Summary

* The main file for the project is “PriceBlenderApplication.java”
* Once application starts-up there is a scheduler configured to run every 1 minute (Configurable property) named as “MarketDataProcessor.java”
* This will read the files (comma separated text files) from configured location in ascending order of their last update, so that the older files get processed first. Convert it into java bean and persist if row is valid.
* The Invalid row (best > non-zero ask) will be logged in log file. Assuming if the ask is 0, then it might be representing the missing data. So it will be considered valid value.
* After processing each file, it will show the latest data for all 3 sources and calculate the best bid, mid and ask based on that snapshot. This output will be generated in datalogger.log file.
* PriceBlenderImpl is the implementation for given Interface priceBlender.
* getBestPrices() is an additional method in interface and implementation that will log the latest snapshot from database and the best bid mid and ask based on that.
* I have used h2 memory data base and shedlock for scheduling service. This will prevent the files to be picked by two instances at the same time. After processing the file will be moved to processed or exception folders along with timestamp.These folder must be created manually.
* Data.sql in resources is having the table structure for shedlock and marketdata.

Processing Logs

Text

Description automatically generated

Data Samples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data1.txt | Data2.txt | Invalid1.txt | Invalid2.txt | Data5.txt |
| marketDataSource,bid,ask  SOURCE\_A,19,21  SOURCE\_B,20,23  SOURCE\_C,18,22 | marketDataSource,bid,ask  SOURCE\_A,19,0 | marketDataSource,bid,ask  SOURCE\_A,0,0  SOURCE\_B,0,0  SOURCE\_C,0,0 | market DataSource,bid,ask  SOURCE\_A,50,21  SOURCE\_B,20,23  SOURCE\_C,18,22 | marketDataSource,bid,ask  SOURCE\_A,15,21  SOURCE\_B,13,16  SOURCE\_C,10,18 |

Output

Table

Description automatically generated