Quickstart guide

Getting Started the 1inch Spot Price API

The 1inch Spot Price API allows you to retrieve on-chain prices of tradable tokens on the Ethereum network. All prices are presented in the native currency WEI. In this tutorial, we will cover how to interact with the API to get token prices using various endpoints.

Prerequisites

Before you begin, make sure you have the following:

1. A code editor to write and run the code (e.g., Visual Studio Code, PyCharm).
2. Python installed on your system.
3. Basic knowledge of Python programming and REST APIs.

Step 1: Import Required Libraries

To start, create a new Python script (e.g., `token_prices.py`) and import the necessary libraries, If you don’t have requests make sure you install it by running `pip install requests`

```
import requests
```

Step 2: Get Prices for Whitelisted Tokens

The first endpoint allows you to get prices for whitelisted tokens. These tokens are pre-defined and can be accessed without specifying any parameters. Let's implement a function to fetch these prices:

Replace your API key with the one found here

```
def get_whitelisted_token_prices():
    url = "https://api.1inch.dev/price/v1.1/1"

    response = requests.get(url, headers={'Authorization': f'Bearer YOUR_API_KEY'})
    if response.status_code == 200:
        prices = response.json()
        print("Prices for whitelisted tokens:")
        for token_address, price in prices.items():
            print(f"{token_address}: {price}")
    else:
        print("Failed to fetch token prices.")
```

Step 3: Get Prices for Requested Tokens

The second endpoint allows you to request prices for specific tokens. To do this, you need to pass an array of token addresses in the request body. Let's implement a function to get prices for requested tokens:

```
def get_requested_token_prices(tokens):
    url = "https://api.1inch.dev/price/v1.1/1"

    payload = {
        "tokens": tokens
    }

    response = requests.post(url, headers={'Authorization': f'Bearer YOUR_API_KEY'}, json=payload)
    if response.status_code == 200:
        prices = response.json()
        print("Prices for requested tokens:")
        for token_address, price in prices.items():
            print(f"{token_address}: {price}")
    else:
        print("Failed to fetch token prices.")
```

Step 4: Get Prices for Multiple Addresses

The third endpoint allows you to get prices for multiple tokens at once. You need to pass multiple token addresses separated by commas in the URL. Let's implement a function to fetch prices for multiple addresses:

```
def get_prices_for_addresses(addresses):
    url = f"https://api.1inch.dev/price/v1.1/1/{','.join(addresses)}"

    response = requests.get(url, headers={'Authorization': f'Bearer YOUR_API_KEY'})
    if response.status_code == 200:
        prices = response.json()
        print("Prices for requested tokens:")
        for token_address, price in prices.items():
            print(f"{token_address}: {price}")
    else:
        print("Failed to fetch token prices.")
```

Step 5: Test the Functions

Now that we have implemented the functions, let's test them by calling each one:

```
if __name__ == "__main__":
    # Test get_whitelisted_token_prices
    get_whitelisted_token_prices()

    # Test get_requested_token_prices
    tokens_to_request = ["0x111111111117dc0aa78b770fa6a738034120c302"]
    get_requested_token_prices(tokens_to_request)

    # Test get_prices_for_addresses
    addresses_to_fetch = ["0x111111111117dc0aa78b770fa6a738034120c302", "0xeeeeeeeeeeeeeeeeeeeeeeee
eeeeeeeeeeeeeeee"]
    get_prices_for_addresses(addresses_to_fetch)
```

Step 6: Run the Script

Save the script and run it using Python. You should see the prices for whitelisted tokens, prices for the requested tokens, and prices for the specified addresses displayed in the console.

That's it! You have successfully used the 1inch Spot Price API to fetch on-chain token prices in Python. You can further integrate this functionality into your applications to get real-time token prices and make informed decisions while trading.

Full script

Here you can find the full script with consideration of the default RPS limit

```
import requests
import time

def get_whitelisted_token_prices():
    url = "https://api.1inch.dev/price/v1.1/1"

    response = requests.get(url,  headers={'Authorization': f'Bearer YOUR_API_KEY'})
    if response.status_code == 200:
        prices = response.json()
        print("Prices for whitelisted tokens:")
        for token_address, price in prices.items():
            print(f"{token_address}: {price}")
    else:
        print("Failed to fetch token prices.")

def get_requested_token_prices(tokens):
    url = "https://api.1inch.dev/price/v1.1/1"

    payload = {
        "tokens": tokens
    }

    response = requests.post(url, headers={'Authorization': f'Bearer YOUR_API_KEY'}, json=payload)
    if response.status_code == 200:
        prices = response.json()
        print("Prices for requested tokens:")
        for token_address, price in prices.items():
            print(f"{token_address}: {price}")
    else:
        print("Failed to fetch token prices.")

def get_prices_for_addresses(addresses):
    url = f"https://api.1inch.dev/price/v1.1/1/{','.join(addresses)}"

    response = requests.get(url, headers={'Authorization': f'Bearer YOUR_API_KEY'})
    if response.status_code == 200:
        prices = response.json()
        print("Prices for requested tokens:")
        for token_address, price in prices.items():
            print(f"{token_address}: {price}")
    else:
        print("Failed to fetch token prices.")

if __name__ == "__main__":
    # Test get_whitelisted_token_prices
    get_whitelisted_token_prices()
    # sleep one second because of RPS limit
    time.sleep(1)

    # Test get_requested_token_prices
    tokens_to_request = ["0x111111111117dc0aa78b770fa6a738034120c302"]
    get_requested_token_prices(tokens_to_request)
    # sleep one second because of RPS limit
    time.sleep(1)

    # Test get_prices_for_addresses
    addresses_to_fetch = ["0x111111111117dc0aa78b770fa6a738034120c302", "0xeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee"]
    get_prices_for_addresses(addresses_to_fetch)
```

Previous
‹ Introduction

Next
Prices for whitelisted tokens ›