Interview Test tasks

Notebook:	doantvinh's notebook		
Created:	17-Jan-20 14:35	**Updated:**	17-Jan-20 17:34
Author:	doantvinh@gmail.com		
URL:	about:blank		

Write a websocket to streaming data from

"**wss://price-azu-01.vndirect.com.vn/realtime/websocket**"

works from 9am->11:30am to 1pm-2:30pm every workdays

1. Create connection to the above websocket

2. Set interval send message every 20s to keep the connection websocket
setInterval(function() {
socketClient.send('ping')
}, 20e3)

consume message to get data as following:

socketClient.send(JSON.stringify({ type: '**registConsumer**', data: { params: { name: <message_name>, codes: <codes> } } })
Danh sách các **message_name**: + STOCK: thông tin mã cổ phiếu + DERIVATIVE_OPT: thông tin mã phái sinh + MI:
Thông tin chỉ số index, Ví dụ muốn lấy thông tin của mã VND, SSI, thì cần consume message_name = **STOCK**, codes = ['VPB', 'SSI'] Ví dụ muốn lấy thông tin của mã phái sinhVN30F2002, thì cần consume message_name = **DERIVATIVE_OPT**, codes = ['VN30F2002']
Lắng nghe message từ price server và thực hiện action mong muốn với thông tin lấy về: socketClient.onmessage = function(message) { let parsedMessage = JSON.parse(message.data); // parse the message // do something with parsed message }
Ngừng consume message khi không cần:
socketClient.send(JSON.stringify({ type: '**stopConsume**', data: { params: { name: <message_name>, codes: <codes> } } })

Task 1:
Streaming data as above instructions for 3 codes:

- DERIVATIVE_OPT: get code 'VN30F2002'
- STOCK: get code 'VCB" and 'VIC'

Task 2:
Sample query data is as following:

{"type":"DERIVATIVE_OPT","data":"**717480000**|0.8|0.8937|0.895|0.8923|0.8922|0.2|0.5|0.2||0.9562|VN30F2003|||**0.8971**|20200319|0.8951|0.8951|0.1|0.8968|0.8969|0.897|0.3|0.4|0.9|20.9|

- parse data in different parallel

- split text by '|' in 'data' part of the message as a dict object
- the **bold** part is timestamp in mseconds, convert to datetime before saving
- save data to some types of database like json files / NoSQL databases as the following logic

Task 3:
Do the following calculation:

- for **DERIVATIVE_OPT** type:

examples:
{"type":"DERIVATIVE_OPT","data":"**717480000**|0.8|0.8937|0.895|0.8923|0.8922|0.2|0.5|0.2||0.9562|VN30F2003|||**0.8971**|20200319|0.8951|0.8951|0.1|0.8968|0.8969|0.897|0.3|0.4|0.9|20.9|
after split by '|' , get the value highlighted in **violet** in the array, for the first line that come every 5 minutes

- for **STOCK** type:

examples:
{"type":"STOCK","data":"10|**1579226872848**|09:07:52|VCB||S|111266323.4|22903092.5|89.5||||||95.7|83.3|||||||3|**89.7**|200|89.5|22||1|88|100|89.5|580||||89.5||||||"}

after split by '|' , get the value highlighted in **violet** in the array, is the price of the stock, at every 5 mins based on the timestamp.

- store the time and the value in violet to 3 different dataframes as example below:

DateTime, Close
2019-12-02 03:05:00+00:00,40100.0
2019-12-02 03:10:00+00:00,40200.0
2019-12-02 03:15:00+00:00,40150.0
2019-12-02 03:20:00+00:00,40050.0
2019-12-02 03:25:00+00:00,40000.0
2019-12-02 03:30:00+00:00,40000.0

- merge 3 dataframes by datetime colume
- calculate the correlation between VN30F2001 and VCB for last 3 hours.