**Prive programming challenge**

1. I also created dummy broker service, if don’t want to use please avoid.

**How to create the build**

Without IDE: - navigate to brokering where the pom.xml file is located and open terminal and use the command to build -> mvn clean install

With IDE(eclipse): - import the projects in eclipse both brokering, ordering applications and right click on the project, select option Run As -> Maven build -> clean install (Please find the screen shot below) and click on Run.

Graphical user interface, text, application, email

Description automatically generated

**How to Run the Services**

Without IDE: -

1. First, we need to run the dummy broker service, cd to extracted zip directory of brokering application and use the command -> brokering\target>java -jar brokering-0.0.1-SNAPSHOT.jar
2. Second, we need to run the ordering service, cd to extracted zip directory of ordering application and use the command -> prive-ordering-challenge\app\target>java -jar ordering-0.0.1-SNAPSHOT.jar

With IDE: - After successfully imported into eclipse Right click on the project main class file -> Run As -> Java Application

**How to access H2 in-memory database**

1. Open browser once the ordering application is up and running and use the URL to access H2 DB <http://localhost:8080/h2-console>
2. Use the below details to connect to DB
   1. Driver Class: org.h2.Driver
   2. JDBC URL: jdbc:h2:mem:challenge
   3. Username: root
   4. Password: root
3. Login required for H2 DB on every restart of the application.

Graphical user interface, application, Word

Description automatically generated

Text

Description automatically generated

**How to use Postman**

1. For new order

URL -> POST <http://localhost:8080/orders>

HTTP Request -> {

"qty" : 500,

"type" : "MARKET",

"code": "AAPL",

"price" : 312.10

}

Graphical user interface, text, application

Description automatically generated

1. For list the orders

URL -> GET <http://localhost:8080/orders>

1. For broker-callback API’s

URL using order req\_id -> <http://localhost:8080/broker-callback?req_id=78d31c75-73a1-4b0e-a73a-18ccae7dfedd&status=ACCEPTED>

URL using id -> <http://localhost:8080/broker-callback?id=96291f82-af7f-4eff-bd4e-ed26282aeedd&status=ACCEPTED>

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Happy Flow:**

Step 1: - create order by calling <http://localhost:8080/orders> with request body

Step 2: - call the broker call back API’s either by id or req\_id <http://localhost:8080/broker-callback?req_id=78d31c75-73a1-4b0e-a73a-18ccae7dfedd&status=ACCEPTED>

Step 3: - Repeat the above step one more time.

Step 4: - Now the order is created, and it is FULLFILLED by the broker.

**--- THE END ---**