Vanguard IMFS Data Engineer Challenge

From: Jose Arguelles (jose@vgimfs.com)

To: premk_ekk@yahoo.com

Date: Wednesday, February 9, 2022, 01:14 PM EST

Hi Prem,

This is Jose from IMFS (we have different emails for our off network macbooks), it was a pleasure getting to know you last week! Nii Saka and the rest of the team would like to see how you do on this take-home project we've set up for our data engineering position. If you have the time, could you please get this done in a week's time (Wednesday, Feb 16th)? If you have any other questions let me know, otherwise here is the prompt with instructions below!

Thanks,
Jose Arguelles

During the execution of this test, you will develop a security-comparison tool from the Dow Jones U.S. index constituents. You will need to wrangle data from a static file and API and load them into a database of your design, with a bonus challenge of creating a user interface.

The challenge is composed of four different blocks and it was intended to test the skills we think a Fintech Engineer should have. You will use an external API, Python, SQL, and a frontend technology/framework of your choice.

1.) Extract

First of all, we want to fetch the historical data of the constituents within the **constituents_history.pkl file.**

You have to fetch all the information since January 1st of 2018, and you have to use the yfinance python library. Read the documentation to find out which API methods you will need to fetch the historical data of each constituent in the pickle file.

To do:

- Create a script in Python that given a ticker (i.e. AAPL), that returns the historical data fetched from the API.

2.) Transform & Load

We are going to store the information in a SQL database of your choice. To do that, you'll need to review the API response structure and create the correct database structure to store this information.

To do

- Using the script you have created on the Extract iteration, insert the information inside a SQL database.
- Create a database structure best suited for this data.

3.) Analyze

Using the database from step 2, calculate the index value (price) for every day that you have data in 2018, 2019, and 2020. Put this data into your database. Following this, identify the sector distribution of the index for 2018, 2019, and 2020 on a relative weight basis (can be a list, dataframe, chart, etc).

Tips

- The Dow Jones U.S. Index is a Price Weighted index - use this to understand how to calculate index value
- When identifying the sector distribution on a relative weight basis, you can look at the index constituents at year end (ie. Dec 30th).

To do

- Calculate the index value (price) for each day that you have data from 2018, 2019, and 2020 and store into your database
- Calculate the sector distribution within the index factoring their relative weights

4.) Visualize (Bonus)

*Keep in mind that you are not required to complete this challenge in order to succeed, but of course, the more you achieve, the better.

Now it's time to display this information in a chart. To do that, you are going to use any frontend technologies/frameworks you are comfortable with. (Our team uses Dash, Streamlit, and Vue.js to create UIs. Points won't be taken off if you don't use these though)

Pro tip: You may need to construct an API to be able to access your database (using a framework such as Flask or FastAPI)!

In this iteration, you have to create a single-page website that fetches the information from the database and shows this information in a chart that you think best represents the data.

The website will have the following elements:

- Dropdown with all the constituents you have in the database.
- Chart with the historical data of the current constituents selected in the dropdown.

The behavior is simple: every time you select a cryptocurrency in the dropdown, the chart has to show the related information in the database.

To do

- Create a one-page website with the following items:
- Dropdown with all the constituents.
- Chart that shows the current constituent's historical data.
- Every time you change the value in the dropdown, the chart has to render the selected constituent data.

How to submit

Upload your completed project to your GitHub, and then send a link to the repository along with any comments you have about your solution, and how to run your code.