

Используя <https://github.com/binance/binance-futures-connector-python>

```
import time
from binance.um_futures import UMFutures
um_futures_client = UMFutures()
max_pr = '0'
max_tm = '0'
cur_pr = '0'
prev_pr = '0'
val_d = {}
val_d[max_pr] = max_tm
time_ctf = 3.6e6
while(True):
    #print(sorted(val_d.items()))
    prev_pr = cur_pr
    cur_tm = um_futures_client.time()['serverTime']
    cur_pr = um_futures_client.ticker_price("XRPUSDT")['price']
    if (int(cur_tm)-int(max_tm) > time_ctf):
        val_d.pop(max_pr, None)
        for entry in sorted(val_d.items(), reverse=True):
            if (int(cur_tm)-int(entry[1]) > time_ctf):
                val_d.pop(entry[0], None)
            else:
                max_pr = entry[0]
                max_tm = entry[1]
                break
        val_d[cur_pr] = cur_tm
    if cur_pr >= max_pr:
        max_pr = cur_pr
        max_tm = cur_tm
    elif (float(cur_pr) <= 0.99*float(max_pr)) and (cur_pr != prev_pr):
        print("Максимальная цена за последний час: ", max_pr, "; цена на данный момент: ", cur_pr, "; падение на ", round((float(max_pr)-float(cur_pr))/float(max_pr), 5)*100, "%")
        time.sleep(100/1000)
```

Для работы с несколькими парами, необходимо сохранять все значения отдельно в списки, чтобы вести учёт за каждой из пар отдельно. Также, для достижения максимальной скорости, необходимо учесть, что хард лимит на "вес" всех вызовов за минуту является 1200. При этом, вызов цены для одной пары весит 1, а вызов цены для всех возможных пар весит 2 (вызов цены только для определенного списка пар невозможен). В связи с этим возможно два решения:

- 1) Вызов цен всех пар один раз за итерацию, а затем его фильтрация и дальнейшая работа с ценами для каждой пары, что позволит проверять цену чаще, но в свою очередь, может появиться проблема с точностью зависимости цены от времени, зависящая от характеристик машины, на которой выполняется программа, так как работа с каждым значением из получаемого нами списка цен занимает определенное время, а значит, что чем дальше мы будем идти по списку, тем выше будет вероятность расхождения цены от

времени, когда она была получена, что может быть заметно на парах с высокой волатильностью.

- 2) Вызов цены для каждой пары отдельно, что займёт больше веса для обхода всех значений (а значит уменьшится количество возможных проверок цен в минуту), чем первое решение при количестве пар  $> 2$ , но в свою очередь будет обладать большей точностью, со значениями менее зависящими от характеристики машины.