Wei-Yin Ko

FinDocAnalyzer v.0.007

USER MANUAL

HELLO THERE!

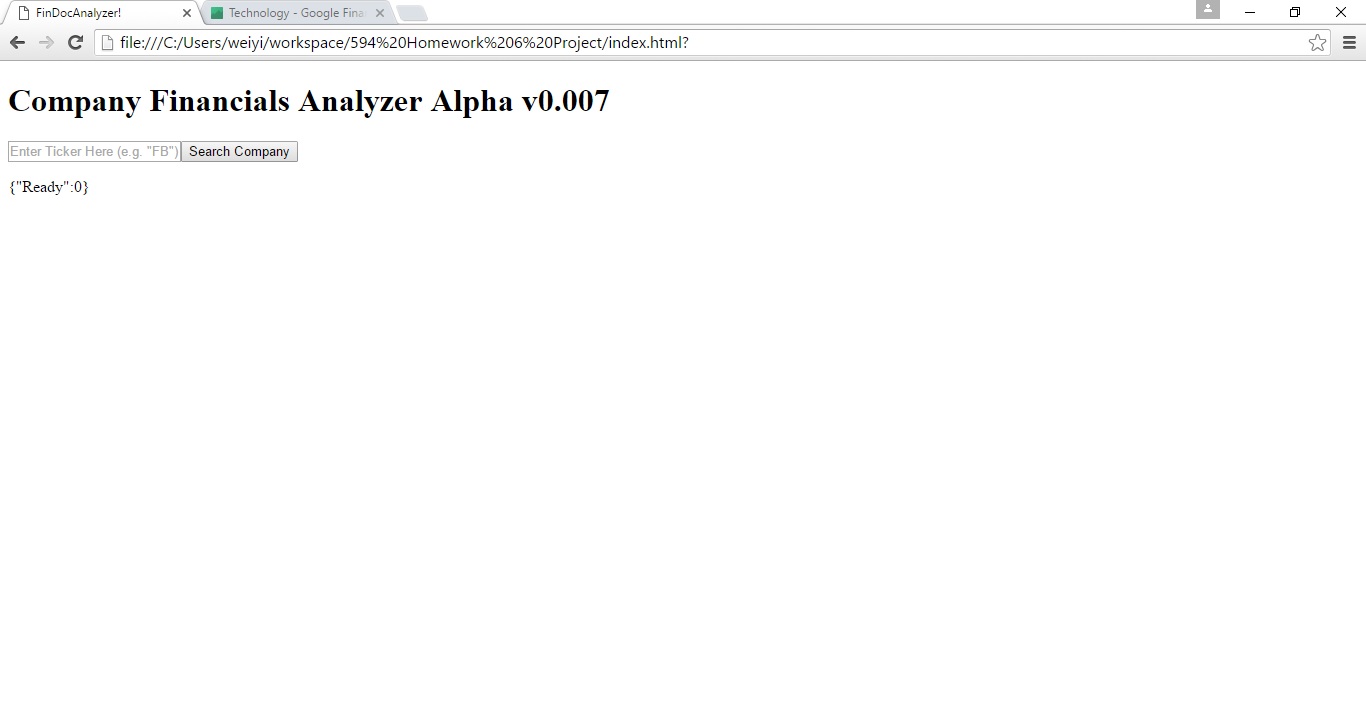
Thanks for trying out the FinDocAnalyzer—the tool for getting quick valuation with one button click! There are only three steps required for you to get the information you want.

Before we move on…:

Assuming that you are running the application locally, first you need to download Node.js [here](https://nodejs.org/en/download/). Once that is done, use the command prompt to navigate to the project folder and type **npm install** to install the needed modules and more (because I accidently imported them from another project…)! Once that is done, type in **node server.js** to kickstart the server and run FinDocAnalyzer in Java to start the program!

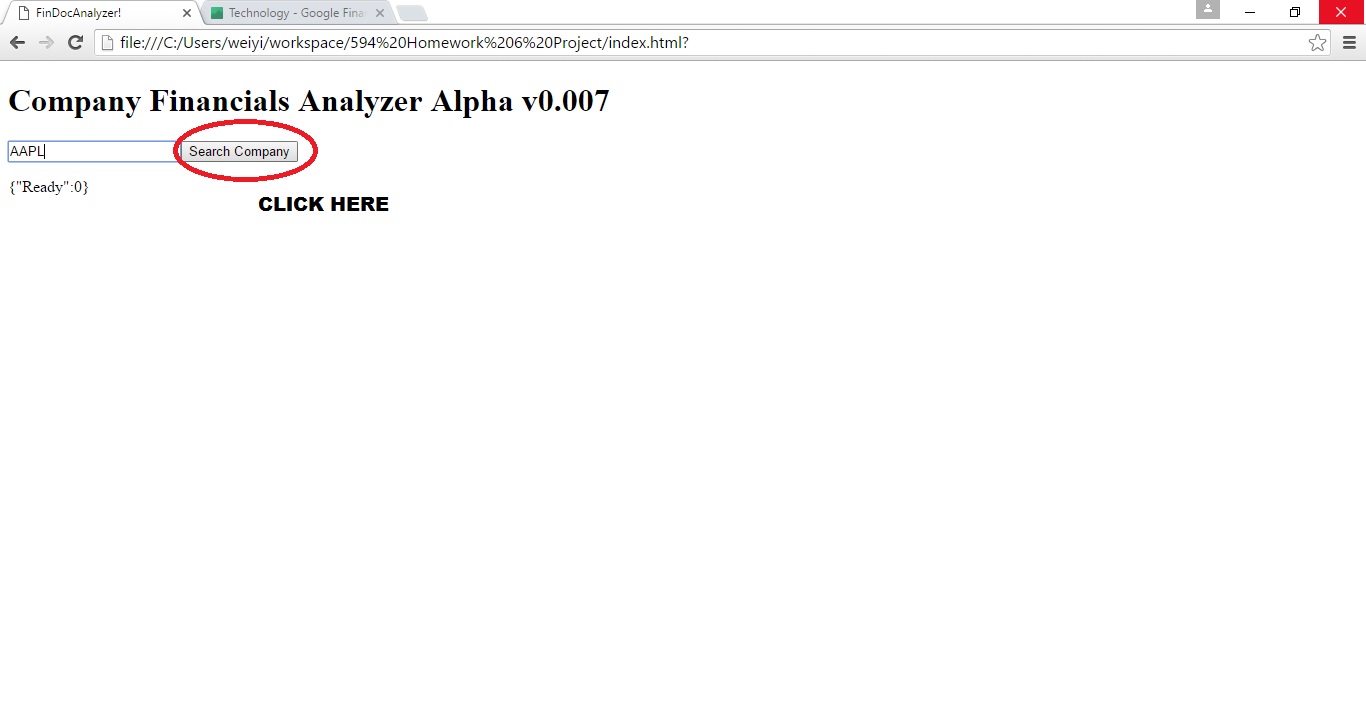
Step 1:

Double-click on **index.html**. You should see this upon startup:



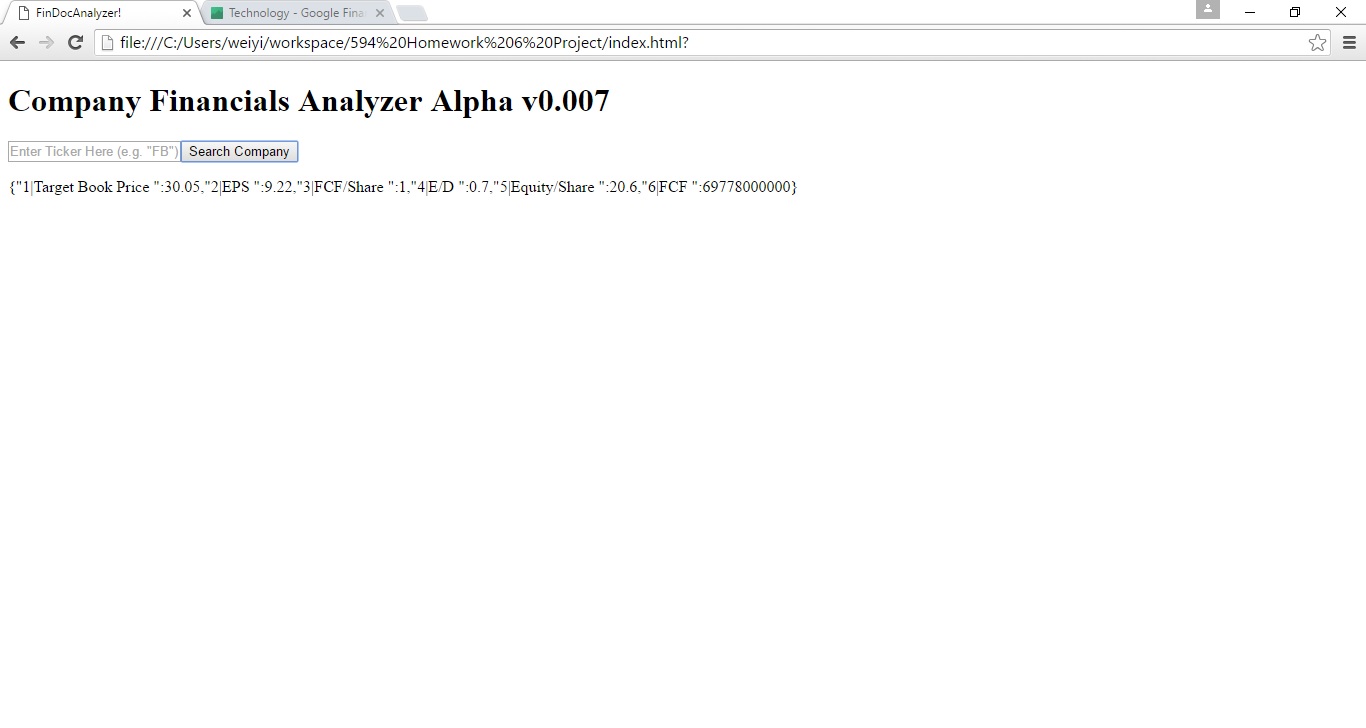
Step 2:

Type in a ticker symbol, in this case I typed in Apple’s, which is AAPL. Once that is done press the button!



Step 3:

Wait for a few seconds and either some calculated valuation metrics should come up (or some error message due to the many, many limitations of the APIs)! That’s all there is to it! Feel free to search for a different company and see what happens—but not too frequently since you can only make 5 APIs a minute!



UHHHH… WHAT AM I LOOKING AT?

That’s a great question! These metrics are some of the most important indicators of a company’s health. Here are some explanations for each metric:

**Target Book Price:** the estimated Book Value (equity over shares) for the upcoming year. Since historical stock prices or industry multiples were somehow not available in any of the calls this is the best we can do. The target is calculated by comparing the most recent periods’ EPS with an average of its last five years’. This projected growth (or shrinkage) is compounded into the Equity over Shares multiple. This shows what, without market forces, the company should be valuated at based on its earnings. Of course, in the actual market the stock prices won’t reflect this value at all, especially for tech companies with often unhealthy fundamentals!

**EPS (Earnings per Share):** This shows the ratio between the company’s income versus its number of shares. This is often used as a “hype” indicator as companies with EPS that are too low when their stock prices are high or growing shows that the market may be being over-optimistic about these firms.

**Equity/Share:** Equity over share shows the “book value” of how much stocks are worth on paper without the impact of market forces. It demonstrates the fundamental value of the company. It can also be used to evaluate industry multiples.

**FCF:** Free Cash Flow—the most basic measure of fundamental analysis. It is the amount of cash leftover after subtracting the amount required to maintain the company’s current assets. This is an important measure because FCF can be used to expand the company’s business—thereby providing its shareholders even more value! It is an indicator of the company’s potential.

**FCF/Share:** Free Cash Flow over Shares shows what potential value the company may have for its shareholders. It’s importance/value differs wildly by industry

**D/E :** Debt over Equity evaluates the composition of the company’s Asset and is a common indicator of how healthy the company’s operation is. Also the “standard” D/E is industry-specific, it is never a good thing to see a D/E that is too high as incurring too much debt is often a very bad sign of a company’s operations. With that being said, if a company’s D/E is very low it might signal a fundraising opportunity by issuing more bonds!

The main takeaway is that for someone interested in equity analysis, these metrics paint a fairly holistic picture of the company for its shareholders. Combined with industry ratios, they provide great insight as to how a particular company is doing relative to its peers and whether they represent a good investment opportunity or not!

CLASSES/MISC

index.html: The user interface which provides the search function for tickers and displays the data returned. It also issues POST and GET requests to the server.

server.js: A simple server with POST and GET functions that provides handles requests from the html page and uses apicalls to execute them.

apicalls.js: a collection of functions for making the API calls to dev.last10k.com and also provides communication with the NodeCommunicator.

FinDocAnalyzer: This class provides a main method for running the NodeCommunicator.

FinDocFactory: The abstract superclass for the financial document parser factories. It has the factory method.

BalanceFactory: The factory class for Balance Sheet Parsers.

CashFlowFactory: The factory class for Cash Flow Statement Parsers.

IncomeFactory: The factory class for generating Income Statement parsers.

RatioFactory: The factory class for Key Ratio Parsers.

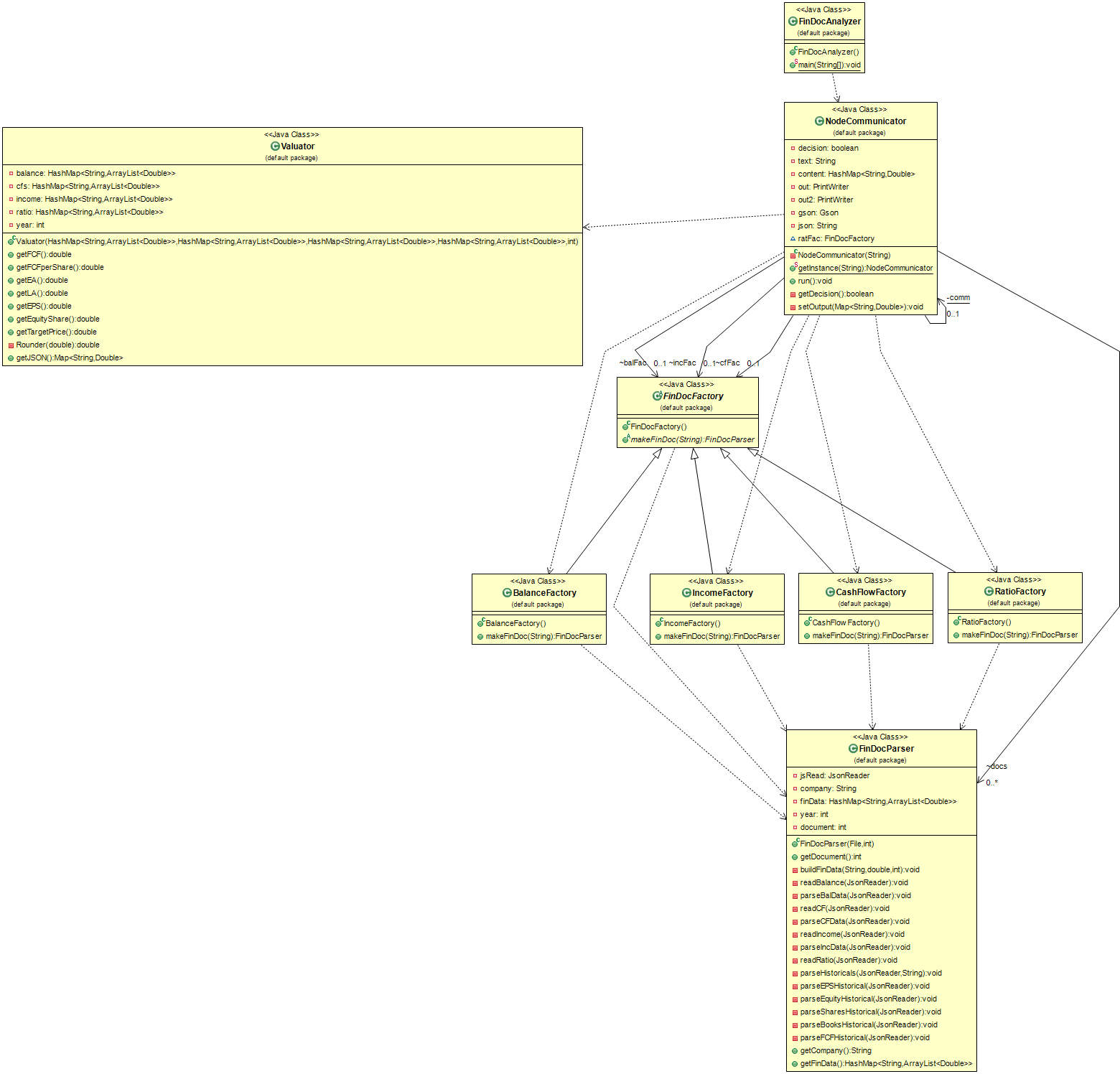
FinDocParser: This class parses the financial document JSONs. In this implementation the parser can handle Balance Sheets, Cash Flow Statement, Income Statement, and Key Financial Ratios.

NodeCommunicator: The NodeCommunicator class runs in intervals to communicate with the Node.js codes. It waits for a start message and periodically utilizes the FinDocParser and Valuator to interpret the JSONs and send a JSON back to Node. It utilizes the Singleton pattern.

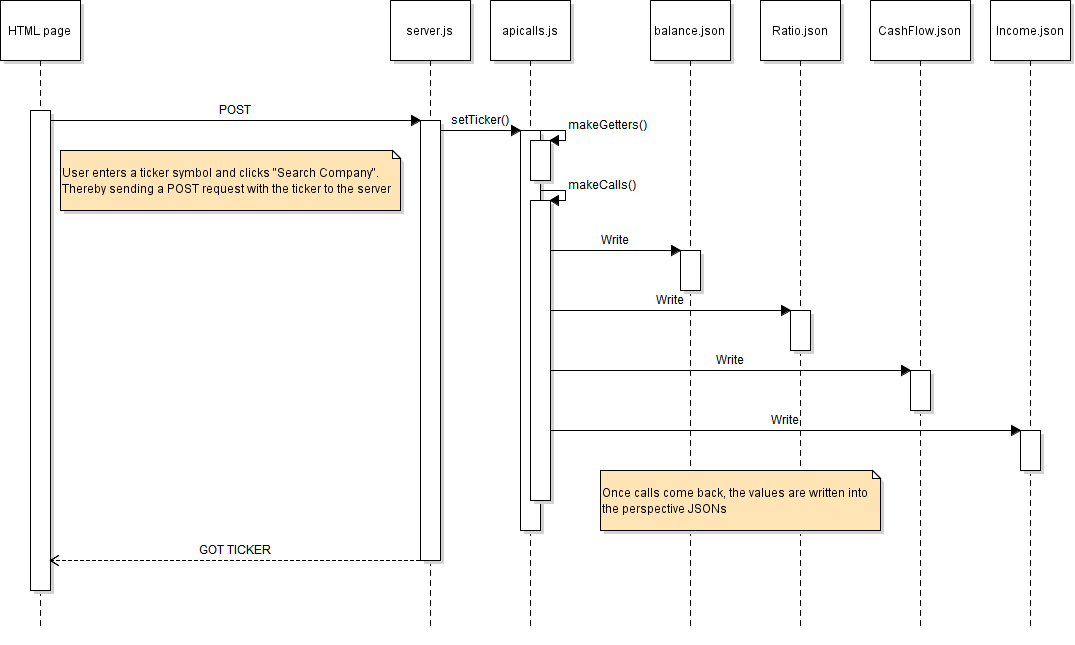
Valuator: This class gets the financial data and calculates the various important ratios along with the target book price.

CLASS/SEQUENCE DIAGRAMS

[Note: Class Diagram only available to java portion since my JS codes do not follow OOP design]



Making API calls:



Getting Valuation Metrics (very wide image)

