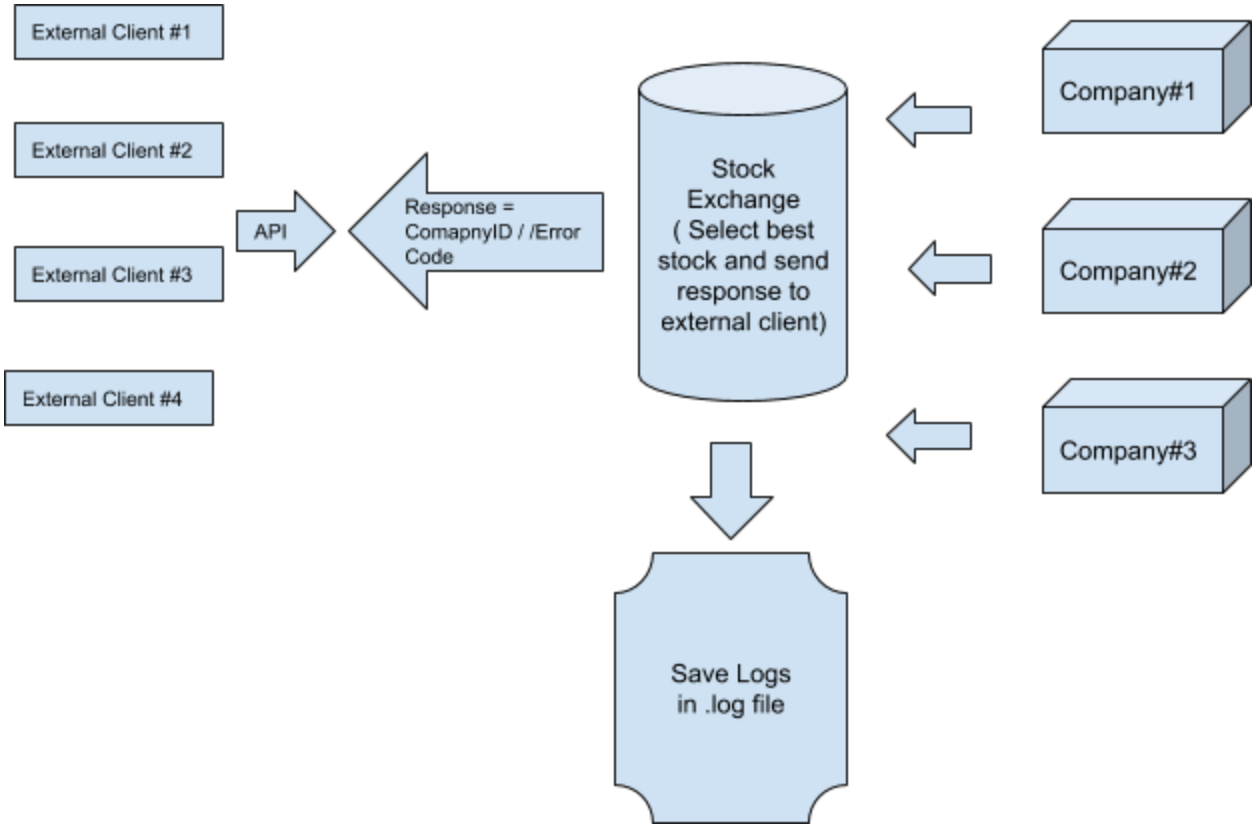
Hiring Assignment - Stock Exchange

Objective

Design & Code a stock exchange through which external clients can buy the stocks of various companies via an API. It needs to support **multi threading** so that many external clients can connect to the stock exchange API for buying stocks concurrently. The exchange will reply back with one company stock (highest price) based on targeting, budget & other factors.

Overflow Diagram



Components to build

- **API:** It will connect an external client to the stock exchange via the command line. API should consist of following parameters
    - Country code: US, IN, FR
    - Category: Automobile, Finance, IT
    - Base Bid: External client needs stock more than this bid **ex**. 10 cent

API Example: localhost.com/countrycode=US&Category=Automobile&BaseBid=10

- **Exchange Endpoint:** Endpoint which will connect the API and collect all the parameters from URL and pass it to exchange internal components.
- **Company's Internal Database:** This database will store all the companies' stock details and update it based on stocks sold to external clients.
    - **CompanyID:** Unique ID for company **ex**. C1, C2, C3.
    - **Country:** Countries where company is based **ex**. US, IN
    - **Budget:** The value of remaining stocks available in dollars **ex**. 10$
    - **Bid:** One company stock value in cents **ex**. 10cents
    - **Category:** Which category the company belongs in **ex**. Automobile, Finance

Sample Database:

Use this database for coding the assignment.

CompanyID	**Countries**	**Budget**	**Bid**	**Category**
C1	US, FR	1$	10 cent	Automobile, Finance
C2	IN, US	2$	30 cent	Finance, IT
C3	US, RU	3$	5 cent	IT, Automobile

- **Exchange Internal Matching Logic:** This is the most important component to code where all matching logic will sit. Follow this order for selecting best company and log all details into log file.
    - **Request Received:** Collect all parameters from API (Country, Category, Base Bid)
    - **Base Targeting:** Match companies based on Country, Category. If no company passed from these filters then send response as "No Companies Passed from Targeting" to external client. Include logs like: BaseTargeting: {C1, Passed},{C2,Failed},{C1,Passed}

    - **Budget Check:** Check if Companies had some budget to sell stocks. If no companies passed from the filters then send response as "No Companies Passed from Budget" to external client. Include logs like: BudgetCheck: {C1, Passed},{C2,Failed},{C1,Passed}

    - **BaseBid Check**: Check if the bid is more than the API base bid. If no companies passed from the filter then send response as "No Companies Passed from BaseBid check" to external client. Include in logs like: BaseBid: {C1, Passed},{C2,Failed},{C1,Passed}

    - **Shortlisting**: If more than one company passed from BaseBid check then select the highest one and send response = CompanyID. Include in logs like: Winner = CompanyID.

    - **Reduce Budget**: Once CompanyID is sent to external client then update Budget New = Budget - Bid and use Budget New for future transactions.

- **Logs File:** Create a single log file for logging all details and each transaction should be one row in it.
- **Multi Threading:** Build multi threading so that multiple external clients can send requests in parallel.
- **Unity and Integration Test**: Write Unity and Integrations test to cover all major work flow.

Technology to Use:

- **Language:** Node.js
- **Database**: Any Database.
- No frontend required.

End Goal

- **Provide exact steps to run code**
- Document explaining the end to end process.
- Provide the whole code setup.
- No over engineering is required
- Less features are ok with strong testing.