qmt 订单撤补策略示例.py

Python

```
'''
    订单撤补策略 -- 单股

    以指定价发出委托后，订单5分钟未成交，撤销订单，以最新价补单，并追踪成交情况

    指定价: 前收盘价*(1-下跌比例)


'''
import pandas as pd
import numpy as np
import datetime


def init(ContextInfo):
    ContextInfo.stock_code = ContextInfo.stockcode + '.' + ContextInfo.market
    ContextInfo.trade_code_list = [ContextInfo.stock_code]
    ContextInfo.set_universe(ContextInfo.trade_code_list)
    ContextInfo.accID = str(account)    # 资金账号

    # 参数
    ContextInfo.drop_rate = 0.09 # 下跌比例


def handlebar(ContextInfo):
    # 如果不是最后一根bar，则忽略
    if not ContextInfo.is_last_bar():
        return
    now_time = timetag_to_datetime(ContextInfo.get_bar_timetag(ContextInfo.barpos), '%Y
```

```
-%m-%d %H:%M:%S')

    order_df = order_info(ContextInfo)    # 委托信息
    deal_df = deal_info(ContextInfo)        # 成交信息

    # 当前股票在成交列表中，则提醒推出实盘交易
    if ContextInfo.stock_code in deal_df['股票代码'].values:
        print('证券代码: {} 已成交, 请停止实盘交易程序'.format(ContextInfo.stock_code))
        return

    # 获取账户前收盘价, 并计算买入限价
    before_close_dict = ContextInfo.get_history_data(10, '1d', 'close')
    before_close = before_close_dict[ContextInfo.stock_code][-2] # 前收盘价
    print('前收盘价:',before_close)
    buy_limit_price = before_close * (1-ContextInfo.drop_rate) # 指定买入价

    last_price = ContextInfo.get_market_data(
                    ['quoter'],stock_code=ContextInfo.trade_code_list,
                    skip_paused=False,
                    period='tick',
                    dividend_type='front_ratio')
    last_price = last_price['lastPrice']
    # 最新价 <= 指定买入价 and 当前股票不在委托列表中，则以指定买入价，买入股票
    if last_price <= buy_limit_price and ContextInfo.stock_code not in order_df['股票代
码'].values:
        order_lots(ContextInfo.stock_code, 10, 'fix', buy_limit_price, ContextInfo, Con
textInfo.accID)

    if ContextInfo.stock_code in order_df['股票代码'].values:
        stock_order_df = order_df[order_df['股票代码'] == ContextInfo.stock_code]
        print(stock_order_df)
        for index, stock_order_se in stock_order_df.iterrows():
            stock_order_se = stock_order_se.to_dict()
            order_id = stock_order_se['合同编号']
            entrust_time = stock_order_se['委托日期'] + stock_order_se['委托时间']
            td_strptime = datetime.datetime.strptime(entrust_time, '%Y%m%d%H%M%S')
            delta= datetime.timedelta(seconds=20)
            yd_strptime = td_strptime + delta            # 600s之后的时间点
            # 委托未成交，且委托超600S,则撤单，并以最新价补单
            if can_cancel_order(order_id, ContextInfo.accID, 'stock') and str(yd_strpti
me) <= now_time:
                cancel(order_id, ContextInfo.accID, 'stock', ContextInfo)    #撤单
                order_lots(ContextInfo.stock_code, 10, ContextInfo, ContextInfo.accID)
# 以最新价补单


# 获取委托信息
def order_info(ContextInfo):
    order_df = pd.DataFrame()
    order_list = get_trade_detail_data(ContextInfo.accID, 'stock', 'order')
    for index, obj in enumerate(order_list):
        order_df.loc[index, '股票代码'] = obj.m_strInstrumentID + "." + obj.m_strExchang
```

```
eID
        order_df.loc[index, "委托价格"] = obj.m_dLimitPrice
        order_df.loc[index, "委托量"] = obj.m_nVolumeTotalOriginal
        order_df.loc[index, "合同编号"] = obj.m_strOrderSysID    # 合同编号
        order_df.loc[index, "已成交量"] = obj.m_nVolumeTraded
        order_df.loc[index, "委托剩余量"] = obj.m_nVolumeTotal
        order_df.loc[index, "委托日期"] = obj.m_strInsertDate
        order_df.loc[index, "委托时间"] = obj.m_strInsertTime
        order_df.loc[index, "委托状态"] = obj.m_nOrderStatus
    order_df = order_df.sort_values(by='委托时间', ascending=True)
    print(order_df)
    return order_df

# 获取成交信息
def deal_info(ContextInfo):
    deal_df = pd.DataFrame()
    deal_list = get_trade_detail_data(ContextInfo.accID, 'stock', 'deal')
    for index, obj in enumerate(deal_list):
        deal_df.loc[index, "股票代码"] = obj.m_strInstrumentID + "." + obj.m_strExchange
ID
        deal_df.loc[index, "合同编号"] = obj.m_strOrderSysID   # 合同编号
        deal_df.loc[index, "成交日期"] = obj.m_strTradeDate
        deal_df.loc[index, "成交时间"] = obj.m_strTradeTime
    deal_df = deal_df.sort_values(by='成交时间', ascending=True)
    print(deal_df)
    return deal_df
    return deal_df
```