

## **Market Data Throttle Control:**

### **Before you being:**

- This exercise accesses your skill set from multiple dimensions. Including but not limited to programming ability, logical thinking, business analysis. In addition, the presentation skills in the upcoming interview.
- Please share any assumption made on the solution.
- The following requirements do not include edge cases. You can make your own assumptions and share in the submission.

### **Requirement:**

- There is a `MarketDataProcessor` class below. This class receives real-time market data from the exchange and then publishes them to other applications.
- `MarketDataProcessor` receives `MarketData` from a source through the `onMessage` method. There is a single thread that calls `onMessage` at an unknown rate per second.
- Modify the `MarketDataProcessor` class to,
  - At least fulfill,
    - Ensure that the number of calls of `publishAggregatedMarketData` method for publishing messages does not exceed 100 times per second, where this period is a sliding window.
    - Ensure that each symbol does not update more than once per sliding window.
  - Prefer to fulfill,
    - Ensure that each symbol always has the latest market data published.
    - Ensure the latest market data on each symbol will be published.
- The `MarketData` class contains the symbol, price and update time. The data types are determined yourself.

### **Notes:**

- No limitation to create any class to support the logic.
- No limitation to use any dependency library. Prefer open source such that our team can execute the project.
- If you have written any test cases, please include them in the submission.
- Data structure are in-memory. Don't need to implement any recovery or failover logic.
- Please implement the logic in Java. Don't use other JVM languages (e.g. clojure, Groovy, Scala).

---

```
public class MarketDataProcessor {  
    // Receive incoming market data  
    public void onMessage(MarketData data) {  
        // Please implement  
    }  
  
    // Publish aggregated and throttled market data  
    public void publishAggregatedMarketData(MarketData data) {  
        // Do Nothing, assume implemented.  
    }  
}
```

---

p.s. feel free to raise question through email