사용방법

1) xcoin\_api\_client.py의 \_\_init\_\_ 함수에 api\_key 와 api\_secret 변수에 각각 api key값과 secret 값을 넣어주세요

2) import

- from rest\_api import \*

3) 함수 호출 방법

- call\_api (api\_type, currency, rgParam={}, t\_sleep=5)

- 파라미터들

api\_type : api type으로 아래와 같은 enum을 넣어주세요 (문자열 아님)

GET\_PRC # Get Price (현재 시세 가져오기)

MKT\_BID # MARKET\_BID (시장가 구매)

MKT\_ASK # MARKET\_ASK (시장가 판매)

REQ\_ORD # REQ\_ORD (지정가 매수 또는 매도) : rgParam을 통해 지정가격, 수량, 매수/매도 타입 지정

CCL\_ORD # CANCEL ORDER 매매 대기 건 취소

CHK\_ORD # CHECK ORDER (지정가 매수/매도 결과를 가져오지만 이것은 사용하지 마세요.

# 아래에 내용이 있지만 chk\_order 함수를 사용하세요.

GET\_BAL # Get Balance (지갑 정보 조회)

currency : 'BTC', 'EOS', ....

rgParam은 필요한 인자를 map으로 구성. 아래 샘플예제를 참조해서 필요한 인자 확인하세요.

t\_sleep : 빗썸 서버가 try again 을 리턴할 때가 종종 있습니다. call\_api는 정상 리턴할 때까지 반복 시도하는 데

반복 시도하기 위한 sleep interval 입니다. (단위 : 초). 기본값은 5초 마다 재시도 합니다.

샘플 예제

- 현재 가격 얻어오기

>>> from rest\_api import \*

>>> r = call\_api(GET\_PRC, 'EOS') // EOS 현재 가격 얻어옴

>>> r

{'status': '0000', 'data': {'opening\_price': '6510', 'closing\_price': '6560', 'min\_price': '6350', 'max\_price': '6980', 'average\_p

rice': '6642.4484', 'units\_traded': '34792947.15503703', 'volume\_1day': '34792947.15503703', 'volume\_7day': '203422280.92040176000

0000000', 'buy\_price': '6560', 'sell\_price': '6570', 'date': '1522421010457'}}

가격 참조 : curr\_eos\_prc = r['data']['closing\_price']

- 지정가 매수/매도 는 rgParam의 'type' 속성을 'bid'나 'ask'로 지정해주고 나머지는 동일

- 지정가 매수

>>> r = call\_api(REQ\_ORD, 'EOS', {'units':0.1, 'price':7000, 'type':'bid'}) # 7000원에 0.1개 매수

>>> r

{'status': '0000', 'order\_id': '1522240318275677', 'data': [{'contNo': '15288358', 'units': '0.100000000000000000', 'price': '6760

', 'total': 676, 'fee': 0.00015}]}

-> order\_id 참조 r['order\_id']

-> r['status']=='0000' 이면 매수 또는 매도가 서버에 정상적으로 걸린 것.

- 지정가 매도

>>> r = call\_api(REQ\_ORD, 'EOS', {'units':0.1, 'price':7000, 'type':'ask'}) # 7000원에 0.1개 매도

>>> r

{'status': '0000', 'order\_id': '1522240512184440', 'data': [{'cont\_id': '15288637', 'units': '0.100000000000000000', 'price': '678

0', 'total': 678, 'fee': 1}]}

- 거래가 체결되었는지 판단하는 함수 (bid/ask type까지 지정해주어야 됨)

- chk\_order(order\_id, currency, bidask, amnt)

- 파라미터들

order\_id는 체결되었는지 확인하고자 하는 매매 건 id로 REQ\_ORD 를 통해 리턴결과로 온 'order\_id'속성값 결과를 넣어줍니다.

currency : 코인 문자열 'EOS', 'BTC', ...

bidask : bid 인지 ask인지 enum 값을 넣어주세요. BID, ASK 두 개의 상수가 정의되어 있음. 문자열이 아닙니다.

amnt : 매수나 매도 걸었던 수량(amount)를 넣어주세요. 이를 기준으로 체결되었는지 판단합니다.

-> 거래가 다수의 건으로 체결될 수 있는 데 결과가 각각의 거래 건 정보로 옵니다. 각 거래건들의 체결 수량을 모두 합해서

이 수량과 비교해서 체결완료되었는지 판단합니다.

샘플 예제

>>> rr = chk\_order('1522240512184440', 'EOS', ASK, 0.1) # r은 지정가 매도 실행 결과

>>> rr

{'result': True, 'status': '0000', 'amnt': 0.1, 'krw': 678, 'fee': 1.0, 'prcs': 6780, 'date': 20180328, 'time': 213512}

rr['result'] == True 이면 거래가 체결된 것입니다.

: amount == rr['amount'] 로 거래 체결여부를 판단하지마세요. bid의 경우 수수료 수량이 빠진 개수가 return 됩니다.

- 거래 취소 함수

- ccl\_order(order\_id, crcy, bidask) # cancel order

>>> r = ccl\_order(r['order\_id'], 'EOS', ASK)