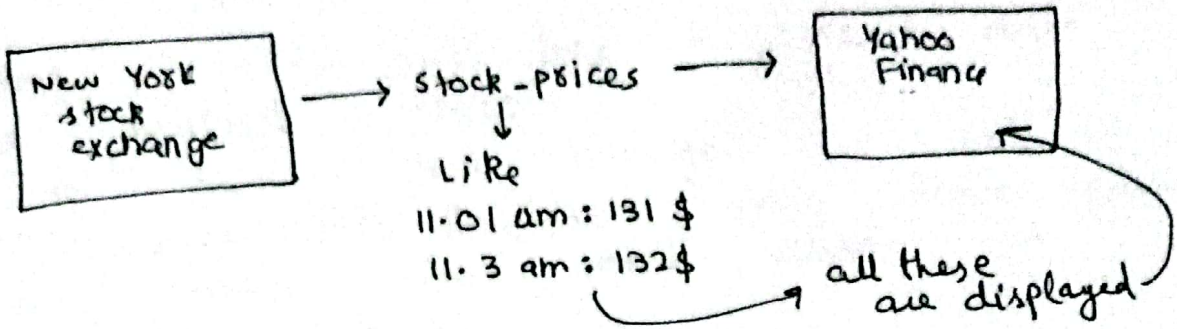
8) Queue :

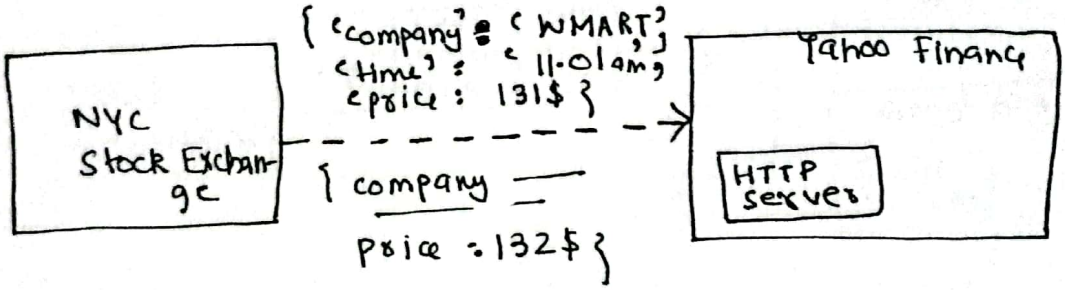
Consider a example of stock exchange website that show charts & stats. like Yahoo Finance.

How They work ?



→ Above means that NYC stock exchange sends the stock info to websik and they display it over there.

Approach 1 of Doing Above Thing :



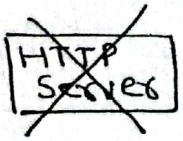
↳ So one thing we can do is we can make HTTP server for Yahoo Finance the accepts price posts requests.

→ So Team of NYC will make post calls along with JSON objects (data)

↳ So they will be displayed.

But There are some Issues:

Issue : 1 :



→ What if the HTTP server down/stop working ?

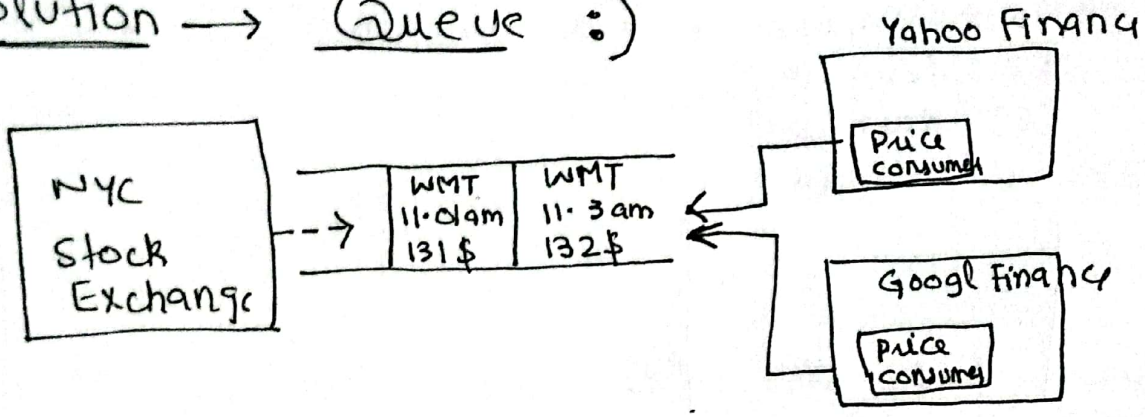
Scanned with CamScanner

So all the post request, (data we are posting will be lost).
↳ It can not be recovered.

ISSUES :2

→ What if NYC S-E has to send data to multiple users?
↳ They have to change the url of HTTP server post Request every time when they want to send data to different users.
↳ → Code Changes

Solution → Queue :)



↳ So The best way is:
To have a memory buffer where NYC S-E can put/store/send the prices one by one and all the platforms can retrieve in that order.

↳ Yahoo & Google will different pointers.
→ Like if Yahoo has consumed price 1 (11.3am) it's pointer will move to price 2.
↳ But Google's pointer will be at price 1.

Scanned with CamScanner

→ This Memory buffer is called Queue.
↳ Thing that will be put first will be out first in Queue.
→ FIFO Data Structure (Queue)
↳ First In First out

Implementation :

Python ⇒ ① List
② collections.deque
③ queue.LifoQueue
} → 3 approaches

C++ ⇒ Std::queue →

```
std::queue<int> q;
q.push(5);
q.push(6);
q.Pop(); //Returns 5
```

→ If we use lists. It's ok but we will have problems of dynamic array.
↳ It may cost too much.

Average Time complexities of Queue :

Access : O(n) → As it approaching last element every time

insert : O(1) → Printing at first place

search : O(n)

deletion : O(1)
↳ all same for worst cases.

Scanned with CamScanner