Life Trading C# Coding Exercise

Version 1.0, March 2018

# Introduction

This exercise is designed to help us assess your .Net Framework coding capabilities with 2-3 hours’ worth of work.

The exercise is meant to replicate closely the real-world domain of trading, but in a simplified form.

# Requirements

1. Provide a responsive UI for displaying price updates from the provided Market Data Service
2. The display has 5 columns:
   1. Symbol – this is key to the data, meaning that when you receive an update for the same instrument, the grid should show the new data, replacing the old. Put another way, the updates aren’t simply added to the grid on each update
   2. BidQty – available quantity on the bid
   3. BidPrice – best bid price
   4. AskQty – available quantity on the offer
   5. AskPrice – best ask price
3. Other standard grid features such as grouping, sorting and filtering are outside the scope of this exercise.
4. A short video to show what the result should look like is below:



# What you’ll find in the skeleton solution

## MarketDataService.dll assembly

1. A precompiled assembly (External Dependencies\MarketDataService.dll) is already referenced in the supplied project.
2. Within the assembly, a PriceClient class is available, along with the matching IPriceClient interface that it implements.
3. The PriceClient synthesizes a client receiving data from a remote server, ready for passing downstream within the client, whereby the receipt of the data is not controller by the client (that is, the server sends data to the client at a rate that is uncontrolled).
4. Note that the PriceClient is a blocking client, so some level of multi-threading will be necessary to keep the User Interface responsive.
5. The PriceClient has a Start() method to allow for the flow of data to start once all your components have been initialized.
6. You should not need to change the implementation of this piece but are free to make note in your comments where this could be changed to help with the end-to-end implementation.

## MarketDataUI Solution

1. The MarketDataUI project within the solution is set to target .Net Framework 4.7.1.
2. The PricesUI class is the main form that gets created and shown upon startup.  
   It contains a standard DataGridView UI component for displaying the received data, which has also been slightly customised to help:  
     
   PricesDataGridView.AutoGenerateColumns = true;  
   PricesDataGridView.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;  
   PricesDataGridView.RowHeadersVisible = false;
3. Three menu items are present – Exit, Start and Stop.
   1. Exit – (already coded) exits the application
   2. Start – starts the feed of data from the PriceClient
   3. Stop – stops the feed of data from the PriceClient

# Notes

1. The resultant solution should compile using Visual Studio .Net 2017 Community Edition. Use of nuget packages are permissible, but please make sure that your packages are clearly listed and restored as part of the build.
2. Keep it simple! Overcomplicating the design leads to less reliable and less understandable software.
3. Write tests (where possible) to ensure your solution does what you want it to.
4. Pay attention to your separation of concerns. A sensible design will make writing tests easier.
5. Try to use C# 7.0 language features where it makes sense. There are several changes to the language in this version that help with code brevity/comprehensibility.
6. It’s not anticipated that you need to interpret or manipulate the BidQty/BidPrice/AskQty/AskPrice columns.