

Trade Master Application III

Problem Statement:

Problem Submissions Solutions Doubts

TradeMaster Application III

Easy • Score 80/80 • Spring Hibernate

Problem statement

In Continuation to the **Trade management system** application now you have to create a custom metric to monitor specific aspects of your application with the help of Prometheus and Grafana.

▼ Tasks:-

1. In the **monitoring** package, create a class with the name **MetricConfiguration** and annotate it with **@Configuration** annotation.
2. Create a method **public MeterRegistry getMeterRegistry()** in the **MetricConfiguration** class and do the following:
 - Create a "CompositeMeterRegistry" object and return it.
 - Annotate the method with "Bean" annotation.
3. Implement the following changes in the **TradeService** class:
 - Create a variable "getTradeHistoryCounter" of data type Counter in the **TradeService** class and make it null initially as shown below.

▼ Reference image

```
private Counter getTradeHistoryCounter = null;
```

- Increment the variable in the method **getTradeHistory(String username)** such that whenever a user tries to access the **"/tradeMasterApp/tradeHistory/{username}"** API, the value of the Count variable increases by 1.

- Increment the variable in the method **getTradeHistory(String username)** such that whenever a user tries to access the **"/tradeMasterApp/tradeHistory/{username}"** API, the value of the Count variable increases by 1.

- Define a constructor for **TradeService** as shown below

▼ Reference image

```
public TradeService(CompositeMeterRegistry meterRegistry){
    getTradeHistoryCounter =
        meterRegistry.counter("get.trade.history.counter");
}
```

4. Run the application.
5. After hitting the **"localhost:8081/actuator/prometheus"** API check that a metric with the name **"get _ trade _ history _ counter"** is being created.
6. Hit the **"/tradeMasterApp/tradeHistory/{username}"** API endpoint and check how our custom metric increments, informing us how many times the API was called.
7. Visualize the metrics data in Prometheus and Grafana by hitting **"localhost:9090"** and **"localhost:3000"**, respectively.

► Special Instructions for submitting the solution:

► Note:-