

Full-Stack Real-Time Stock Exchange Application

Background:

As a full-stack developer, you are tasked with creating a real-time stock exchange application. This application will consist of a back-end ASP.NET Core Web API to handle stock-related functionalities and a front-end Angular dashboard for users to view live stock data and create orders.

Back-End Task: ASP.NET Core Web API

Task: Real-Time Stock Exchange API

Requirements:

1. **Endpoint for Stock Data:**
 - Create an endpoint (/API/stocks) to retrieve real-time stock data. Include stock symbols, current prices, and timestamps.
2. **Endpoint for Stock History:**
 - Implement an endpoint (/API/stocks/{symbol}/history) to retrieve historical stock data for a given stock symbol.
3. **Endpoint for Order Creation:**
 - Create an endpoint to add orders that include stock symbol, order type (buy/sell), and quantity.
4. **Endpoint for Orders History:**
 - Create an endpoint (/API/orders) to retrieve orders.
5. **Real-Time Updates:**
 - Implement real-time updates for stock prices using technologies like SignalR.
6. **Authentication:**
 - Secure the API with a basic authentication mechanism, such as API key or token-based authentication.

Sample Stock Names for Back-End Task:

- AAPL (Apple Inc.)
- GOOGL (Alphabet Inc.)
- MSFT (Microsoft Corporation)
- AMZN (Amazon.com Inc.)
- TSLA (Tesla Inc.)

Front-End Task: Angular Dashboard

Task: Real-Time Stock Exchange Dashboard

Requirements:

1. **Stocks Dashboard:**
 - Create a page that displays a list of stocks in real-time. Show relevant information such as stock symbols, current prices, and real-time updates.
2. **Real-Time Stock Updates:**
 - Implement real-time updates for stock prices using Angular features or libraries like Angular WebSocket and RxJS.
3. **Order Creation Form:**
 - Implement a form or modal to allow users to create orders for a selected stock. Include fields for stock symbol, order type (buy/sell), and quantity.
4. **Orders History:**
 - Add a section to display the user's order history, including order type, stock symbol, quantity, and timestamp.

Additional Considerations:

- Implement responsive design for a seamless user experience on various devices.
- Ensure error handling and user feedback mechanisms are in place.
- Use Angular best practices for component structure and state management.

Sample Stock Names for Front-End Task:

- Apple Inc. (AAPL)
- Alphabet Inc. (GOOGL)
- Microsoft Corporation (MSFT)
- Amazon.com Inc. (AMZN)
- Tesla Inc. (TSLA)

*** Deliverables:**

- A fully functional real-time stock exchange application that meets the specified requirements.
- Documentation detailing how to run the application and use its features.
- Push code on public GitHub repository and send link.