

In order for the data source to run, always make sure to add a shebang line containing#!/usr/bin/env python3 and to print the function output. A simpleHello World! example can be seen below.

!/usr/bin/env python3

import sys

def

main ( ) : return

"Hello World!"

if **name** ==

"**main**" : try : print (main ( )) except Exception as e : print (str (e) ,

file

sys . stderr) sys . exit (1)

More Examples

However, in order to better understand the structure and implementation of a data source, let's look at two more examples below: Gold Price and Token Total Supply.

Gold Price

First, let's look at a data source that queriesGoldPrice.org for the current price of gold. The script itself is written in Python and although this specific data source does not expect any arguments, a data source can have any number of arguments required.

!/usr/bin/env python3

import requests import sys

URL

"https://data-asg.goldprice.org/dbXRates/USD" HEADERS =

{ 'User-Agent' :

'curl/7.64.1' }

def

main ( ) : try : pxs = requests . get (URL , headers = HEADERS) . json ( ) return pxs ['items'] [0] ['xauPrice'] except Exception as e : raise e

if **name** ==

"**main**" : try : print ('{0:.2f}' . format (main ( ))) except Exception as e : print (str (e) ,

file

sys . stderr) sys . exit (1)

Token Total Supply

As mentioned above, a data source can take any number of arguments. The example below, gives an example of a data source which requires two arguments. The data source shown below queries the given network's RPC endpoint for the total supply of a given token address.

!/usr/bin/env python3

import requests import sys

def

main (rpc , to) : headers =

{ "Content-Type" :

"application/json" , } data =

("""{ "jsonrpc": "2.0", "method": "eth_call", "params": [{ "to": "%s", "data": "0x18160ddd" }, "latest"], "id": 1 }""" %

(to)) response = requests . post (rpc , headers = headers , data = data ,) return

int (response . json ( ) ["result"] ,

16)

if **name** ==

"**main**" : try : print (main (sys . argv [1] , sys . argv [2])) except Exception as e : print (str (e) ,

file

sys . stderr) sys . exit (1)

More examples

For more data source examples, please refer to the ones available on ou[Mainnet](#) to get an idea of the different types of data source used on BandChain:

- [Latest crypto prices from CoinGecko](#)
- [Latest stock prices from Finage](#)
- [Latest forex prices from Alpha Vantage](#) [Previous Introduction](#) [Next Deployment](#)