OHLC Engine

TABLE OF CONTENTS

Problem Statement

Create a Dashboard based on which User would be able to analyze the sentiment of the specific stock. You're required to create an Analytical Server "OHLC" (Open/High/Low/Close) time series based on the 'Stock List' dataset which will be provided to you in a separate JSON format and its output should be a timely report printed in Charts. (**Note**: *You can use chartjs or any other free API for displaying Charts*).

Input

The 'Stock List' dataset which is provided in JSON format comprise of certain attributes, following is the associated meanings of some of the attributes.

```
struct OHLC{
symbol : Stock Ticker string
open : opening price Double
high : highest price Double
low : lowest price Double
close : closed price Double
volume : volume in a day Integer
date : date of transaction Date
}
```

Rest all attributes can be ignored for generating the OHLC Bar charts.

Detailed Problem Charter

a. You can use your chosen language for business logic that you are designing for back-end but it has to be well scalable.

b. You're creating a Multi-module scalable back-end engine which will be responsible for taking certain data as input and process the data to showcase the raw data in desired format.

c. Your module will have certain sub-modules which will be working independently and will communicate within themselves via *REST/Queues* which you need to decide for yourselves.

d. You'll be working on creating a Multi-Module program that has achieved the desired result.
    i. **Sub-Module-1(Reader)** : Reads the 'Stock List' data input (line by line from JSON), and sends the packet to the FSM (Finite-State-Machine) module.

    ii. **Sub-Module-2(FSM)** : FSM computes OHLC packets based on intervals and constructs 'BAR' chart data, based on timestamp.

    iii. **Sub-Module-3(Web Service)** : Client requests from Dashboard come here. Publishes (transmits) the BAR OHLC data as computed in real time.

e. You'll be creating a Dashboard on the front-end with certain UX features for allowing the users to filter and regenerate the Chart on a specific interval.
    i. User should have the ability to search the different stocks based on the symbols and view the chart.
    ii. Users should be able to filter the stock's performance on specific date range. (Weekly/Monthly)
    iii. Users should be able to view the history of their searches.
    iv. Users should have ability to switch between OHLC, Candlestick charts, Colored Bar, Vertex Line and Hollow Candle.
    v. User can see additional information about selected company. You can use this free resource to get company information by ticker symbol or use any other free API available on internet.

Submission

a. Basic README.txt : Simple Design Criteria, methods, workers, data structures etc.

b. Instructions to setup and run the solution.

c. Decent package structure.

d. Maven/Gradle or any other tool to build solution

What is OHLC?

It is an open-high-low-close chart (also OHLC) is a type of chart typically used to illustrate movements in the price of a financial instrument over time.
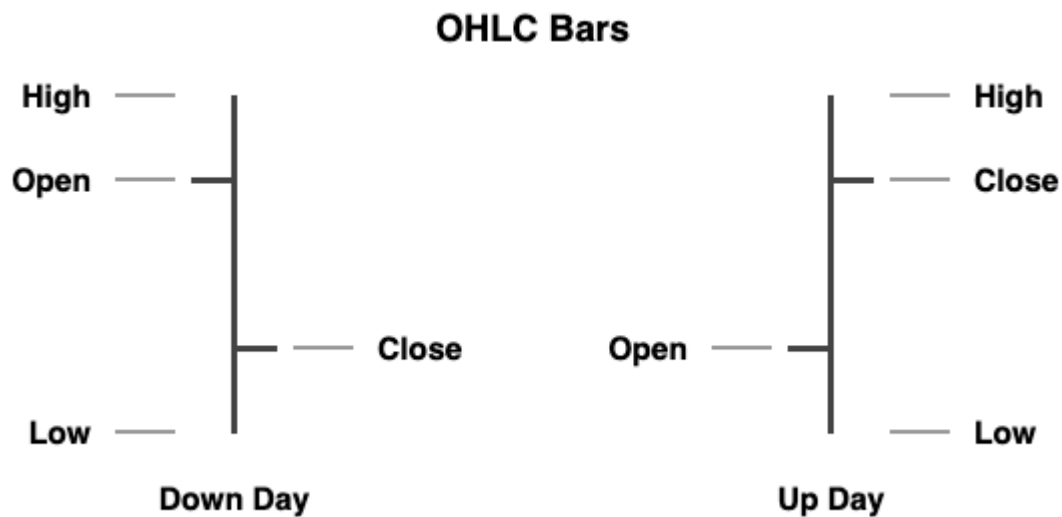
Key Takeaways

- An OHLC chart shows the open, high, low, and close price for a given period.
- It can be applied to any timeframe.

- The vertical line represents the high and low for the period, while the line to the left marks the open price and the line to the right marks the closing price. This entire structure is called a bar.

- When the close is above the open, the bar is often colored black. When the close is below the open the bar is often colored red.

Open-high-low-close Charts (or OHLC Charts) are used as a trading tool to visualize and analyses the price changes over time for securities, currencies, stocks, bonds, commodities, etc. OHLC Charts are useful for interpreting the day-to-day sentiment of the market and forecasting any future price changes through the patterns produced.

GitHub

Initial Cloning and Branch Creation

1. *Cloning* - Clone the remote-repository on your local-working directory as shown below:

   git clone https://github.com/vivekjaipur/BHU-"your team name in one word here".git

   Example :
   If your team name is Team X, your git hub repository will be as below
   https://github.com/vivekjaipur/BHU-teamx
   and command to clone repository will be as below
   git clone https://github.com/vivekjaipur/BHU-teamx.git

2. *Branch Creation* - Create your new working-branch from main. Your branch name should follow format **<*teamName-date*>**.
   ```
   git checkout -b teamXYZ-25082021
   ```

Above command checks out a branch called **"teamXYZ-25082021"** based on main, and the -b flag tells Git to create the branch if it doesn't already exist.

Push And Merge/Pull Request

Once you are done with the coding for the project, then follow these steps to raise a Pull Request on GitHub.

1. *Push* - Execute below command if newly-created working-branch **(teamXYZ-25082021)** needs to be pushed on the central repository.

```
git push -u origin teamXYZ-25082021
```

2. *Merge/Pull Request* - If the development code is ready to be reviewed, please raise a pull-request from the remote repository. In order to raise a pull-request, please follow below steps:

    a. Open the existing GitHub remote url in the browser.
    b. Under the code section, please select your working-branch(by default, it's pointed to main branch). In case, you don't see, please make sure you are referring to branches options within the leftmost-dropdown.
    c. After switching to working-branch, you may see the "Compare & pull-request" button on the top of the branch highlighted.
        Please click that button and provide title and description details for the pull-request.
    d. Once done, please click "Create pull request" button to submit the assignment for the review.

Want to read up on GitHub, please check this quick read