VALR-ORDERBOOK-GUIDE

Assessment Specifications:

Language	Kotlin
HTTP Server	VERT.X
Author	Diveshan Naidoo
IDE	IntelliJ

API Endpoints Implemented:

POST Order

POST http://localhost:8080/api/order

JSON BODY to Submit:

JSON Content

```
{
  "type": "BUY",
  "quantity": 0.7,
  "price": 154000,
  "currencyPair": "BTCZAR"
}
```

Parameter	Description
type (required)	BUY or SELL
quantity (required)	Base amount can't be => 0
price (required)	Price to set order
currencyPair (required)	Currency pair that the order is for

Expected Success Response:

Status: 201 Created Size: 108 Bytes Time: 4 ms

Response Headers 3 Cookies Results Docs

```
{
  "type": "buy",
  "quantity": "0.70000000",
  "price": "154000.00",
  "currencyPair": "BTCZAR"
}
```

GET Orderbook

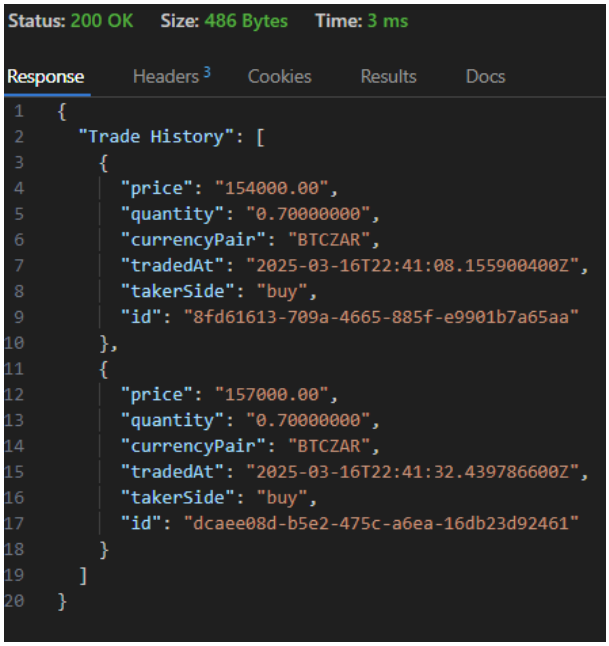
GET http://localhost:8080/api/orderbook

Expected Success Response:

GET Trade History

GET http://localhost:8080/api/tradehistory

Expected Success Response:



Orderbook Operational Specifications:

- Accepts buy and sell orders via api/order endpoint
- Orders are separated into buys and sells
- Buys are sorted by highest price first
- Sells are sorted by lowest price first
- Automatically matches BUY and SELL orders when a trade can be executed
- After trades are executed, they get stored and the orderbook updates dynamically
- Facilitates partial trade execution, with logic that adjusts the quantity values and re-inserts them into the orderbook
- Quantities are aggregated based on the price level
- If multiple orders have the same price, their quantities are aggregated, and the order count increments, instead of adding more entries into the orderbook
- Can retrieve trade history via the api/tradehistory endpoint, which shows the last 20 executed trades

UNIT Testing Results:

Test	Result
testGetOrderBook	PASSED
testSubmitValidBuyOrder	PASSED
testSubmitValidSellOrder	PASSED
testSubmitInvalidOrderNegativeQuantity	PASSED
testSubmitInvalidOrderNegativePrice	PASSED
testSubmitInvalidOrderMissingPrice	PASSED
testSubmitInvalidOrderMissingQuantity	PASSED
testSubmitInvalidOrderMissingCurrencyPair	PASSED
testSubmitInvalidOrderInvalidType	PASSED
testSubmitInvalidOrderMissingType	PASSED
testGetTradeHistory	PASSED
testExactMatchOrderExecution	PASSED
testPartialOrderExecution	PASSED
testHighPrecisionOrder	PASSED

Class Structure:

Class	Description
src/main/./././MainVerticle.kt	Class to initialise vert x server and configure endpoints
src/main/./././Order.kt	Class to store orders and trades
src/main/./././OrderBook	Class that contains backend logic which feed endpoints
src/test/./././MainVerticleTest.kt	Test Suite containing tests which verify the system