CAB230 Stocks API – Client

Super Stocks

Connor McHugh

n10522662

Table of Contents

Re: The video demonstration:

Unfortunately, due to the fact that I do not own a microphone, I was not able to do a voiceover for the video, but all points that I would have discussed will be mentioned somewhere in the report. Also, in the video, when I demonstrate filtering by date, the selector pop-ups do not show up in the OBS window capture, please be assured that the selectors do in-fact work.

Introduction

Purpose & Description

The *Super Stocks* web application is designed to allow users to view and analyse stock market statistics taken from the REST API. The application provides a number of ways of making this easier and more accessible to the user, such as the ability for filter stocks by industry, and view statistics in a certain date range. The latter of these techniques requires the user to have registered an account and to be logged in.

In the making of this web application, it was decided that a number of modules would be used in order to enhance its functionality/aesthetics. The first of these is ag-Grid-react which was used to create nicely styled tables with many customisable features. The second of these modules is reactstrap, which is a react implementation of the ever-popular Bootstrap framework. This was used for most of the components and controls in the application in order to create an attractive and uniform look and feel.

Completeness & Limitations

The *Super Stocks* web application uses react routing for navigation, and is capable of handling both the unauthenticated and authenticated query routes, as well as the user endpoints. The data fetched from the Query endpoints is presented in an ag-Grid table, user input is handled by the reactstrap form components.

Use of End Points

/stocks/symbols

This end point is used to fetch all of the stocks when first going to the *stocks* page. It also allows for the results to be filtered by industry.

/stocks/{symbol}

This end point is used when an unauthenticated user clicks on a stock in the main stock list. It shows the latest information from the selected stock.
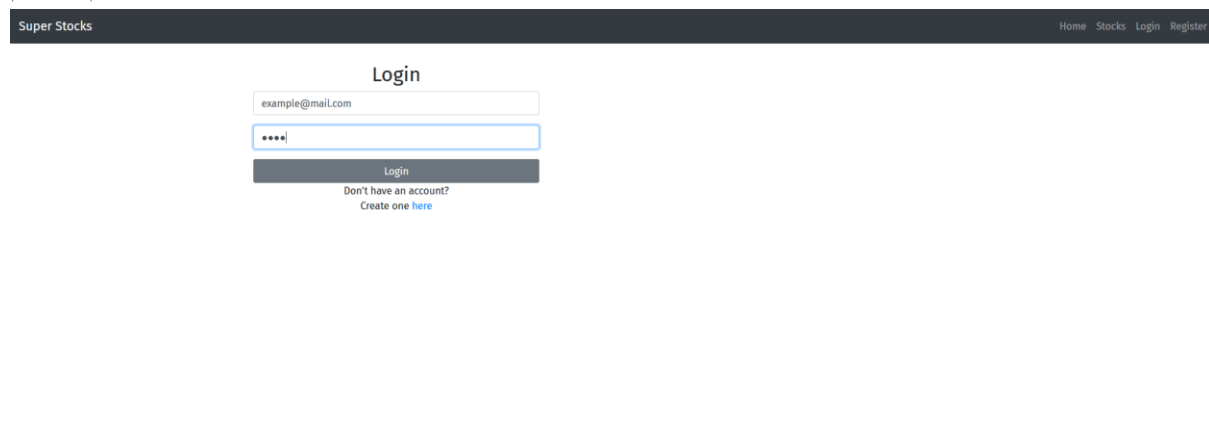
/stocks/authed/{symbol}

This end point is used to display all information about the given stock within the date range provided. It is only available to users who are logged in.

/user/register

This endpoint is used in order for users to register a new account in order to view more detailed information about a stock.

/user/login

This end point is used for users that have already created an account to login in order to view more detailed information about a stock.

Modules Used

ag-Grid-react

This module allows for fully featured tables, with built-in filtering, sorting, and features to handle cell/row selection.
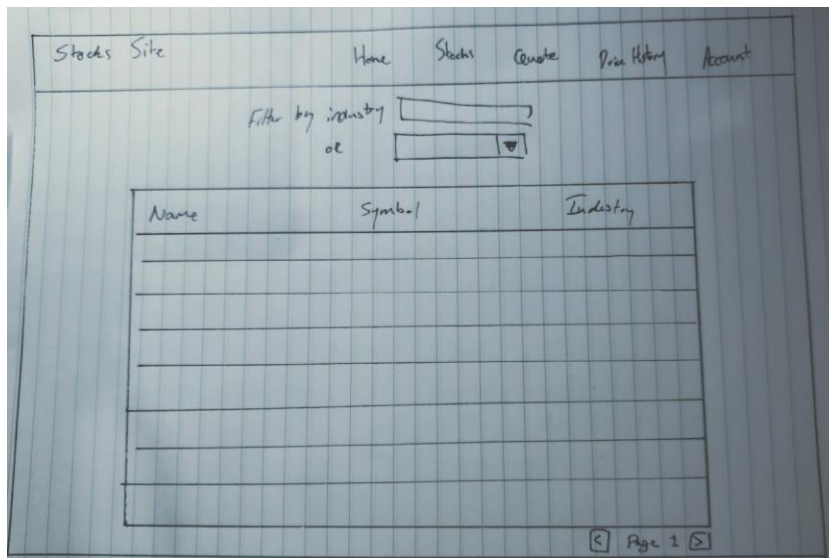
https://www.ag-grid.com/react-grid/

reactstrap

This module allows for the easy implementation of Bootstrap 4 components into a react webapp, without the need to use jQuery.

https://reactstrap.github.io/

Application Design

Navigation & Layout

The structure of the web app follows that of the layout of a fairly standard web page with a navigation bar spanning the width of the page at the very top with links to the various pages of the application. This structure was chosen as it will be most familiar to most users, so that they will not experience as learning curve when they first begin to use the application.

Technical Description

Architecture

The image below (Figure 1) shows the `src` directory of the web application. Upon running the web app, the file `index.js` is called, which then renders the file `App.js`.

components	12/05/2020 10:39 AM	File folder	
pages	12/05/2020 10:39 AM	File folder	
App.css	13/05/2020 12:27 AM	Cascading Style S...	1 KB
App.js	12/05/2020 10:39 AM	JavaScript File	2 KB
index.js	14/04/2020 10:54 PM	JavaScript File	1 KB

Figure 1: ./src directory

`App.js` handles all of the react routing for the application, so it renders the `.jsx` files in the `pages` directory depending on the url.

Inside the `pages` directory (Figure 2) are all of the `.jsx` files. Each of these is for an individual page of the web application.

Home.jsx	13/05/2020 4:04 PM	JSX File	1 KB
Login.jsx	13/05/2020 3:11 PM	JSX File	3 KB
Logout.jsx	13/05/2020 3:35 PM	JSX File	1 KB
PriceHistory.jsx	13/05/2020 4:15 PM	JSX File	5 KB
Register.jsx	13/05/2020 5:04 PM	JSX File	3 KB
Stocks.jsx	13/05/2020 4:35 PM	JSX File	4 KB

Figure 2: ./src/pages directory

Inside the `components` directory (Figure 3), are the `.jsx` files for any custom components used by the web application. The only component used at this the navigation bar at the top of the application (there are logged in and logged out variants of this component).

HeaderIN.jsx	13/05/2020 3:36 PM	JSX File	1 KB	
HeaderOUT.jsx	13/05/2020 2:31 PM	JSX File	2 KB	

Figure 3: ./src/components directory

Test Plan

Task	Expected Outcome	Actual Outcome	Screenshot (Appendix A)
View stocks	All stocks should be displayed	All stocks are displayed	A - 1
Filter stocks by partial industry	Stocks of industries containing the entered string should be shown	Stocks of industries containing the entered string are displayed	A - 2
Filter stocks by full industry name	Stocks of the entered industry should be displayed	Stocks of the entered industry are displayed	A - 3
Login	User should be able to login to an existing account	User can login to their account	-
Register	User should be able to register a new account	User can create a new account	-
Logout	User should be able to logout of there account	User can logout of their account	-
Filter by date	Authed users should be able to filter stock data by date	Authed users can filter stocks by date	A - 4
Display chart	Authed users should be able to see a chart of close values for stock data	Chart functionality not implemented	-

Difficulties / Exclusions / Unresolved & Persistent Errors

I faced a number of difficulties throughout the development of this web application. The first of these issues was that I would get errors when trying to populate the table with data from the API when the query to the API returned only one result. The issue was caused by the fact that I used the `map` function to populate the data, but it only works with an array, so single values had to be put into an array for `map` to work.

The second major issue that I encountered was with elements that would have to change depending on whether or not the user was logged in. The issue was caused because the components were not being re-rendered after logging in. This was fixed by forcing a page reload after logging in or out.

The only issue that I believe to remain in the web application at this time is related to error handling. In some cases, when there is an error, a single error message is displayed no matter what the actual error is.

The only feature that has not being implemented is charting for authenticated stock data, which I would like to have implemented using chartJS.

User Guide

To filter stocks by industry: go to *Stocks* page, enter an industry name and click *Apply filter*

To see more information about a stock: click on it from the list of stocks from the *Stocks* page (you will see more information if you are logged in)

OR (if authenticated)

You can also filter by date from here by clicking the date selectors, choosing start and end dates from the popups, then clicking apply.

To login or register: go to the login / register screen and enter your credentials.

To logout, click the logout button in the navigation bar, then the logout button again on the confirmation screen.

References

ag-Grid (n.d.). *ag-Grid Documentation.* https://www.ag-grid.com/documentation-main/documentation.php

Reactstrap (n.d.). *reactstrap components.* https://reactstrap.github.io/components/

Appendix A

Appendix A - 1

Appendix A – 2

Appendix A - 3

Filter by industry:	Health care	Apply Filter	
Name	Symbol		Industry
Agilent Technologies Inc	A		Health Care
AbbVie Inc.	ABBV		Health Care
AmerisourceBergen Corp	ABC		Health Care
Abbott Laboratories	ABT		Health Care
Aetna Inc	AET		Health Care
Allergan Plc	AGN		Health Care
Align Technology	ALGN		Health Care
Alexion Pharmaceuticals	ALXN		Health Care
Amgen Inc	AMGN		Health Care
Anthem Inc.	ANTM		Health Care
Baxter International Inc.	BAX		Health Care
Becton Dickinson	BDX		Health Care
Biogen Inc.	BIIB		Health Care
Bristol-Myers Squibb	BMY		Health Care
Boston Scientific	BSX		Health Care

1 to 15 of 61 |<   <   Page 1 of 5   >   >|

Appendix A – 4

ABC

From: 02 / 02 / 2020 ⊗ to: 04 / 03 / 2020 ⊗ Apply

Date	Open	Close	Low	High	Volumes
2020-03-04	87.42	85.87	84.45	88.95	2190380
2020-03-03	84.53	87.16	83.89	88.32	4527950
2020-03-02	83.55	84.32	81.47	85	3827500
2020-02-28	92.05	86.59	86.56	92.94	2566050
2020-02-27	93.8	93.71	93.1439	95.0051	1907070
2020-02-26	95.36	92.93	92.78	95.77	2239370
2020-02-25	94.72	94.93	94	96.04	1541700
2020-02-24	96.15	97	95.3	97.39	1213730
2020-02-21	95	96.53	93.95	96.99	1808100
2020-02-20	94.59	95.13	93.84	95.2	915843
2020-02-19	94.29	94.31	93.14	94.89	1566000
2020-02-18	94.16	94.36	90.23	94.74	1602250
2020-02-17	94.92	94.52	94.42	95.48	1155550
2020-02-14	95	95.78	94.04	97.5	1315140
2020-02-13	94.33	94.57	92.72	94.61	1156070
2020-02-12	91.7	93.6	91.355	94.18	1371620