ALGO TRADING IN PYTHON

#2 : CONNECT TO THE EXCHANGE (REST API)

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 #2

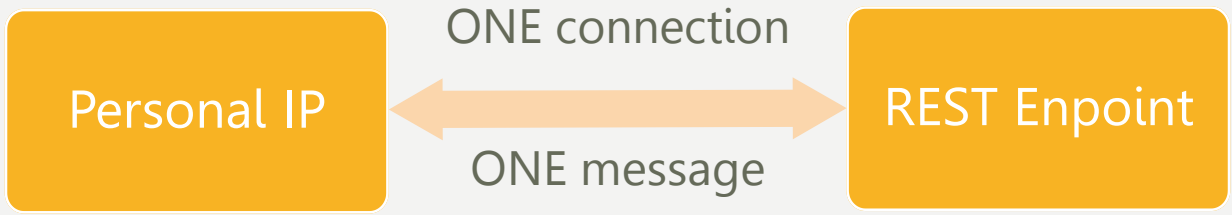
1. REST API
2. Request Structure
3. Message Handling
4. Python tricks and tips
5. Coding exercises

REST API

- What is REST (Representational State Transfer)?

  A set of rules to map a piece of data to an URL

- How it works?



- Support Python modules : requests, json, urllib, hmac, Crypto
- Recommended packages :
  - ccxt
  - binancepy : https://github.com/lambdamirror/Binance-Trading-Modules
  - igpy : https://github.com/lambdamirror/IG-Trading-Modules

REQUESTS STRUCTURE

- Components of a request
  - Enpoint (URL)
  - REST method: GET / POST / PUT / DELETE
  - Headers
  - Data (body)
- Example:

```
r = requests.post(  request_url ,
                    data=json.dumps(body) ,
                    headers=headers  )
```

MESSAGE HANDLING

- Components of a message
  - Headers
  - Data (body)
- To read the message's body:

  message = r.json()
- Example (on Binance):

  request = GET /fapi/v1/ping → message = {}

  request = GET /fapi/v1/time → message = { "serverTime": 1499827319559 }

PYTHON TRICKS AND TIPS

(binancepy) format and examples of responses is available at :

https://binance-docs.github.io/apidocs/futures/en/

(igpy) format of responses is available at :

https://labs.ig.com/rest-trading-api-reference

(ccxt) implied REST method :

params={"dualSidePosition":'true'}

exchange.fapiPrivate_post_positionside_dual(params)

→ POST /fapi/v1/positionSide/dual

CODING EXERCISES

1. Write a function to return a price level that sits between the best and second best prices available in the market.

For example: Assume the current market depth is

{ BID : ...1, 2, 3, 4 | 5, 6, 7, 8... : AKS }

then if we want to place a BUY order, the function should return price=3.5, and for SELL order, the function should return price=5.5

2. Build a class to control a **Portfolio** which does the following functions:
   - Control a list of tradable Instruments
   - position_locks() : if there is an open position of an Instrument in the account, remove it from the tradable list (consider hedge mode also)
   - equity_distribution() : return the order size (in dollar amount) given a percentage on the total equity, and the maximum number of buy/sell orders can be placed with the available equity
3. Requests for candle data has a limit for how many candles we can get upon one request, e.g limit=500. Write a function that requests data between startTime and endTime, that overcome the limit of one request.

Hint: try to excecute a sufficient number of requests, with each request gets the maximum number of candles.

***Instructions:** Fill in the blank in the REST_message_handle notebook, then run the next block to test the results