

Il codice:

```
from ibapi.client import *
from ibapi.wrapper import *

class TestApp(EClient, EWrapper):
    def __init__(self):
        EClient.__init__(self, self)

    def nextValidId(self, orderId: OrderId):

        mycontract = Contract()
        mycontract.symbol = "AAPL"
        mycontract.secType = "STK"
        mycontract.exchange = "SMART"
        mycontract.currency = "USD"

        self.reqContractDetails(orderId, mycontract)

    def contractDetails(self, reqId: int, contractDetails:
        ↪ ContractDetails):
        print(contractDetails.contract)

        myorder = Order()
        myorder.orderId = reqId
        myorder.action = "BUY"
        myorder.tif = "GTC"
        myorder.orderType = "LMT"
        myorder.lmtPrice = 1440.80
        myorder.totalQuantity = 1

        self.placeOrder(myorder.orderId, contractDetails.contract,
            ↪ myorder)

    def openOrder(self, orderId: OrderId, contract: Contract,
        ↪ order: Order, orderState: OrderState):
```

```

    print(f"openOrder. orderId: {orderId}, contract:
    ↪ {contract}, order: {order}")

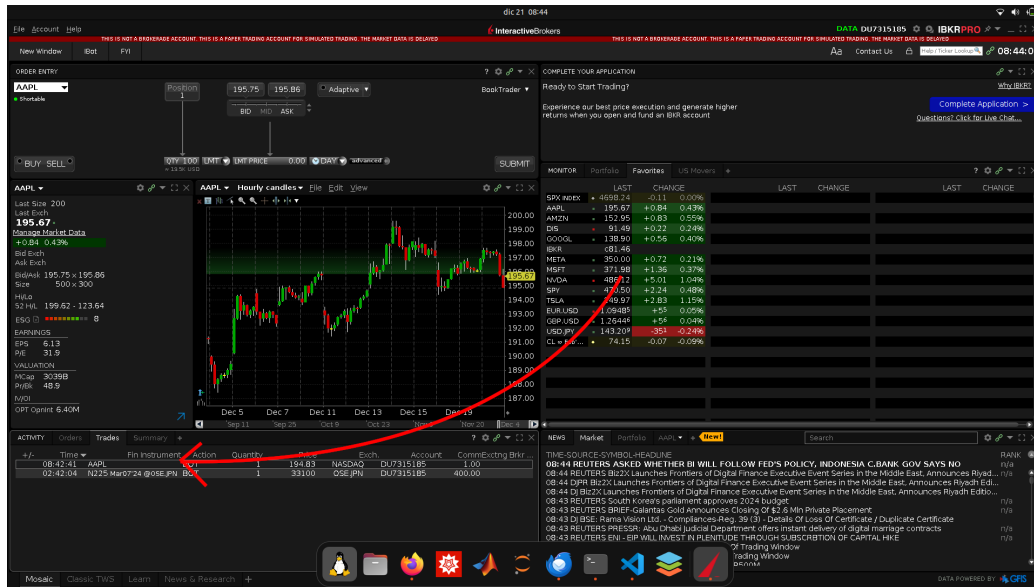
def orderStatus(self, orderId: OrderId, status: str, filled:
    ↪ Decimal, remaining: Decimal, avgFillPrice: float, permId:
    ↪ int, parentId: int, lastFillPrice: float, clientId: int,
    ↪ whyHeld: str, mktCapPrice: float):
    print(f"orderId: {orderId}, status: {status}, filled:
    ↪ {filled}, remaining: {remaining}, avgFillPrice:
    ↪ {avgFillPrice}, permId: {permId}, parentId: {parentId},
    ↪ lastFillPrice: {lastFillPrice}, clientId: {clientId},
    ↪ whyHeld: {whyHeld}, mktCapPrice: {mktCapPrice}")

def execDetails(self, reqId: int, contract: Contract,
    ↪ execution: Execution):
    print(f"reqId: {reqId}, contract: {contract}, execution:
    ↪ {execution}")

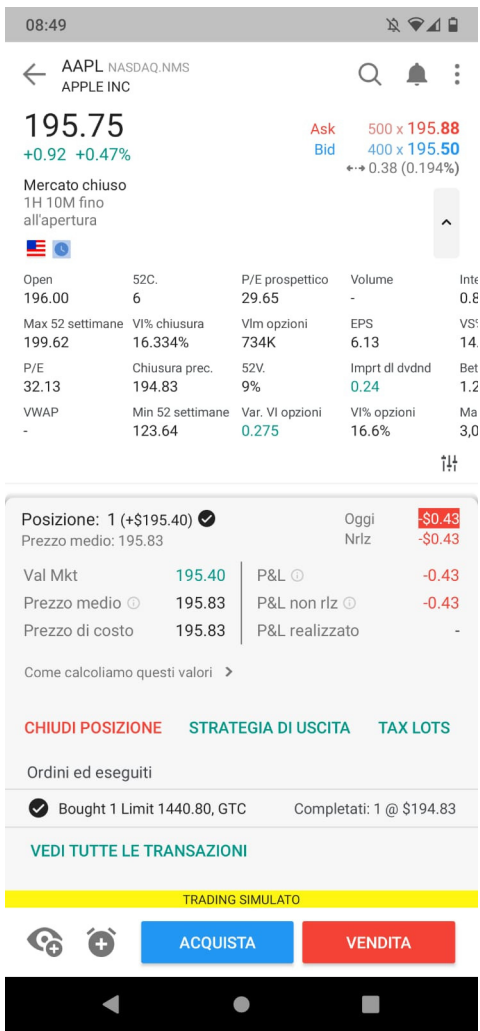
app = TestApp()
app.connect("127.0.0.1", 7497, 100)
app.run()

```

La cosa interessante è che appare anche sul tws:



Che nell'app:



Questo mi risparmia di mettere giù una dashboard per il controllo manuale in python. Tra l'altro ibkr mette prioritarie le operazioni da tws e app che le api, salva culo incorporated(1).

Comunque come dicevo la parte interessante è l'analisi. Ora scarico i dati e faccio un modello di media mobile semplice con una finestra di 5.

$$M = x_1 + x_2 + x_3 + x_4 + x_5 / 5.$$

O:

$$M = \frac{x_1 + x_2 + x_3 + x_4 + x_5}{5}$$

Che in python diventa:

```
a = x_1 + x_2 + x_3 + x_4 + x_5  
b = a / 5
```

Esiste una comoda funzione per farlo con una riga ma preferisco fare le cose a manina quando ho tempo. Il modello lo possiamo formalizzare con una sommatoria. Ma perchè complicarci la vita in modo formale?

$$\frac{1}{k} \sum_{i=n-k+1}^n p_i$$

Beh, guardate come diventa in python puro lavorando formalmente, molto più semplice!

```
a = [10,2,3,4,5]  
b = 0
```

```
for x in a:  
    b = b + x
```

```
y = b/5
```

E qui una seconda analisi, questo è lo stocastico con il metodo Monte Carlo (Usando numpy e non a manina):

```
S0 = 100
r = 0.05
sigma = 0.25
T = 2.0
I = 10000
ST1 = S0 * np.exp((r - 0.5 * sigma ** 2) * T +
sigma * math.sqrt(T) * npr.standard_normal(I))
```