**Application Overview**

This application consists of two Micro services named as:

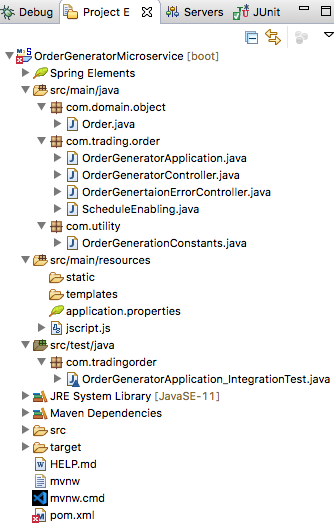
1. OrderGeneratorMicroservice
2. OrderProcessingMicroservice

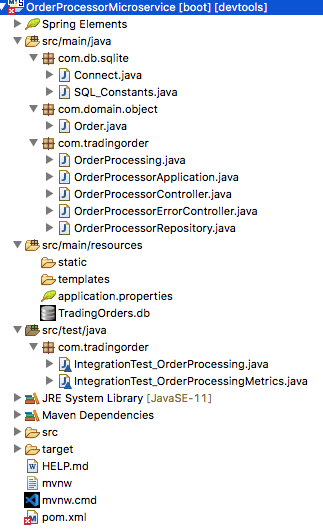
Both services communicate synchronously through REST. The processing service handles all the data received and after processing persists it in TradingOrders.db database. Integration test has been written using JUnit.

**Tools used**

* Maven 3.0+
* JDK 1.8+
* Spring Boot
* Docker
* GitHub
* SQLite

**Project Code Structure**



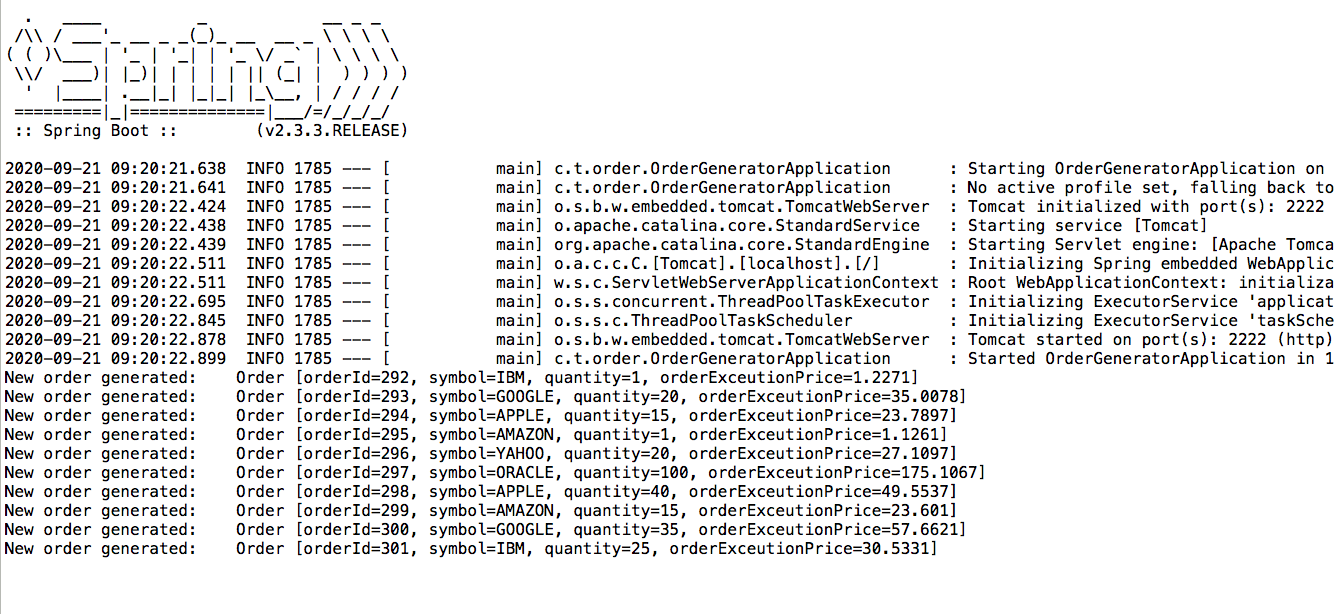


**How to run Application?**

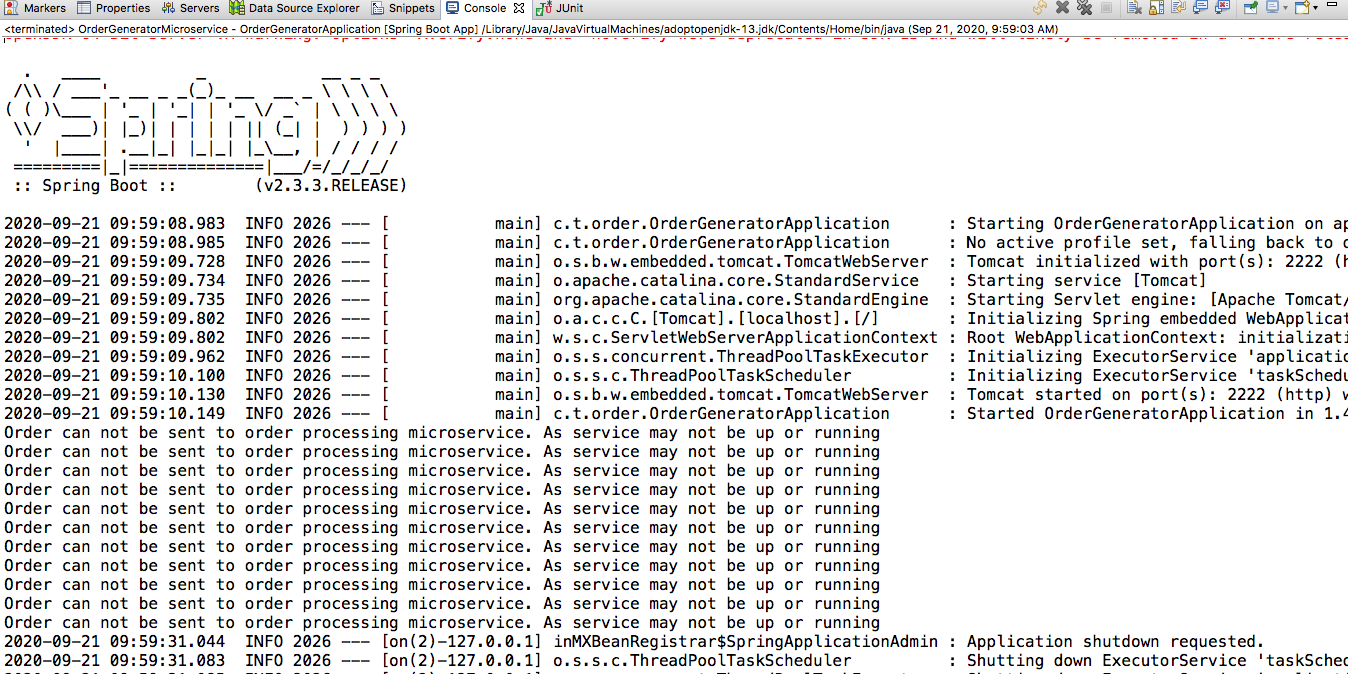
1. Import both above mentioned projects to Eclipse IDE.
2. Start services in order.
3. Run OrderProcessingMicroservice using Spring Boot App option.
4. Run OrderGenertaorMicroservice using Spring Boot App option.
5. Verify results in below mentioned formats.

**Order Generator Micro service**

Once the service is up and running it will automatically start generating orders based on sample input data after every 2 seconds and send them to order processing service and the resultant output can be verified on standard console output like below.



As both services are bound to communicate in synchronized way if Order Processing service is not up then Order Generation service will not be able to send orders and will result as following.



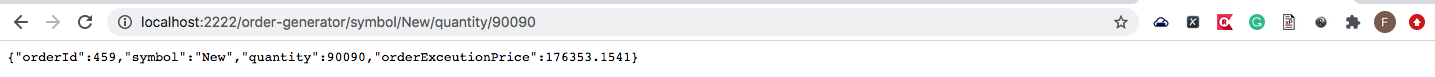
We can also test the above service end point using browser. An example request and response is shown below:

URL:

http://localhost:2222/order-generator/symbol/New/quantity/90090

Response:

{"orderId":459,"symbol":"New","quantity":90090,"orderExceutionPrice":176353.1541}



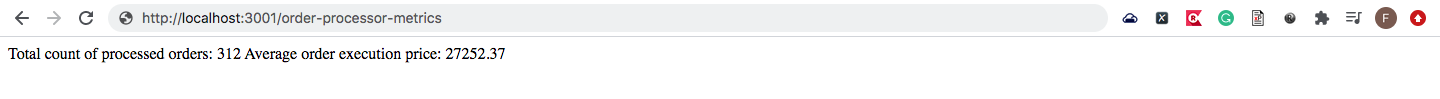
**Order Processor Micro service**

An example request and response is shown below:

URL: http://localhost:3001/order-processor-metrics

Response:

Total count of processed orders: 473 Average order execution price: 19109.768



Results can also be viewed on standard output console as below.



**Github Repository URL**

https://github.com/Faiza-Aftab/MicroservicesTask