

Daily meeting 1: Day 2

Yesterday: On day 1 I really wanted to be able to successfully log in to Yahoo Finance and navigate to the watchlist. I also wanted to be able to display the stock data to the console. I was able to meet my goal. In addition, I also created my database Schema.

Goals for Day 2

Be able to take the scraped data and insert into the database. I need to figure out how to store multiple scrapes in the database.

Accomplished

Switched to scraping data from columns rather than by row. I think this will be better for parsing. Started parsing the different elements of the stock data.

Daily Meeting 2: Day 3

Yesterday: I was hoping to have been further along in parsing the stock data. I made the decision to scrape data from the column vs. row.

I spent a decent chunk of time trying to do so. I ended up using XPath to grab each column's aria-label. I created arrays for each stock element's parsed data.

I may refactor later and switch to lists, to be determined.

Goals for Day 3

I want to finish parsing all stock elements. ALSO, I may create the front-end ASP.NET file for the project based upon somebody's rec. Time dependent → insert data into StockData database.

Accomplished

Created the front end file. Parsed all data from the stocks. I may need to change the data types, tbd.

Daily Meeting 3: Day 4

Yesterday: I finished parsing the stock elements. I created arrays that would store the parsed data. All stock symbols are in 1 array, lastPrice data is in another array, etc.

Issues: I spent a good amount of time trying to figure out why a couple of the stock elements wouldn't parse. Turns out, the `cw` statement I was using, referenced the wrong data!

Goals for Day 4

My main focus today is to research how to insert data into my database dynamically. I'm debating whether or not to create a dictionary that will store all instances of the Stock class. If I immediately store the stock data in the database, I don't really see why I would need to create the dictionary...

Accomplished

I looked into ways to insert my stock data. Wrote some SQL commands to try to store data. Not quite there.

Daily Meeting 4: Day 5

Yesterday: Most of my time spent was on Google trying to figure out how to insert my stocks into my database. Added a little bit of SqlCommand's to experiment with.

Goal for the Day

To get my stock data into the database.

Accomplished

Nothing concrete. I played around with my SqlCommands some more. Unfortunately, inserting my data into the database is still a work in progress..

Daily Meeting 5: Day 6

Yesterday: Mostly spent time researching inserting data into database. Played with my code hoping to make some progress.

Goal for today

Same as yesterday \rightarrow insert data into database.

Accomplished

Yes! Finally inserted the data!
At first I didn't make my primary key auto-increment (I honestly don't know why!), so I changed it to AI.
I ran my program so much, the Id's were in the low 1000's. I have to do something about this \rightarrow reset Auto incrementer for testing purposes. I created a method that clears the data from the table. and a method that reseeds the primary id.

Daily Meeting 6: Day 7

Yesterday: Now that I have my data in the table, I need to be able to store multiple scrapes. At the same time, I need to be able to display the latest scrape once I get the front end up and running.

Goals for the day:

I need to make a second table that will be used to store multiple scrapes. My existing table will be renamed to LatestScrape. I think I need to create a second SQL command object that will also insert the data into the tables.

Accomplished

- Created a second table to store just the most recently scraped data.
- Successfully inserted data into new table. Tried to delete old data but then only the last stocks data is entered.

Daily meeting 7: Day 8

Yesterday: created my second table. Attempted to keep only newest data by deleting old data.

Goals for the day

I need to be able to store newly scraped data into the latestScrape table. I will switch gears; update the data instead of deleting it.

I need to change the primary key to the stocks symbol, because the same stocks will be entered every time. If I can get the data to update, I really would like to start the front end phase.

Accomplished

My latestScrape table is successfully updated with each scrape. I did not get to tackle the front end - def. need to do research before anything.

Daily meeting 8: Day 9

yesterday: finished code to update the table that will hold most current stock data.

Issues: Had some trouble figuring out what SQL commands were needed to update a table.

Goals for the day

Do lots of research regarding the ASP.NET aspect of the project

Accomplished

No coding. Just reading and watching of materials.

Daily Meeting 9: Day 16

Yesterday: Research

Research
Research

Goals for the day

- Start front end
- connect database if possible

Accomplished

This was the most frustrating day so far. Things were constantly going wrong. The first ms docs I used, I couldn't even run the starter project due to IIS Express issues. I deleted and recreated a few different projects. Eventually, I created a project and even got my database connected. There's a lot of generated code and I'm not sure how I feel about that.

Daily Meeting 10: Day 11

Yesterday: Created front end. Connected database to it. Some initial issue w/ IIS Express but creating a whole new project fixed this.

Goals for the Day

Add a button to run a Scraper, possibly start the API aspect of project. I don't want to worry about changing the look of the site until the API/RestSharp parts are integrated.

Accomplished

I added authorize tags to the needed controllers. I deleted a bunch of pregenerated code that I didn't need. I created a new ActionResult in the Current Stocks controller that would run the scraper when clicked. I brought in the Selenium Scraper project by creating a new ScraperService folder and bringing in the code.

Daily Meeting 11: Day 12

Yesterday? Was a good day! I was able to successfully add a newsrape button to update the current stocks.

Goals for the Day

Work on my HTML Agility Pack scraper. I hope to get a good amount done today, we'll see what happens.

Accomplished

Created my HAP project and the stock class for the data I will scrape. I got the initial code set up for my HTMLWeb object.

I played around with XPath and tried grabbing the stock data from Google finance stocklist.

Daily meeting :12: Day 13

Yesterday: No luck scraping data from the Google Finance page yet.

Goals For the Day

Scrape data using HTML Agility pack. Today may be a research kind of day. I may switch stocks site because there's something about the Google Finance page that makes it unscrapable.

Achieved

I switched sites to scrape from. I can see that I'm able to access the table itself, but not any of the data contained inside of it.

Daily Meeting 13: Day 14

Yesterday: I'm having a lot of trouble extracting data from the table.

Goals for the Day

Scrape the table. Do more research and find some good examples of HAF in action.

Accomplished

Experimented with various XPaths. The dev tools in my browser has been really helpful making sure I am using the correct XPath.

Daily Meeting 14: Day 15

Yesterday: No progress w/ HAP code
Did Research

Goals for the Day

Somebody in class helped me realize why I couldn't scrape the Edward Jones stock list. The site doesn't require logging in, the stock list I put together was being stored in my cache. It's good to know it wasn't my code.

- Find new site to Scrape

Accomplished

I picked the Nasdaq site as my source of stock info. They have watch lists that like the 'most active' list and that is where I will extract the stock data from.

Daily Meeting 15: Day 16

Yesterday: I realized my mistake w/ trying to scrape data from the Edward Jones site. I decided to use Nasdaq.com instead, their most active stock list.

Goals for the Day

Finally scrape data using XAP

Accomplished

Using XPath, I created a `tableNode` variable and made the table containing the stock data as my root node.

Using a foreach loop, I traveled through the table nodes and extracted the inner text, storing the data in appropriately named variables. I then created Stock objects for each stock scraped from the site.

Daily meeting 16: Day 17

Yesterday: Scrapped data from the Nasdaq website using HTML Agility - Pack

Goals for the Day

Add my Nasdaq info to the database.
Hopefully I can start the RESTsharp portion today.

Accomplished

I did get the Nasdaq data into my database. To parse the 'change percent' info, I had to use Regex which was nice. I found an API to use, World Trading Data. I created some classes - ApiCall to model the JSON response I would be getting. I played around w/ making an API call.

Daily meeting 17: Day 18

Yesterday: Set up my RESTSharp project

Goals for the Day

Finish up w/ the RESTsharp API,
look into the front end aspect.

Accomplished

I was able to grab individual pieces of data from the JSON response object. I deleted a bunch of fields in my ApiCall class. I only need a few pieces of data from the API, not all of them. Created the frontend project, an ASP.NET mvc core file. I also successfully entered my API stock data into my database.

* I brought in my Selenium Scraper files and created my model/controller for it.

Daily Meeting 18: Day 19

Yesterday: Cleaned up unnecessary code
RestSharp project. Created the frontend
project; brought my scraper code into it.

Goals for the Day

Even though I created the basic
front end, I didn't do a whole
bunch. I want to continue working
on that. Add my other scrapers to
it.

Accomplished

I deleted some unnecessary
fields from my Selenium scraper files. I
added the H&P stock model - Nasdaq
stocks. Once my Nasdaq model was
done, I created the controller for
it. I refactored my Database.cs
file in the ScraperService folder - when
it came to creating the SQL parameters,
I had more code than needed.

I also created a view that has
links for viewing each set of stock
data.

Daily meeting 19: Day 26

Yesterday: Played w/ my ScraperService files and added the Nasdaq model and controller. It was a good day!

Goals for Today

Bring in my RESTSharp scraper, get all stock data to display on the frontend.

Accomplished

I added a model and controller for the stocks pulled with RESTSharp, with World Trading Data API. I also created a RestSharpScraperService folder to store the code needed to run a new scrape. Played around w/ paginating my stock history pages.

Daily Meeting 20: Day 21

Yesterday: Got all stock data to display on the front end.

Goals for the Day

Add some buttons! I need buttons to perform a new scrape for each of the scrapers and to clear the stock history tables.

Accomplished

I now have buttons to perform new scrapes for all three of the stock scrapers. And! I have buttons that clear the history tables and then redirects you to the current stocks page. All that's left is tweaks and messing w/ the CSS.