TrendTrader Manual - Version 0.1

As of 2020-07-19

Overview of the Structure

Key components for the TrendTrading (trtrading):

- **controller**: defines the schedule to check version of Kiwoom API and to run the trtrader (mainly to cope with Kiwoom's infrequent API update while automate trtrader as much as possible)
- **daytask**: provides code for controller to run for version check and trtrader
- **Kiwoom**: defines Kiwoom API functions
- **trtrader**: defines trtrader main db (master_book) and main logic
- **extlistgen**: provides external buy and sell command to trtrader
- **bounds.xlsx**: provides levels of LLB, LB, UB per the number of reinvestments

Controller

- Currently, main tasks of controller are to run Kiwoom API well before market open for version check, and to run trtrader during market open
- Running sequence:
  - if executed before version check time, controller will run Kiwoom API and connect to server at the version check time (by executing daytask); if version check is unsuccessful, the controller exits (done by checking trade log file)
  - if executed after version check time and before trtrader run time, controller will run trtrader (by executing daytask) at the trtrader run time (e.g., market open time)
  - if executed after trtrader run time, controller will run trtrader (by executing daytask) immediately if the current time is before the market close time
  - a single trtrader run (or a single daytask run) will finish once the market closes (or when version check is done)
  - at each day's market closes, controller will be in a wait mode for until the next day's version check time, and the daily routine continues
- controller uses os.system("python daytask.py"), which might not be recommended way to run nested python code. However, Kiwoom API seems not properly finishes under other methods (e.g. by creating Kiwoom API object and remove/delete the object would not result in clean removal of Kiwoom API), and then it is not likely possible to automate the trtrader running everyday (with proper version check)

daytask

- runs Kiwoom API by creating a Kiwoom instance if run time is before version check time defined in controller
- runs trtrader by creating a trtrader instance during market open time
- checks holidays as defined in controller

Kiwoom

- Functions related with Kiwoom API are implemented (refer to Slack autotrading section for description, or the original book on wikidocs.net)

trtrader

- Fundamental principle of trtrader

  Select target and timing, and when the decision is right make follow-up investments to maximize earning and when not minimize loss

Overview of the algorithm

- Look for targets that have high potential of substantial price increase (value) in the near future (timing)
- Make a test investment of a ticket size into each of the targets
- If the choice of the target and the investment timing are both right, the target's price should increase: if the price hits a certain level (UB), then make follow-up reinvestment of a ticket size
- On the other hand, if either the choice of the target or the timing is wrong then the price should decrease: if the price hits a certain level (LB), then sell all shares of the target at a loss
- The loss should be small as it is a loss with respect to initial investment of a ticket size; consider this as the price for the wrong decision
- However, as trtrader will only trade during market open time, there could be price jump between market close and open, and sometimes there could be large price decline without time to react
- In such a case, it is most likely due to a macro event affecting multiple stocks at the same time, and rash selling due to the algorithm should be avoided: therefore, if the price hits a certain level (LLB) lower than LB, the selling should be suspended
- Once suspended, the stock should be held until the price recovers to higher than LB, and then it could be traded normally under trtrader logic
- For stocks that are reinvested, they follow the same logic as described above, but the bounds (UB, LB, LLB) are updated accordingly
- Max number of reinvested is predefined, and max number of bounds elevation is also predefined (as there could be bounds elevation without making reinvestment)

- Database structure
  - Transaction history is managed in master_book DataFrame: loaded from excel (or newly created) and saved to excel
    - Active items in master_book are the current account holdings
    - Only Kiwoom API transaction results are added into master_book (including reinvestments) as a new record
    - Bounds elevation, changing status to suspend due to hitting LLB, releasing, and etc which adjusts records in master_book that are not involving actual Kiwoom API transactions are handled in trendtrading_mainlogic
  - trtrade_list DataFrame is a run-time DataFrame which contains orders to be executed in Kiwoom API
  - external_list is a DataFrame that is loaded from an excel file and added to trtrade_list
  - Kiwoom trade_log text file is a separate log file run by Kiwoom class
- Procedeure of trtrader running
  - Kiwoom API is connected to Kiwoom server (currently test server) using account number defined herein
  - Bounds are loaded from an external excel file
  - master_book is loaded or created
    - Refer to the comments in trtrader next to the definition of CREATE_NEW_MASTER_BOOK
    - If newly created, an empty excel file with column names are created with definition of START_CASH
    - master_book is (then) loaded from the excel file
  - master_book integrity is checked
    - If master_book is newly created, integrity checker will load existing stock list to the master_book
    - Cash level is checked whether account cash is actually larger then START_CASH negative investment total of account holding stocks, and set cash level in master_book to START_CASH - investment total (plus buying fee)
    - If master_book is loaded from an existing master_book excel file, then the integrity checker checks whether the records in master_book matches the current account holdings

    - Tax and fee are adjusted when matching as the tax/fee calculation method for master_book is different from the way for record received through Kiwoom API
    - The integrity checker always ignores items in EXCEPT_LIST, and raises Exception if master_book contains items in EXCEPT_LIST (although not explicitly checking, it checks the length of active records matches between master_book and current holdings instead)
  - tax and fee adjustments
    - tax: 0.25%, fee: 0.35% for buy and sell each under Kiwoom test server
    - when buying, buying fee is deducted from cash while investment total (invtotal) is not affected
    - for evaluating the current holding total value (cvalue), tax and selling fee are already deducted
  - main procedure: run_(), trendtrading_mainlogic_(), trade_stocks()
    1. run_ executes trendtrading_mainlogic that would result in appending to the existing trtrade_list by checking master_book (for details about trendtrading_mainlogic, refer to the comments in the code; basically implements the trtrading algorithm described above)
    2. run_ checks if there is external_list and loads if exists
    3. run_ runs trade_stocks would execute "yet" items in trtrade_list
    4. "failed" items in trtrade_list would not retried to trade_stock
       - This might be fine if the item in the trtrade_list is generated through trendtrading_mainlogic
       - However, if the item is loaded from external list, failed item might be lost due to this no-retrial
  - closing procedure: close_()
    - saves master_book to the excel file (overwrite and prev excel file is lost - which is no concern as master_book contains all prev data)
    - prints master_book in easier readable format
    - prints result summary (e.g., overall return rate)

extlistgen

- generates an excel file to be loaded by trtrader that contains list of orders to be executed by Kiwoom API
- although the external list excel file contains orders, trtrader adjust each order according to the following rules:
  - codes in EXCEPT_LIST are ignored
  - codes for buying currently holding stocks are ignored
  - codes for buying with zero volume is set to buy a ticket size
  - if cash (which is being checked) is not enough for buying, system raises Exception
  - codes for selling stocks that are not in account are ignored
  - codes for selling with quantity either zero or more than holding are adjust to selling max quantity

bounds.xlsx

- Located as defined in trtrader
- defines levels of LLB, LB, UB according to the number of reinvestments
- Later this excel file might has to be incorporated into trtrader or a python code at least for parameter optimization