Stock Trading n8n Workflow

This workflow automates aspects of stock trading, from market data acquisition and analysis to trade execution. It leverages n8n's ability to integrate with various APIs, such as market data providers (e.g., Marketstack) and brokerage APIs (e.g., Alpaca API).

Workflow Overview

1. **Trigger:** Initiate the trading process (e.g., scheduled, market event).
2. **Data Acquisition:** Fetch real-time or historical stock market data.
3. **Market Analysis:** Analyze the acquired data to identify trading opportunities.
4. **Decision Making:** Determine whether to buy, sell, or hold based on analysis and predefined rules.
5. **Trade Execution:** Place buy or sell orders via a brokerage API.
6. **Notification & Logging:** Record trade details and send notifications.

Detailed Steps and Suggested Tools

Phase 1: Trigger

- **Option A: Scheduled Trigger:**
    - **n8n Node:** `Cron`
    - **Description:** To periodically check market conditions and execute trades (e.g., every hour, at market open/close).
- **Option B: Webhook Trigger (for external alerts):**
    - **n8n Node:** `Webhook`
    - **Description:** If an external system (e.g., a custom market scanner, a news alert service) detects a significant market event, it can trigger this workflow via a webhook.

Phase 2: Data Acquisition

- **n8n Node:** `HTTP Request` (for Market Data API)

- **Description:** Use an `HTTP Request` node to connect to a market data API (e.g., Marketstack, Alpha Vantage, Finnhub) to retrieve relevant stock data. This could include:
    - **Real-time Quotes:** Current prices, bid/ask, volume.
    - **Historical Data:** Past prices (OHLCV - Open, High, Low, Close, Volume) for technical analysis.
    - **Company Fundamentals:** Earnings reports, financial statements (less frequent updates).
    - **News Feeds:** Relevant news articles that might impact stock prices.
    - **Authentication:** Ensure proper API key authentication as required by the data provider.

Phase 3: Market Analysis

- **n8n Node:** `Code` (JavaScript/Python), `Set`, `Function`
- **Description:** This is where the trading strategy is implemented. You can perform various types of analysis:
    - **Technical Analysis:** Calculate indicators like Moving Averages (MA), Relative Strength Index (RSI), MACD, Bollinger Bands. This typically involves using the `Code` node (Python is excellent for this with libraries like `pandas` and `ta-lib` if available, or manual calculations).
    - **Sentiment Analysis:** If you're acquiring news data, you could integrate with a sentiment analysis API (via `HTTP Request`) or use a simple keyword-based approach within a `Code` node to gauge market sentiment.
    - **Pattern Recognition:** Identify specific chart patterns or price movements.
    - **Data Transformation:** Use `Set` or `Function` nodes to transform raw data into a format suitable for analysis.

Phase 4: Decision Making

- **n8n Node:** `IF`, `Switch`, `Code`
- **Description:** Based on the analysis from Phase 3, the workflow decides whether to execute a trade. This involves setting up conditional logic:
    - **Trading Rules:** Define your buy/sell/hold rules using `IF` or `Switch` nodes. Examples:
        - `IF (RSI < 30 AND MA_short > MA_long) THEN BUY`
        - `IF (Price crosses below MA_200 AND Volume is high) THEN SELL`
    - **Risk Management:** Incorporate rules for position sizing, stop-loss, and take-profit levels.
    - **Confidence Score:** You might assign a confidence score to your trading signals and only proceed if it meets a certain threshold.

Phase 5: Trade Execution

- **n8n Node:** `HTTP Request` (for Brokerage API)
- **Description:** If a trading decision is made, use an `HTTP Request` node to interact with your brokerage API (e.g., Alpaca API, Interactive Brokers API, or others that offer programmatic trading). This will involve:
  - **Authentication:** Securely authenticate with your brokerage account.
  - **Placing Orders:** Send `POST` requests to the brokerage API's order endpoint with details such as:
    - `symbol`: The stock ticker.
    - `side`: `buy` or `sell`.
    - `type`: `market`, `limit`, `stop`, etc.
    - `qty`: Quantity of shares.
    - `price`: (For limit/stop orders) The specific price.
    - `time_in_force`: `day`, `gtc` (good 'til canceled), etc.
  - **Order Confirmation:** Process the response from the brokerage API to confirm the order was placed successfully and retrieve the order ID.
  - **Error Handling:** Implement robust error handling for failed orders (e.g., insufficient funds, invalid symbol, API errors).

Phase 6: Notification & Logging

- **n8n Node:** `Telegram`, `Slack`, `Email`, `Google Sheets`, `Database`
- **Description:**
  - **Trade Confirmation:** Send notifications about executed trades (buy/sell) including details like symbol, price, quantity, and profit/loss.
  - **Alerts:** Notify if a trading signal was generated but no trade was executed (e.g., due to risk management rules).
  - **Logging:** Record all trading activity (signals, decisions, executed trades, errors) in a `Google Sheet` or a `Database` for performance tracking and auditing.

Considerations

- **API Keys & Security:** Store all API keys and sensitive credentials securely within n8n's credentials management. Never hardcode them.
- **Paper Trading First:** Always test your automated trading strategies extensively with paper trading (simulated money) before deploying them with real capital.
- **Market Hours:** Be aware of market open and close times, and adjust your `Cron` triggers accordingly.

- **Latency:** For high-frequency trading, n8n might introduce too much latency. It's better suited for swing trading or longer-term strategies.
- **Error Handling:** Implement comprehensive error handling and retry mechanisms for API calls, especially for trade execution.
- **Regulatory Compliance:** Ensure your automated trading activities comply with all relevant financial regulations.
- **Internet Connection:** A stable and reliable internet connection is crucial for automated trading.
- **Brokerage API Limitations:** Understand the limitations and capabilities of your chosen brokerage API (e.g., supported order types, rate limits).

This workflow provides a solid foundation for building an automated stock trading system using n8n. Remember that successful trading requires continuous monitoring, analysis, and adaptation of strategies.