Scenario

You are a software engineer at 'Trading Places', a small but growing hedge fund, and you have been assigned to work on a new trading platform. This platform allows traders to register a strategy that the platform will execute. A strategy encompasses buying or selling a set number of shares when the defined condition of the strategy is met.

Trading Places uses a small broker, 'Reutberg', for issuing checking current prices and carrying out the buy and sell instructions.

Instructions

Your task is to create a simple service that allows traders to register these strategies for execution and then execute them as appropriate. The service should check the Reutberg price of the specified shares and buy or sell a defined quantity if the price moves by a specified ***percentage*** from the price at the time the strategy is registered.

For example:

- a trader might register a strategy to place a trade for the ticker (a code used to identify shares in a particular company) "MSFT", with the instruction to buy 10 shares if the price drops by 2%.
- a trader might register a strategy to place a trade for the ticker "GOOGL", with the instruction to sell 100 shares if the price goes up by 1%.

The traders should be able to register strategies with the service and also unregister them. A simple API to facilitate this is sufficient; there is no need to provide a UI.

When the strategy's conditions are met it should execute by calling the appropriate method against the Reutberg API. A strategy should only execute once.

You may reference any free to use external libraries of your choice.

For simplicity you can assume that all values referring to prices are in USD.

For the purposes of this exercise, a valid ticker identifier is an uppercase alphanumeric string of length 3 to 5 inclusive.

You are provided with skeleton ASP.NET Core service to get you started and should concentrate on filling in the business logic rather than the hosting. You should not need to change any of the existing code, however you are free to do so if you choose.

Resources

StrategyManagementService

A pre-configured Background Service. This will "tick" and execute the CheckStrategies method once a second. This method is currently empty and should be filled in to provide the core functionality.

TradingPlaces.Resources.dll

Library containing utility classes used by the service.

Reutberg.dll

Third party library, containing the service to connect to Reutberg infrastructure, issue price quotes and execute trades.