Test instructions

The following are the instructions for completing the test. ***Please read through them all before starting.***

For this project you will be using an open source Stocks feed API to provide incoming data. This API is open source and requires no authentication. The documentation (should you need to look at it) is here: https://iexcloud.io/docs/api/

You must create a Java Spring Boot project which has the following properties:

- A **non-web** based Spring Boot application - which outputs messages to the console / terminal window.
- On load – query the following stocks API: https://sandbox.iexapis.com/stable/ref-data/symbols?token=Tpk_ee567917a6b640bb8602834c9d30e571
- This gives the symbols/names/state for each trading company.
- Reference: [https://iexcloud.io/docs/api/#symbols]
- Loop through each company that is enabled and put a request onto a queue for that company to get its data.
- Have multiple threads processing the queue - downloading the current stock information for that company, using the following API: https://sandbox.iexapis.com/stable/stock/{stock code}/quote?token={token}
- i.e. https://sandbox.iexapis.com/stable/stock/MSFT/quote?token=Tpk_ee567917a6b640bb8602834c9d30e571
- Reference: https://iexcloud.io/docs/api/#quote
- Save the data from each stock quote into a DB structure that can be queried.
- The DB can be an In-Memory DB in the project or a Docker or local instance of MySQL / Postgres / Cassandra / MongoDB i.e. any data repository you prefer.
- The console window should output at regular intervals (say every 5 seconds) the following stats:
  - The top 5 highest value stocks (in order – largest first, then order by company name)
  - The most recent 5 companies that have the greatest change percent in stock value
- Once the process has completed (looped through all companies), it should start again and download the current stock information for each company again.

- **If the information has changed for a company, then this change needs to be recorded in the DB, i.e. a log for each company of the changes to their stocks over time. IMPORTANT**

Stretch goals

If you finish the above in good time and wish to continue, then the following are stretch goals - pick one (or many).

- Add logging - using slf4j in conjunction with either logback or log4j2
- Use lombok to remove as much boilerplate code as possible
- Perform the queries against the DB using Spark

Important!

Please to make sure you commit each individual change seperately with appropriate commit comment and push to the remote. This will ensure that the code reviewer can see the progress of work, and also ensure you can work in small discrete packets.