ALGO TRADING IN PYTHON

#3 : REAL-TIME DATA STREAMING (WEBSOCKET)

Long Tran – Snap Innovations Pte Inc

COURSE OUTLINE

- Session/Week 1 : Trading strategy and backtesting in Python
- Session/Week 2 : Connect to the exchange (REST api)
- Session/Week 3 : Real-time data streaming (websocket)
- Session/Week 4 : Errors handling and Q&A

LAYOUT : SESSION #3

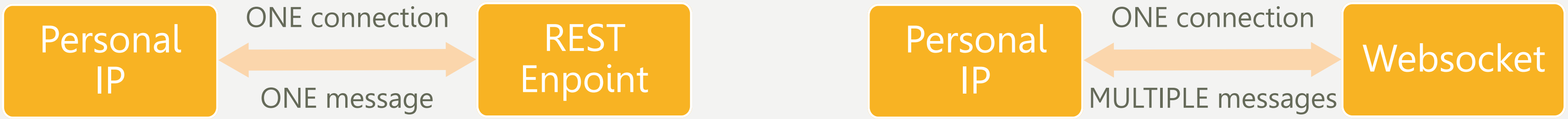
1. Websocket
2. Multi-threading
3. Algo Assembling
4. Python tricks and tips
5. Coding exercises

WEBSOCKET

- What is Websocket?

  A stateful protocol that maintains the initial TCP (Transmission Control Protocol) connection

- How it works?



Personal IP

ONE connection

ONE message

REST Enpoint

Personal IP

ONE connection

MULTIPLE messages

Websocket

- Support Python modules : requests, websocket
- Recommended packages :
  - tornado
  - binancepy : https://github.com/lambdamirror/Binance-Trading-Modules
  - igpy : https://github.com/lambdamirror/IG-Trading-Modules

MULTI-THREADING

- Support package in Python: threading
- Suggested threads distribution:
  - data_stream(ws) : send subscriptions to the exchange
  - strategy(ws) : generate signals upon the messages from the exchange
  - book_manager(ws) : control the life span of the signals generated in strategy(ws)
- Examples (on Binance):
  - Subscriptions : "btcusdt@aggTrade", "btcusdt@markPrice@1s “
  - Message:

```
{  "e": "markPriceUpdate",  // Event type
   "E": 1562305380000,      // Event time
   "s": "BTCUSDT",          // Symbol
   "p": "11185.87786614",   // Mark price
   "r": "0.00030000",       // Funding rate
   "T": 1562306400000       // Next funding time  }
```

ALGO ASSEMBLING

- Install support packages:

  websocket, threading, requests, urllib, hmac, hashlib, tqdm, matplotlib

  Cmd: pip3 install < *package name* >

- Modules:
  - BB_algo.py
  - tradingpy.py
  - binance.py
  - wss.py
  - (support) ultilities.py, indicators.py

***Note:** The algo runs on Linux/Ubuntu terminal.

PYTHON TRICKS AND TIPS

# LINUX useful commands:

Command	Function	Command	Function
cd	change direction	cat	read a file
mkdir	create new folder	nano / vim	modify a file
touch	create new files	mv	move/rename a file
rm -f	remove files (folders)	cp	copy a file

# Other Python commands:

```
$ sudo apt-get update
$ sudo apt-get install python3.6
$ sudo apt install python3-pip
$ python3 <filename>.py <args>
```

CODING EXERCISES

1. Modify on_message() function to handle messages of market depth, to print out 2 best bids and 2 best asks. Hint: adjust data_stream() function to subscribe to 'depthUpdate' stream first.
2. The websocket will be closed after the predefined running time. Write a while loop at the end of wss_run() to close all position after the websocket is closed. Note: you can freely decide the rules for closing the positions.
3. Impletment the logics:
   - If an ORDERED signal waiting too long on the market orders book, cancel the order.
   - If an ORDERED signal is only PARTIALLY_FILLED, cancel the remaining quantity after a certain period of time

***Instructions:**

- Modify wss_run.py for Problem 1
- Modify wss.py for Problem 2, 3