

Тестовое задание

19 октября 2021 г.

Предположим, что мы хотим разработать робота, который будет зарабатывать на разнице в цене между двумя биржами - так называемая арбитражная стратегия. Тестовое задание состоит в реализации самого простого варианта этой стратегии (и нерабочей в реальных условиях, не торгуйте с помощью нее на реальные деньги!), а также некоторых мыслей по ее улучшению, как с технической части, так и с продуктовой части.

1 Описание алгоритма

Работа происходит с биржами Binance и FTX. Робот подписывается на обновление цены тикера BTC/USDT (или любого другого тикера, указанного в конфиге) для Binance и для FTX (точнее на обновление order'ов в orderbook'e). Для подписки на обновления order'ов можно использовать открытое API этих бирж или любые готовые библиотеки. Когда best ask на одной бирже становится меньше, чем best bid на другой бирже, робот должен оповестить об этом пользователя (через print, через запись в файл или через любой другой способ оповещения).

2 Требования к реализации

1. Язык программирования

Разработка должна обязательно вестись на Python 3.

2. Конфигурация робота

Робот должен иметь возможность конфигурироваться с помощью файла. Тип файла может быть произвольным, например, `.yaml`, `.json`, `.txt`, `.xml`. Пример конфига в `.yaml`-формате:

```
ticker: BTC/USDT
```

Можно добавить какие-то еще поля на свое усмотрение или изменить структуру файла, если потребуется/захочется. Например, можно добавить параметр `size` - размер, на который входить в сделку (пусть в рамках этого задания открывать реальную сделку и не требуется, но его можно использовать для расчета P&L).

3. Запуск робота

Запуск робота нужно производить через Docker и docker-compose.

4. Poetry

Для создания проекта и управления зависимостями в виртуальном окружении использовать poetry.

5. Улучшения и проблемы

Написать небольшой текст, в котором расписать преимущества и недостатки этой стратегии, этой реализации или любые другие сопутствующие проблемы и возможные решения. Можно расписать как с технической стороны, так и с продуктовой/логической. Максимум 5 пунктов. Несколько примеров:

- (a) Любые арбитражные стратегии должны работать максимально быстро. Лучше переписать код на `c++` или `rust`.
- (b) Комиссии при плейсинге ордеров могут быть гораздо больше, чем профит от сделок.