VINCIT

Rising Star Pre-assignment

Scrooge McDuck is once again requesting consultation from fellow ducks at Vincit.

This time Scrooge has his eyes on cryptocurrency — bitcoin to be exact — and he needs a tool to analyze its market value for a given date range.

Your mission, should you choose to accept it, is to create an application that meets Scrooge's needs. You are free to use any technology of your choosing. The resulting application can be for example a web page, an API backend, a mobile application, or anything else you deem suitable.

Application

Scrooge wants to use the application to get the following information for different date ranges he is analyzing:

Additional information:

- *Both start and end dates should be included in a date range.*
- *A day's price means the price at 00:00 UTC time (use price data from as close to midnight as possible as the day's price, if you don't have a datapoint from exactly midnight).*
- *Allow the user of your application to pass the start and end dates of the date range in some way, e.g. via input fields in a UI or as parameters to an API.*

A. **How many days is the longest bearish (downward) trend within a given date range?**

  - Definition of a downward trend shall be: "Price of day N is lower than price of day N-1"
  - Expected output: The maximum amount of days bitcoin's price was decreasing in a row.

  Example: In bitcoin's historical data from CoinGecko, the price decreased **2** days in a row for the inputs from 2020-01-19 and to 2020-01-21, and the price decreased for **8** days in a row for the inputs from 2020-03-01 and to 2021-08-01.

B. **Which date within a given date range had the highest trading volume?**

  - Expected output: The date with the highest trading volume and the volume on that day in euros.

C. **Scrooge has access to Gyro Gearloose's newest invention, a time machine. Scrooge wants to use the time machine to profit from bitcoin. The application should be able to tell for a given date range, the best day for buying bitcoin, and the best day for selling the bought bitcoin to maximize profits. If the price only decreases in the date range, your output should indicate that one should not buy (nor sell) bitcoin on any of the days. You don't have to consider any side effects of time travel or how Scrooge's massive purchases would affect the price history.**

  - Expected output: A pair of days: The day to buy and the day to sell. In the case when one should neither buy nor sell, return an indicative output of your choice.

Continues →

Use CoinGecko’s public API to get the needed data

https://www.coingecko.com/en/api/documentation

You will only need to use the `/coins/{id}/market_chart/range` endpoint. Read its documentation to understand how it works. Note that the API returns data with different granularity depending on the date range's length. **Tip: You should add 1 hour to the `to` input to make sure that you always get data for the end date as well.** Scrooge’s Money Bin only holds euros, so that is the only fiat currency you need to consider.

For example, the following URL can be used to fetch bitcoin’s price, market cap and volume information in euros (€) from January 1, 2020 to December 31, 2020:

```
https://api.coingecko.com/api/v3/coins/bitcoin/market_chart/range?vs_currency=eur&from=1577836800&to=1609376400
```

The answer must be returned as:

- **A link to a public Git repo in a hosting service of your choice** (GitHub, GitLab etc.) **OR**
- **A Git bundle** (You can create a bundle file from your repo by running: `git bundle create myreponame.bundle --all`)

What we value:

- Clean code
- Ease of use — Either host your solution somewhere where it can be used immediately, or include clear directions (e.g. in a README file) for running your solution.
- Simplicity — Minimize the use of external libraries and dependencies. We want to see how you manage with a programming language of your choice, not how many packages you are able to import. You are of course highly encouraged to use any conveniences or standard library utilities that ship with your chosen language. It's also fine to build your solution around a single 3rd party library or framework, if that adds value to your solution.
- Extensibility — Scrooge only wants these three features for now, but very likely wants to hire us to add capabilities to the application after it has proved its value to him.

Vincit will review the code, and we like readable and maintainable code that follows good coding conventions. You may ask if you have any questions. Have fun coding!