IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 8, Issue 5, October - November, 2020
ISSN: 2320 – 8791 (Impact Factor: 2.317)
www.ijreat.org

Development of Exchange Market Trade Application

[1]S.Dhanasekaran, [2]Yalamuri Dinesh, [3]Sana Venkateswara Rao
[1]Associate Professor at, Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education, Krishnan Koil 626126, Tamil Nadu, India
[2]Student at, Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education,Krishnan Koil 626126, Tamil Nadu, India
[3]Student at, Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education, Krishnan Koil 626126, Tamil Nadu, India

Abstract

BBXT TRADE APPLICATION SYSTEM" is software developed for managing various prices of market symbols.It is the desktop application which is written in .NET Framework.In global there are so many stock market exchanges and market symbols are there and having market symbols data for each exchange like orderbook data and trade transaction data these and all data are tracking in the desktop application.This particular project deals with tracking the market symbols data and showing the data in the market access viewer.It contains orderbook data and transaction details of market symbols.In this application easy to check the market symbols data. The BBXT TRADE APPLICATION is one of the desktop application which is written in c# programming language and using .NET framework. The project defines that showing the stock market exchange data like orderbook data , trade transaction data and timestamp of market symbol transactions in the Market Access Viewer. Each exchange have many market symbols which are BYBIT exchange have BTCUSD.
.

KeyWords
Websocket endpoint,Rest endpoint,.NET Framework,Orderbook data,Trade data, Connection Pool,Json Parser Library,WPF,Data Binding

1.Introduction

Each Exchange has their own website and their data . But it is difficult to track each exchange data by opening each website .If BYBIT shows their own data and BITFINIX shows their own orderbook data and trade transaction data . By this I have to open each website and have to track. Trade application methodology is easy to use and easy to track every exchange data. the approval processes.Showing all exchange market data like Orderbook data like bids and asks and Trade transactions data with timestamp in the market access viewer.By using public websocket Api's of each exchange market and integrating in the code like this grabbing all exchange data in one application.Output there will be a WPF viewer and having many tabs for Order Book,trade history and for quotes . For each exchange market data will be bind in viewer .In the orderbook data will be changing . There will be updating,modifying,deleting and adding the orderbook data .It is real time data which can be accessed from a remote server and binding in the viewer.

2 Methodology

2.1 For Websocket Integration:

Having Websocket Apis for each exchange market and having many exchange market symbols like BTCUSD,ETHUSD etc.There will be endpoints for getting market symbols,orderbook data and trade

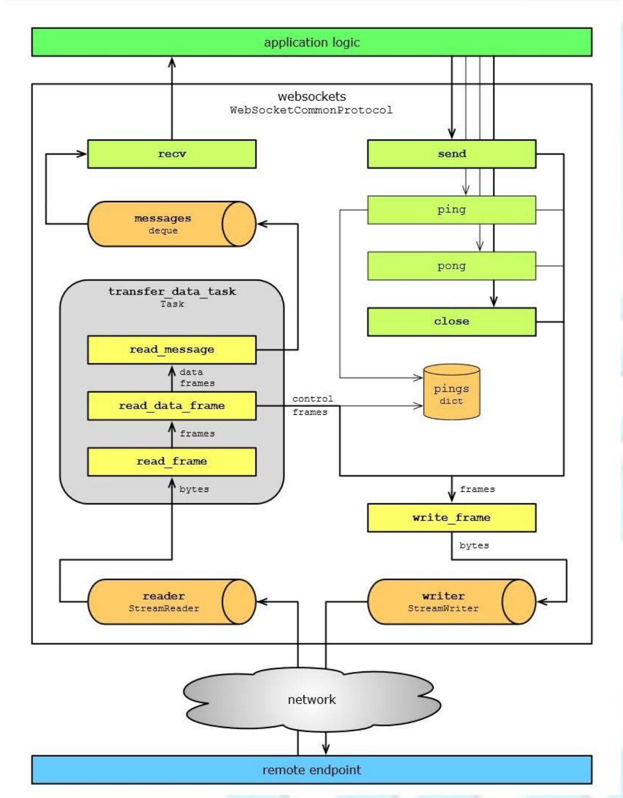
executions.Maintaining the websocket connections to keep maintaining the remote server connections then only can access the data of each symbol. After getting symbols, have to subscribe to the symbols and will get orderbook data and trade data.Using stream controllers to maintain data in the correct manner and using Json Libraries to parse the data and binding the data to WPF viewer.Proposed project having technical parts which have to subscribe the symbols and maintaining the websocket and rest API connections.

2.2 For Rest Api Integration:

Having Rest Apis for each exchange market and having many exchange market symbols like BTCUSD,ETHUSD etc . There will be endpoints for getting market symbols,orderbook data and trade executions.

First thing is getting symbols in our application by using symbols endpoint and the response will come as Json format.After getting JSON format response have to parse the symbols by using JSON libraries which are having in .NET.

Using Stream Controller for maintaining each symbols data like orderbook and trade executions when getting each symbols data and binding the data in WPF viewer.



2.3 Monitoring System

This system also monitors the activities done by the people involved in this process which provides additional security to the organization or company. This collects the data such as login and logout time, activity done by them during the login time such as Downloads done, messages and data sent from the login system. I.P address and the location of the system are also taken by the system in default and all the information is stored in the database helps in investigation to find out the persons involved in the crime during the tender process such as bid code fraud and to minimize the chances of fraud.

2.4 Data

Types of data that we collect from the data source:

Metadata - Static information that could help describe data

Transactions - Transactions, trades, executions, matches

Order books - Resting orders as deep and granular as possible

Quotes - Top orders, each from one side of the book (bid/ask)

General data - Statistics related to asset, symbol or data source

Data type - Metadata - Static information that could help describe data

2.4.1 Data type - Transactions, trades, executions, matches

Transactions (called also: trades, executions or matches) representing transactions occurred on the data source.done by the head of the organization or company and his expert and personal team.

Transaction must contain contain:

1.price

2.amount (in the base currency of the asset or number of contracts)

3.timestamp in the UTC timezone when we first received it

Transaction can also additionaly include:

1.an identifier of the transaction
2.aggressor of the trade (buy/sell)
3.timestamp in the UTC timezone when it occured on the exchange

2.4.2 Data type - Order books - Resting orders as deep and granular as possible

The order book for the specific markets representing the resting orders published by the market participants which are willing to buy or sell on given symbol. Sell orders in the order book are named "asks", buyers have named "bids". Each order contains price, amount and the type (bid or buy/ask or sell). Ask prices are always lower than the bid prices.

2.4.3 Data type - General data - Statistics related to asset, symbol or data source

General data purpose is to collect data that can't be classified into common data types supported by the framework.
List is long but some of the data types are generated automatically by the framework from other data (aggregated data).
General data item must contain:

- type
- value in the decimal format
- timestamp in the UTC timezone when we first received it

2.4.4 Data type - Quotes - Top orders, each from one side of the book (bid/ask)

Quote representing best bid and ask price at a given time.
Quote must contain contain:

- best ask price
- best bid price

Conclusion

This overall, can access all exchange market symbols data like orderbook and trade history or executions data.This is an easy way to monitor all markets data and don't want to open each exchange website to monitor . All the data will bind to the WPF viewer . We can select any exchange and can monitor the exchange market data.

References

[1] Simon Fong, Zhuang Yan “Design of a web-based tendering system for e-Government procurement.” In Proceedings of the 3rd International Conference, Bogota, Columbia, November 10-13, 2009.

[2]TejasC.Patil,Ashish.P ,P.S Gawande.“Tender and Bidding Process in Construction Projects ”IJISET - International Journal of Innovative Science, Engineering & Technology”, Vol. 3 Issue 3, March 2016. ISSN, 2348–7968.

[3]M.Majoros,"SOFTEST Model of a commercial program testsystem", Proc ACM Congress on Software Quality Assurance, 1982.

[4]C. Ramamoorthy, "Application of methodology for the validation of process

control software", IEEE Trans. Software Eng., vol. SE-7, 1981.

[5] H.M Sneed "Automated Software Quality Assurance", in preceding of IEEE Transactions on Software Engineering (Volume: SE-11 , Issue: 9 , Sept. 1985)

[6] Biffl S., Halling M." Investigating the Defect Detection Effectiveness and Cost Benefit of Nominal Inspection Teams", IEEE Transactions of Software Engineering 29(5), pp 385-397, 2003.

[7] M. Hoffman, D. Blei, and P. Cook. Easy as CBA: A simple probabilistic model for tagging tender. In Proc. ISMIR, pages 369–374, 2009.

[8] J. Kittler. Combining classifiers: A theoretical framework. Pattern Analysis and Applications, 1(1):18–27, 1998.

[9] R. Miotto, L. Barrington, and G. Lanckriet. Improving auto-tagging by modeling semantic co-occurrences. In Proc. ISMIR, pages 297–302, 2010.

[10] Biffl, S, Schatten A. & Zoitl A. " Integration of Heterogeneous Engineering Environments for the Automation Systems Lifecycle" Proceedings of the IEEE Industrial Informatics (INDIN) Conference, Cardiff, UK