Creating and implementing a Security Class

According to the Assignment 2 Brief 4.4 the current system represents a security in terms of a ticker symbol only which means that no extra information are provided for the user. Therefore, a possibility of executing more complex operations is not available. Hence the need for creating such a class arises.

The security class will contain four attributes: symbol, name, sector, industry. The usability of the mentioned attributes will be allowed through their getters and setters.

There is also need to implement a method which would return a current market value.

Known facts about security: - listed on the market under a symbol

* volume of the security for a particular time period is an amount of shares traded for that period
* open – price at which the period opened
* close – price at which the period closed
* highest – the highest price for the period
* lowest – the lowest price for the period

Current market price should be the closing price for a last recorded period.

**getCurrentMarketValue() implementation**

How it is done

MarketDataServer class requires configuration data for reading from it in order to:

1. create self.path = url, self.symbol = security symbol, self.interval, self.function, self.datatype, self.apikey
2. getURL for querying the data for a particular symbol, in particular interval returning in a particular data type(JSON)

The object from the MarketDataServer class characterizes the market data source which can be in two forms : JSON by calling getDataAsJSON() or TXT when calling getDataAsText().

So in order to implement the method we need to:

1. create a variable, which holds the pointer to the PriceServer object, called self.price\_srvr
2. create the PriceServer object where we pass Alphavatage object, as a parameter which takes another parameter of type String (getConfigFileName)
3. now we can call getLastRecodedPriceBySymbol method on PriceServer object returning the last price as a String