**DataScientest**

**Practical project:**

**OPA**

**Project mentor:** Simon C.

**Team:** Soumia, Julius, Maciej

**Description :**

Nowadays, the world of crypto is starting to take on an important and growing role. These are simply quite volatile and unstable financial markets based on Blockchain technology. The main goal of this project is to create a trading bot based on a Machine Learning model that will help an investor to trade a crypto asset on crypto markets. Based on the training the model should provide prediction of prices and signals when to buy/sell an asset.

**History of changes**

|  |  |  |
| --- | --- | --- |
| Date | Version | Description of change |
| 19.03.2025 | V1.0 | Initial version |
| 20.03.2025 | V1.1 | CoinCap API examples added |
|  |  |  |
|  |  |  |
|  |  |  |

Content

[1. Collecting data 2](#_Toc193383430)

[1.1. CoinCap API JSON samples 3](#_Toc193383431)

[2. Storing data (to be worked out) 5](#_Toc193383432)

[3. Machine learning model 5](#_Toc193383433)

[4. Diagrams 6](#_Toc193383434)

[5. Evolution 6](#_Toc193383435)

# Collecting data

|  |  |  |
| --- | --- | --- |
| Choices |  | Extras |
| Bionance API | Needs registration  There is a limit of **300 connections per attempt every 5 minutes**. | Supports WebSocket |
| CoinCap API | Doesn’t need registration  Limitations:  Free Tier (No API Key)   * 200 requests per minute * 11 years historical data   Free Tier (API Key)   * 500 requests per minute * 11 years historical data   <https://docs.coincap.io/#intro> | Supports WebSocket |

Ein Bild, das Text, Screenshot, Schrift, Dokument enthält.

KI-generierte Inhalte können fehlerhaft sein.

## CoinCap API JSON samples

* + 1. **GET/candles**

Request

| Key | Required | Value | Description |
| --- | --- | --- | --- |
| exchange | required | poloniex | exchange id |
| interval | required | m1, m5, m15, m30, h1, h2, h4, h8, h12, d1, w1 | candle interval |
| baseId | required | ethereum | base id |
| quoteId | required | bitcoin | quote id |
| start | optional | 1528410925604 | UNIX time in milliseconds. omiting will return the most recent candles |
| end | optional | 1528411045604 | UNIX time in milliseconds. omiting will return the most recent candles |

Response

| Key | Description |
| --- | --- |
| open | the price (quote) at which the first transaction was completed in a given time period |
| high | the top price (quote) at which the base was traded during the time period |
| low | the bottom price (quote) at which the base was traded during the time period |
| close | the price (quote) at which the last transaction was completed in a given time period |
| volume | the amount of base asset traded in the given time period |
| period | timestamp for starting of that time period, represented in UNIX milliseconds |

Example:

Request:

curl --location 'api.coincap.io/v2/candles?exchange=poloniex&interval=h8&baseId=ethereum&quoteId=bitcoin%0A'

Response:

{

"data": [

{ "open": "0.0708000000000000", "high": "0.0710450000000000",

"low": "0.0706434000000000", "close": "0.0708350100000000",

"volume": "1818.1433015300000000", "period": 1530720000000 },

{ "open": "0.0708316700000000", "high": "0.0715192000000000",

"low": "0.0706544100000000", "close": "0.0712472600000000",

"volume": "1649.2534471200000000", "period": 1530748800000},

],

"timestamp": 1533581190540

}

* + 1. **GET/assets/{{id}}/history**

Request

| Key | Required | Value | Description |
| --- | --- | --- | --- |
| id | required | bitcoin | asset id |
| interval | required | m1, m5, m15, m30, h1, h2, h6, h12, d1 | point-in-time interval. minute and hour intervals represent price at that time, the day interval represents average of 24 hour periods (timezone: UTC) |
| start & end | optional | 1528470720000 | UNIX time in milliseconds. omitting will return the most recent asset history. If start is supplied, end is required and vice versa |

Response

| Key | Description |
| --- | --- |
| priceUsd | volume-weighted price based on real-time market data, translated to USD |
| time | timestamp in UNIX in milliseconds |

Example:

Request:

curl --location 'api.coincap.io/v2/assets/bitcoin/history?interval=d1'

Response:

{

"data":

[

{ "priceUsd": "6379.3997635993342453", "time": 1530403200000 },

{ "priceUsd": "6466.3135622762295280", "time": 1530489600000},

{ "priceUsd": "6601.0724971279524219", "time": 1530576000000},

{ "priceUsd": "6581.0092630267574887", "time": 1530662400000},

…

],

"timestamp": 1533581103627

}

There is no need to go for web scraping.

Let us start with three crypto coins:

* XRP:Ripple (EUR, USD)
* ETH:Ethereum (EUR, USD)
* PEPE:Pepe coin (EUR, USD)

# Storing data (to be worked out)

I would suggest to store historical data in PostgreSQL database. Do we want to store data in one table for all coins or we use one table per coin? We can always go for one table and use VIEWS to separate coins data.

We should store historical data using the minimal data period, which is a minute. It allows us to aggregate it for longer periods like 3min, 5min, 10min, etc. by ourselves without generating additional API traffic or we store data in different tables to avoid programmatical aggregation.

I think we are not going to store streaming data. We can always do a manual request by user to update historical data (always with the smallest possible time granularity ) causing the automated model training execution.

# Machine learning model

I would like to train the model based on different time periods:

* 5 min, 10 min, 15 min, etc.
* The model should be retrained automatically after historical data has been updated.

# Diagrams

Ein Bild, das Diagramm, Reihe, Text enthält.

KI-generierte Inhalte können fehlerhaft sein.

# Evolution

* Monitoring
* Kafka