

Silver Bars Marketplace Readme

Following assumption made out of requirements provided by the business:

- API required to register, cancel and view summaries of order
- API will be used by UI teams
- Since UI teams will not directly use Java lib code, I'm providing a spring boot RESTfull service with a communication JSON payload.
- No database nor UI required
- Since no database is required, I'm using a ConcurrentHashMap in order to simulate a cache repository
- Mandatory fields to register an order are userId, quantity, price, orderType (BUY or SELL). If these fields are not provided an error status 0 and a message will be passed to the user. If successful status 1 and success message will be passed to the user.
- Mandatory field for cancelling an order is orderId. The orderId is auto generated during the order registration process and will be return to user. If orderId is not provided an error status 0 and a message will be passed to the user. If successful status 1 and success message will be passed to the user.
- Order will not be deleted during cancellation. Cancelled orders will be tagged with a orderStatus CANCELED. Newly registered orders will be tagged ACTIVE. I'm holding cancelled order in the repository for future historical reasons. Cancelled orders will not be processed in the order Summary calculation.
- Multi-currency nor quantity units are not mentioned. Therefore, I'm hard coding the summary as currency: £ and quantity unit as kg. I would have rather given an option to select currencies or quantity units.
- Please check unit test coverage for REST service.

Status code = Error (0) or Success (1)

Silver Bars Marketplace app runs as a Spring Boot RESTful service.
The port is currently set (application.properties) to 8081

Build & run test:

mvn clean install

Run spring boot service: mvn spring-boot:run

Test URLs:

registerOrder(Order order)<http://localhost:8081/registerOrder>

Payload:

```
{"userId":"user1","quantity":3.5,"price":306.0,"orderType":"SELL"}
```

Will return:

```
{  
  "status": 1,  
  "message": "Successfully Registered",  
  "orderId": 1,  
}
```

```
"userId": "user1",  
"quantity": 3.5,  
"price": 306,  
"orderType": "SELL",  
"orderStatus": "ACTIVE"  
}
```

cancelOrder(Order order)<http://localhost:8081/cancelOrder>

Payload:

```
{"orderId": "1"}
```

Will return

```
{  
  "status": 1,  
  "message": "Successfully Canceled",  
  "orderId": 1,  
  "userId": "user4",  
  "quantity": 2,  
  "price": 306,  
  "orderType": "SELL",  
  "orderStatus": "CANCELED"  
}
```

getOrdersSummary()

<http://localhost:8081/getOrdersSummary>

For registered users:

```
{"userId": "user1", "quantity": 3.5, "price": 306.0, "orderType": "SELL"}  
{"userId": "user2", "quantity": 1.2, "price": 310.0, "orderType": "SELL"}  
{"userId": "user3", "quantity": 1.5, "price": 307.0, "orderType": "SELL"}  
{"userId": "user4", "quantity": 2.0, "price": 306.0, "orderType": "SELL"}  
{"userId": "user1", "quantity": 3.5, "price": 306.0, "orderType": "BUY"}
```

Will return

```
{
```

```
"status": 1,  
"message": "Successfully Retrieved",  
"orderSummary": {  
  "SELL": [  
    "5.5 kg for £306.0",  
    "1.5 kg for £307.0",  
    "1.2 kg for £310.0"  
  ],  
  "BUY": [  
    "3.5 kg for £306.0"  
  ]  
}  
}
```