# Guidelines for ETL Project

Shared code for project:

<https://colab.research.google.com/drive/1j88LIGZgA-6VPalcvm3rcKicFq0iKqmu>

## Finding Data

Your project must use 2 or more sources of data. We recommend the following sites to use as sources of data:

Data Source 1

<https://thestockmarketwatch.com/markets/today.aspx>

Data Source 2:

<https://finnhub.io/docs/api>

Data Source 1 was used to grab the biggest losers and winners that dictated what data we gathered from FinnHubb.

The goal of the project will be to gather this data and let that drive the stock trading script

Data Source 2 was used to grab the actual stock data.  We showed the example of news but there is much more information on the internet.  We plan to further implement this data to add historical stock data and balance sheet data.

FinnHub

URL of page to be scraped

url = 'https://thestockmarketwatch.com/markets/after-hours/trading.aspx'

# Retrieve page with the requests module

response = requests.get(url)

# Create BeautifulSoup object; parse with 'lxml'

soup = BeautifulSoup(response.text, 'lxml')

# Make a df with the gainers. NB: This is the 2nd table ‘[1]’

gainers = pd.read\_html(soup.prettify())[1]

# Check the table.

gainers.head()

# Make a df with the losers. NB: This is the 3rd table ‘[2]’

losers = pd.read\_html(soup.prettify())[2]

# Check the table.

losers.head()

## Data Cleanup & Analysis

Once you have identified your datasets, perform ETL on the data. Make sure to plan and document the following:

\* The sources of data that you will extract from.

1. FinnHubb
2. The stock marketwatch.com

\* The type of transformation needed for this data (cleaning, joining, filtering, aggregating, etc).

1. Python to scrape and clean
2. SQL to join filter aggregate etc
3. Pandas to built data frames to insert into SQL
4. JSON for getting the web scraping informationis pretty printed for fitting into a dataframe

\* The type of final production database to load the data into (relational or non-relational).

1. SQL

\* The final tables or collections that will be used in the production database.

1. Stock data and its guided from the biggest market movers
2. Daily top  losers
3. Daily top winners
4. Results of news from top losers (buy sell rating, how many analysis are following, etc)
5. Results of top winners (buy rating, sell rating, how many analysis are following etc)

You will be required to submit a final technical report with the above information and steps required to reproduce your ETL process.

## Project Report

At the end of the week, your team will submit a Final Report that describes the following:

Data was formatted using try pandas, and a for loop with try excepts.  We ignored when data was not present as the beginning.  The issue with top losers are sometimes the losing stocks are penny stocks, and thus no analyst follow the stocks.  So, basically we skipped pulling that data when it was not present.

\* \*\*E\*\*xtract: your original data sources and how the data was formatted (CSV, JSON, pgAdmin 4, etc).

\* \*\*T\*\*ransform: what data cleaning or transformation was required.

Formatting is very important for SQL.  We had to make sure that the SQL schema was correct, i.e. when the company name was 500 in the first pull and not 5000 in others the schema had to correctly be fitted to fit our data, or have an try else function.  We did not want to have the line of code that always replaced the data becausewe want to capture and record the data across time

\* \*\*L\*\*oad: the final database, tables/collections, and why this was chosen.

We chose to use SQL because our table was gathered with the idea that it will always be the same.  For example our ticker symbol is always going to be text, never numbers.  Therefore, because our data was mainly in rows not columns and was very defined we used MsSQL.  We chose MsSQL because it is robust and large enough for us to handle the information

Please upload the report to Github and submit a link to Bootcampspot.

- - -

### Copyright

Coding Boot Camp © 2019. All Rights Reserved.