Stock-Monitor

Mia Jia

This project is a small system to monitor stock prices for a list of companies symbols from Yahoo Finance API.

The system contains 4 parts:

- MySQL database which stores all necessary data;
- A java program which autonomously fetches stock info and quotes from Yahoo Finance and inserts the data into the MySQL database;
- A web application which provides a RESTful API to perform all CRUD operations on companies and query the stock price history that the stand alone program has collected.
- Implement a UI for your CRUD build with JavaScript, D3, HTML/CSS and JSP.

Run Demo

How to run the project:

- Set the MySQL environment: port: 8889. (Username: root, Password:123) You can change it at JdbcUtilsSingleton.java
- Go to the project root. Run the command in terminal "mvn test tomcat:run". You may need to install maven on your environment.
- The demos are now ready to be launched.

Components

Database and DAO Layer

- I transformed the original data (YahooFinance.Stock) fetch from Yahoo into the format that can be easily processed(MyStock). Then I imported the transformed data into MySQL database through JDBC. I use four tables to present the transformed data.
  1. Stock Info (store static information of selected stock)
  2. Realtime_Stock_Data (store real time quote information of selected stock, refreshes every 10 s and contains latest quote of one stock up to 5 days)
  3. Daily_Stock_Data (store daily quote information of selected stock, refreshes everyday and contains latest quote of one stock up to 30 days)
     , price, high, low, open, close, timestamp, volume
  4. Weekly_Stock_Data (store weekly quote information of selected stock, refreshes every week and contains latest quote of one stock up to 52 weeks)

- After inserting the tuples into the tables in the database. In com.logicmonitor.msp.dao package, I implement the CRUD for MySql database through SQL queries. For example, the SQL statement \select symbol, price, high, low, open, close, timestamp from realtime_stock_data where symbol=?)" can return selected value from realtime_stock_data table.
- Except for manipulate 4 table at same time, DAO also provides some methods to manipulate separate table. The reasons are:

1. These four table require difference update frequency: Realtime_Stock_Data refreshes every 10 s; Daily_Stock_Data refreshes everyday; Weekly_Stock_Data refreshes every week. For Stock Info, it won't fresh until remove by user.
2. Different require from front end will request from user need only need to access one table. For example, if the user want to access one stock info for the last year, DAO only need the data return from Weekly_Stock_Data(which stores 1 year data with weekly interval).

- Base on the reasons list above, There's no need DAO manipulate all table same time.

Domain

- Stock_info and Stock_Price are created for mapping the Stock data from Yahoo DataFecher to information we want to store in mySQL database. The reasons to separate the info from price are listed in 'Database and DAO Layer' section.

Yahoo Data Fetcher

- Provide methods to fetch stock data from Yahoo Finance API

Service

The service module provides a single function as interface, fetchAddNewtoDB(String symbol). When this function is invoked, it fetch history price of one stock from Yahoo and save to mysql, and after that, it calls another function fetchAddtoDB(List<string> symbol) which create 3 threads fetching up-to-date price of a list of stocks and save to realtime price table, daily price table, and weekly price table in mysql.

A list of symbol is maintained in this Service module, each time fetchAddNewtoDB is invoked, this list is updated.

Web Application

- The project includes the following client-side functionalities:
  - Add a new symbol to be monitored.
  - Delete an existing company from monitored list.
  - List all companies that exist in database.
  - Get company's history performance (last year/ last month);

Data Visualization and Interaction using D3(under-developement)

- **Drop down menu:** User can select any valid attributes for y axis, chart type and time range.
- **Tooltip:** With the help of tooltip, we can see more information we want when we move mouse over the element in the chart.
- **Timeline:** User could zone in and zone out timeline to get different view.
- **Sort**: When chosen bar chart, you can sort the elements in the chart.
- For more sophisticated charts and analyses could be implemented very easy by using D3 library.