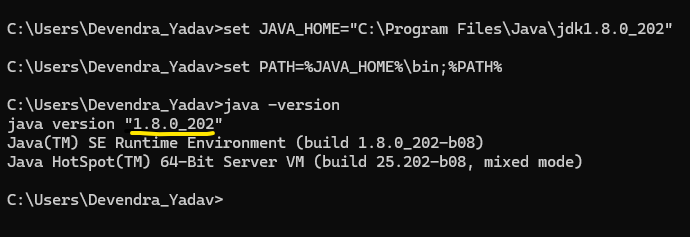
# **README : JAVA ASSIGNMENT**

## Environment Setup

1. First make sure the java version is 8 when you check on CMD.
2. If its not 8 then please set correct JAVA\_HOME and PATH as shown below.
3. 
4. **cryptoassignment.zip** file containing the source code.

**A black text on a white background

Description automatically generated**

* 1. cryptoassignment : this is the java source code.
  2. input : this is the dir containing the sample input csv files

1. Extract the zip file as desired location. Lets say we extracted to “**C:\assignment**” dir.
2. We have some params configured in application.properties for input dir where csv will be read from.
3. A black text with black lines

   Description automatically generated with medium confidence

**trader.positions.input.dir** : input csvs will be picked up from this location. It will be polled every 2 secs and data will be fetched and saved in database.

**trader.positions.input.processed.dir** : Once the csv is successfully processed. It will be moved to this processed dir.

**trader.positions.input.error.dir** : If there are some non csvs files or some issue in reading them then those files will be moved to this error dir.

1. Create these dirs. In case you want to use other path then please update application.properties accordingly.

## Package Building

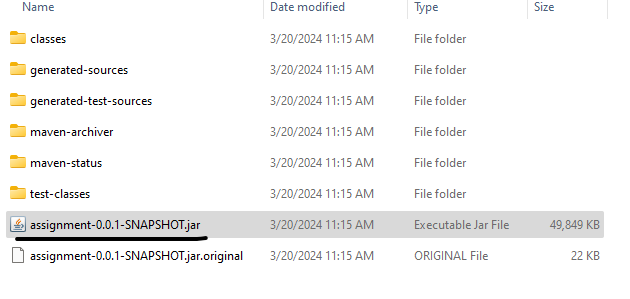
1. The assignment asked for gradle. But due to my current project that’s related to banking, we are not able to freely install new softwares in our machines. Hence I have used maven.
2. Please have maven installed.
3. Set M2\_HOME and MAVEN\_HOME environment variables pointing to maven dir.
4. Add $MAVEN\_HOME\bin to PATH environment variable.
5. Now you are in the desired location containing the project. Also maven and java 8 is installed.
6. To build the package jar execute following cmd.

 ‘**mvn clean package**’

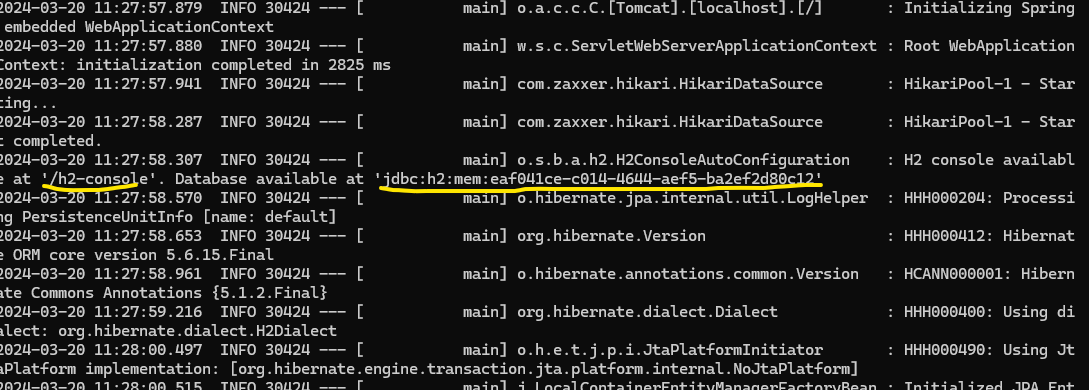
A black background with white text

Description automatically generated

1. This cmd will create a jar inside ‘target’ folder.



## Run project

1. Run cmd “**java -jar target\assignment-0.0.1-SNAPSHOT.jar crypto\_user1**”
   1. ‘crypto\_user1’ is the parameter that we are passing whose portfolio we want to see in output.
   2. 
   3. H2 console is enabled. Can login using the connection string seen in the logs as mentioned above.
   4. H2 console url : <http://localhost:8080/h2-console>
      1. User name : sa
      2. There is no password. Its blank
   5. A close-up of a computer code

      Description automatically generated
   6. You will see 3 tables created.
      1. **SECURITY**: This contains details about securities. Its prepopulated using **data.sql** that part of the project.



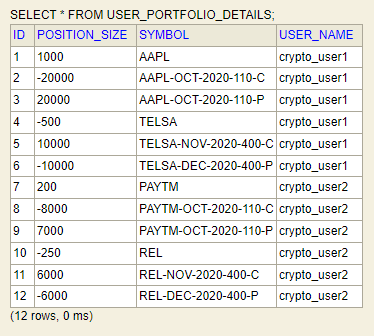
* + 1. **SECURITY\_VARIABLES** : this contains the predefined variables used to simulate the security prices. This also is prepopulate using data.sql that’s part of the project.

A screenshot of a computer

Description automatically generated

* + 1. **USER\_PORTFOLIO\_DETAILS** : this contains the user data that will be fetched by reading the input csv files.

We can have multiple csv input files for different users. Csv file name will be considered as username. Input csv will be read every 2 secs (defined in application.properties). Once processed, it will be moved to another dir.



* 1. Put the input csv files in input dir as mentioned in application.properties.

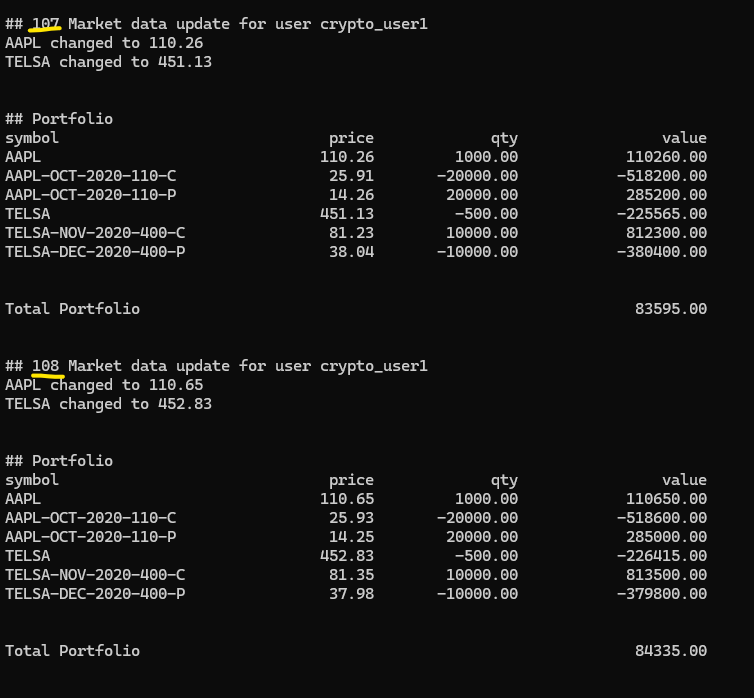


* + 1. Attached some sample input files as well. Can use them.
  1. In application.properties there are variables that define different intervals.

A computer code with black and white text

Description automatically generated

* 1. Market data will be refresh at every 2 secs and the new values of different security prices will be updated in database.
     1. You can see every 2 secs the prices will change in database (SECURITY table)
     2. 
     3. After some secs.
     4. A screen shot of a computer screen

        Description automatically generated
  2. In the console you can also see the portfolio will keep updating
  3. 
  4. You can see, its running for some time and the counter is increasing (marked in yellow).

## **Improvements**

1. Due to lack of time (sick) several things are missed in the project.
   1. Unit test cases
   2. Thinking about all the kind of scenarios possible. Positive/Negative.
   3. There will be several bugs present in the code. Fixing of those bugs.
   4. Instead of providing customer name as param 1 time at the beginning, subscriber to show portfolio could have been coded in a better way.
2. In assignment, the ask was for gradle, but I used maven.
   1. My current project is a banking project. So I cant freely install things in my machine.
   2. So instead of spending time to find a workaround/permission to get gradle, I just used maven that we use here.