Test Case - .NET Tech

Overview

Please build a service which would provide REST API and WebSocket endpoints for live financial instrument prices sourced from a public data provider and will efficiently handle over 1,000 subscribers.

Requirements

1. REST API:
   - Endpoint to get a list of available financial instruments. Take just a few, for example: EURUSD, USDJPY, BTCUSD - will be sufficient
   - Endpoint to get the current price of a specific financial instrument.

2. WebSocket Service:
   - Subscribe to live price updates for a specific financial instrument(s) from the list above.
   - Broadcast price updates to all subscribed clients.

3. Data Source:
   - Use a public API like Tiingo to fetch live price data and instrument details https://www.tiingo.com/documentation/websockets/forex .
     Or this source can be used (it does not require a key, but has only websocket):

     ```
     wss://stream.binance.com:443/ws/btcusdt
     ```

     Subscribes to the server with the following message:

     ```
     {
       "method": "SUBSCRIBE",
       "params": [
         "btcusdt@aggTrade"
       ],
       "id": 1
     }
     ```

4. Performance:
   - Efficiently manage 1,000+ WebSocket subscribers with a single connection to the data provider. There is no need to simulate such a workload, just make comments in the code where you will handle that

5. Logging and error reporting:
   - Please implement event and error logging capabilities. Level of details and message structure is up to you

- No need to setup any logging platform, it’s ok to stream events to the console stdout

Further details

- Time for implementation: up to 4 days. If you need more time, please let us know
- When we run the solution provided we use **VS Code** and **MacOS**. So please make the documentation accordingly
- The result should look like:
    - Github repository with the code and instruction on how to run it on local computer
    - Any additional documents should be provided either as a shared Google Doc or in the email or in PDF attached to the email.
    - Any secrets (for example API key) should be provided in reply email