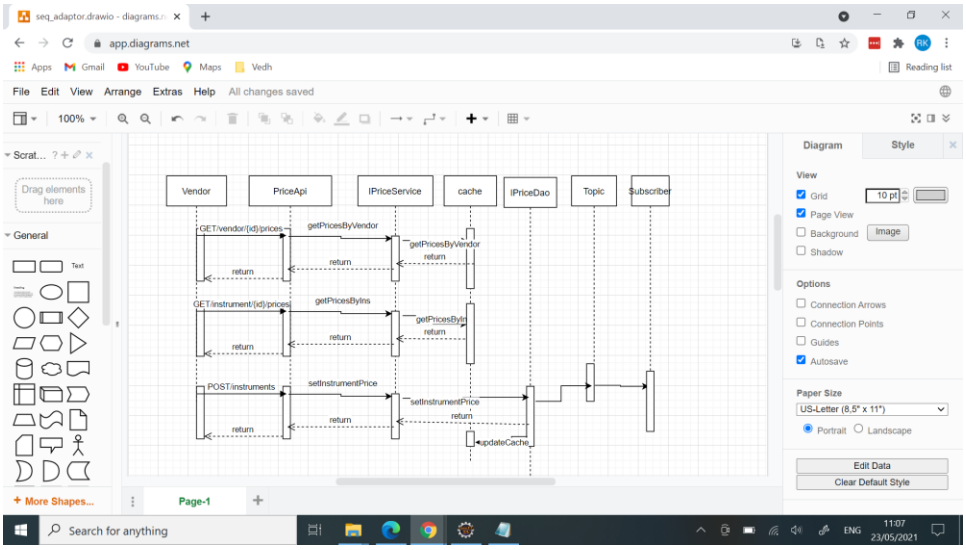
Price Retriever

The price retriever is component to store/update trade price instruments.

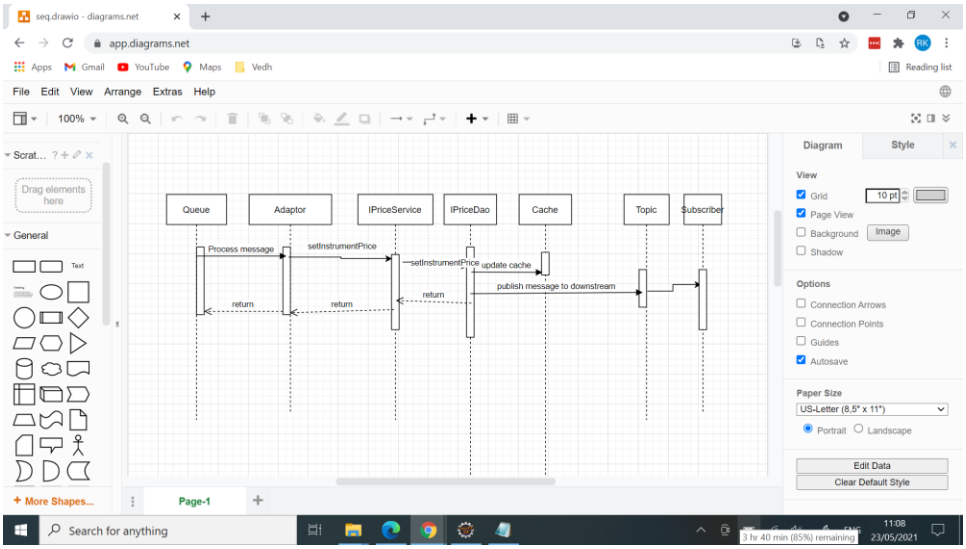
It uses in-memory cache to store the prices and publishes any changes/updates/new entrties to downstream subscribers. There are two ways to store the prices for the client one via REST api and another via sending messages to queue.

Sequence Diagram

The vendor price updates will be processed as shown below at high level.



Flow for messages to the Queue -->

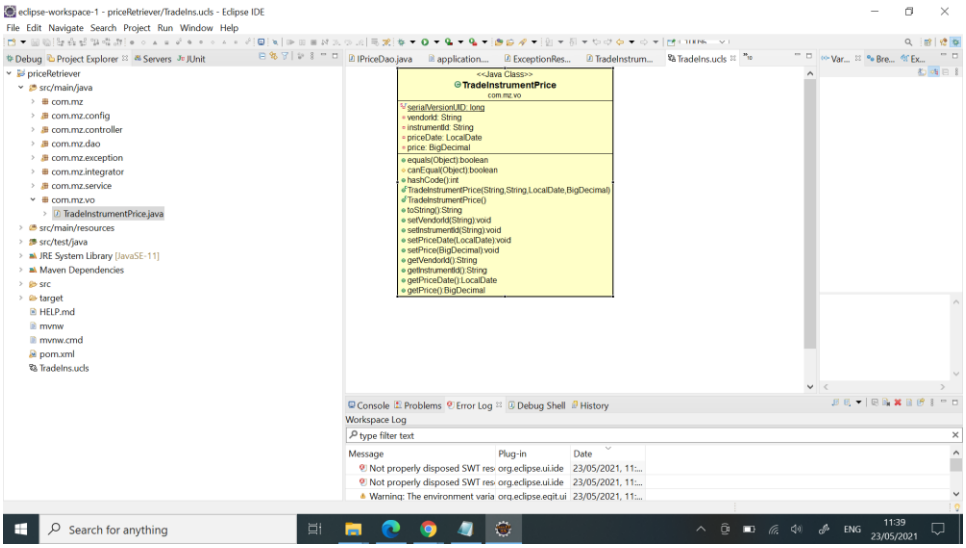


I have attached the sequence diagrams done in Draw.io in the zip file for reference.

Vendor Jms Integration Flow

This integration flow show case how to read vendor prices published in real time via a messaging channel. The solution uses Spring Integration (Java DSL) to implement to this flow.

Domain class -



Message format

JSON message format for vendor and clients.

Sample message

```
{"vendorId": "v1", "instrumentId": "i1", "price": “100”,"priceDate": "2021-05-21"}
```

Datastore -->

The solution currently uses in-memory maps stores the prices. This can be changed with different implementation of DAO if needed.

Cache

The solution uses Spring in-memory cache, is configured to delete the records older than 30 days.

This can be changed.

REST end point -->

- REST API Endpoint - http://localhost:8080
- Swagger plug-in can be added to get more documentation for API’s

Challenges faced -->

- Wasn't sure if multiple caches in spring can be used hence used catch resolver. Did some reading around documentation.
- Unit testing took some time due to an error of not using correct application class under spring boot.
- Context swithcing with family and task during off hours was little challenging.
- Tooling was a challenge – took a lot of time using draw io.