vc.ru

Михаил Шардин
05.11.2024

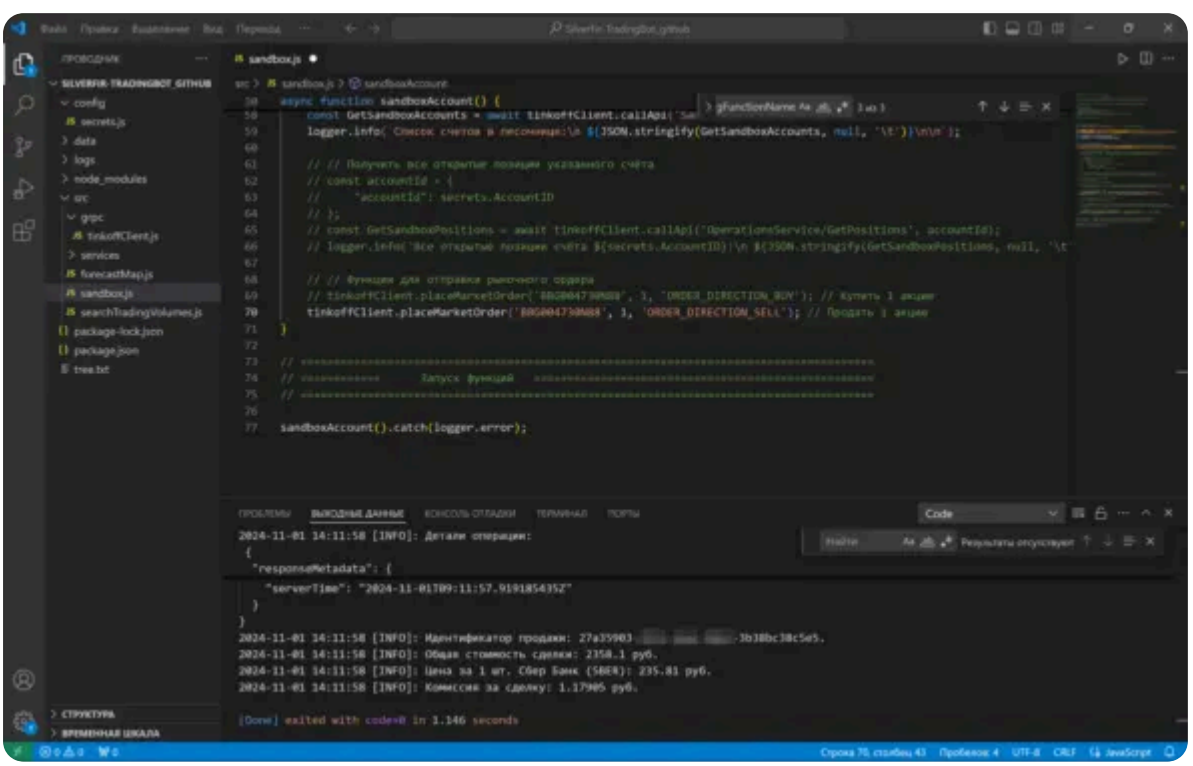
Подписаться

Тестирование торгового робота на Московской бирже в режиме «песочницы»

Перед тем как использовать торгового робота на живых деньгах хочется всё протестировать на демо-счете (или «песочнице»). Это когда программные ошибки не имеет особой стоимости.

Я планирую использовать робота на Московской бирже, через АПИ одного из брокеров. Чтобы частному инвестору начать торговать на бирже нужен брокерский счет. Однако минимальное число российских брокеров имеют свои API (на текущий момент я знаю только ФИНАМ, Алор, Тинькофф Инвестиции). По субъективным причинам я выбрал работать с T-Bank Invest API (это бывший Тинькофф) через среду выполнения JavaScript Node.JS

- Открыть счёт.
- Пополнить баланс счёта рублями через специальный запрос.
- Посмотреть все свои открытые счета в песочнице.
- Купить 1 акцию.
- Продать 1 акцию.
- Получить все открытые позиции указанного счёта.
- Закрыть счёт.



Операция продажи через OrdersService/PostOrder

SilverFir-TradingBot\src\sandbox.js

Этот код Node.js взаимодействует с API Tinkoff Invest, позволяя имитировать торговые операции на виртуальном счете, что позволяет протестировать некоторые функции API в ручном режиме. Вот что делает этот код:

1. Импорт модулей

- secrets: импортирует ключи доступа и идентификаторы из внешнего файла конфигурации (secrets), что помогает защитить

- logger: импортирует модуль ведения журнала, который записывает журналы в файл или консоль. Это важно для отслеживания активности бота и отладки.
- logFunctionName: импортирует утилиту для получения имен функций, что упрощает ведение журнала текущего контекста функции.
- TinkoffClient: импортирует клиентский модуль для взаимодействия с API Tinkoff Invest. Этот клиент обрабатывает запросы к API.

2. Настройка клиента

- API_TOKEN: получает токен API (в режиме песочницы) из внешнего файла конфигурации (secrets) для аутентификации.
- tinkoffClient: создает экземпляр TinkoffClient с токеном песочницы, настраивая связь API для среды песочницы.

3. Функции песочницы

- sandboxAccount(): это основная функция, демонстрирующая различные операции с учетной записью песочницы, с несколькими действиями, которые в настоящее время закомментированы.
- logFunctionName(): регистрирует имя функции в консоли, что полезно для отслеживания в сложных приложениях.
- GetSandboxAccounts: получает все открытые позиции указанного счёта.

Закомментированные операции:

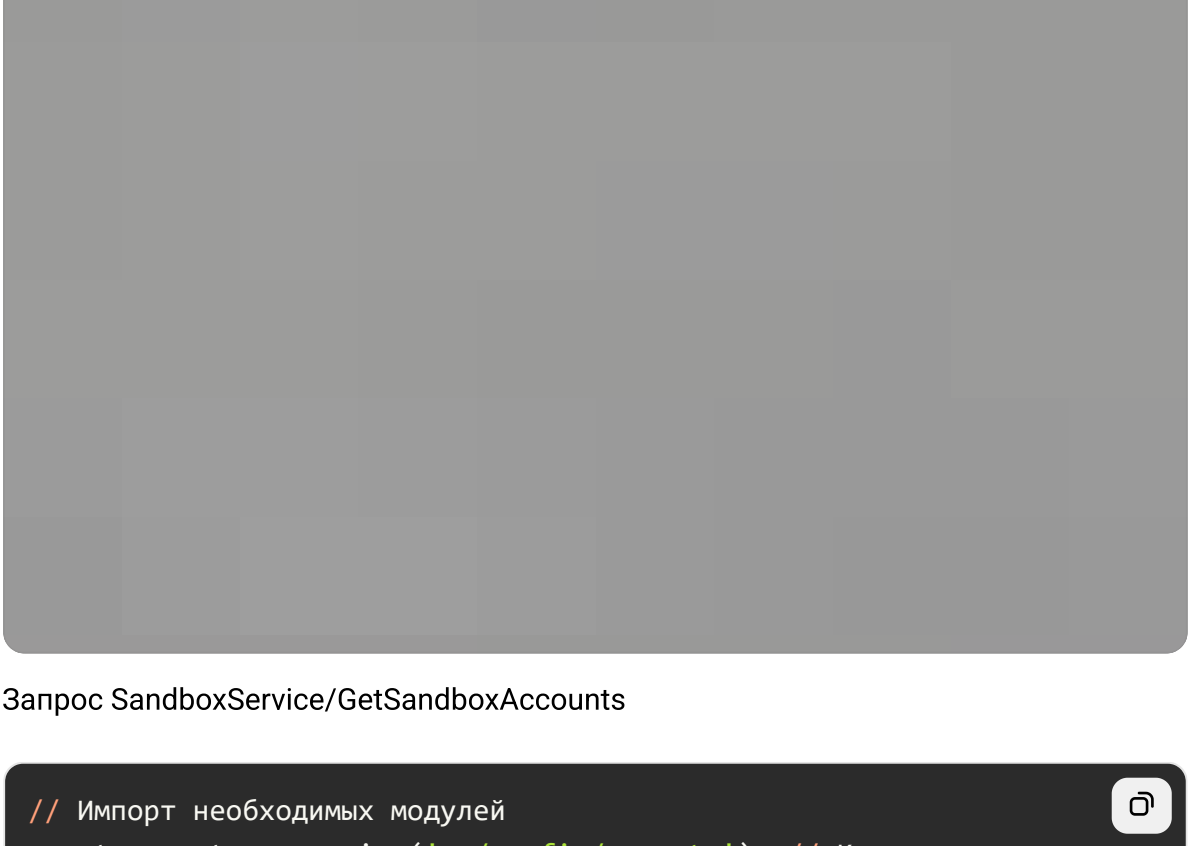
- OpenSandboxAccount: регистрирует новый счет в песочнице, что позволит начать тестирование заново.
- SandboxPayIn: зачисляет средства на счет в песочнице в российских рублях (RUB). Здесь указанная сумма составляет 30 000 руб.
- CloseSandboxAccount: закрывает указанный счет в песочнице, используя его accountId, что позволяет выполнить сброс после тестирования.
- GetSandboxPositions: извлекает и регистрирует все открытые позиции для указанного идентификатора счета.

- placeMarketOrder: отправляет рыночные ордера на покупку и продажу указанного инструмента (здесь BBG004730N88). Это позволит протестировать функциональность размещения ордеров в песочнице.

Ошибки

- sandboxAccount().catch(logger.error): запускает sandboxAccount асинхронно и регистрирует любые обнаруженные ошибки.

Эта структура кода демонстрирует, как взаимодействовать с виртуальным торговым счетом в API Тинькофф. Закомментированные блоки кода указывают на дополнительные функции, которые можно активировать при необходимости, такие как открытие, пополнение и закрытие счетов песочницы, а также размещение ордеров на покупку/продажу.



Запрос SandboxService/GetSandboxAccounts

```
// Импорт необходимых модулей
const secrets = require('../config/secrets'); // Ключи доступа и идент

const logger = require('./services/logService'); // Логирование в файл
```

```
const TinkoffClient = require('./grpc/tinkoffClient'); // модуль для в
const API_TOKEN = secrets.TbankSandboxMode;
const tinkoffClient = new TinkoffClient(API_TOKEN);


async function sandboxAccount() {
    // https://tinkoff.github.io/investAPI/swagger-ui/#/SandboxService
    logger.info(`Запуск функции ${JSON.stringify(logFunctionName())}\n

    // // Регистрации счёта в песочнице
    // const OpenSandboxAccount = await tinkoffClient.callApi('Sandbox
    // logger.info(`Регистрации счёта в песочнице:\n ${JSON.stringify(

    // // Пополнение баланса счёта песочницы
    // const RUB = {
    //     "accountId": secrets.AccountID,
    //     "amount": {
    //         "nano": 0, // Дробная часть отсутствует
    //         "currency": "RUB",
    //         "units": 30000, // Сумма в рублях
    //     }
    // };
    // const SandboxPayIn = await tinkoffClient.callApi('SandboxServic
    // logger.info(`Пополнение баланса счёта песочницы:\n ${JSON.strin

    // // Закрытие счёта в песочнице
    // const accountId = {
    //     "accountId": secrets.AccountID
    // };
    // const CloseSandboxAccount = await tinkoffClient.callApi('Sandbo
    // logger.info(`Закрытие счёта в песочнице:\n ${JSON.stringify(Clo

    // Посмотреть счета в песочнице
    const GetSandboxAccounts = await tinkoffClient.callApi('SandboxSer
    logger.info(`Список счетов в песочнице:\n ${JSON.stringify(GetSand

    // // Получить все открытые позиции указанного счёта
    // const accountId = {
    //     "accountId": secrets.AccountID
    // };
    // const GetSandboxPositions = await tinkoffClient.callApi('Operat
    // logger.info(`Все открытые позиции счёта ${secrets.AccountID}:\n
```

```
    // tinkoffClient.placeMarketOrder('BBG004730N88', 1, 'ORDER_DIRECT
    // tinkoffClient.placeMarketOrder('BBG004730N88', 1, 'ORDER_DIRECT
}

// ================================================================
// ============      Запуск функций   ============================
// ================================================================

sandboxAccount().catch(logger.error);
```

Быстройдействие

Я не ждал какого-то особо быстродействия. Для человека это очень быстро, но вот для робота это медленно. Это придётся учесть при разработке торговой стратегии.

```
[Running] node "d:\Synology ...\SilverFir-TradingBot_github\src\sa
2024-11-01 14:11:57 [INFO]: Запуск функции "sandboxAccount"

2024-11-01 14:11:58 [WARN]: Операция продажи выполнена успешно для Сбе
2024-11-01 14:11:58 [INFO]: Детали операции:
 {
  "orderId": "27a35903-2134-4aaf-XXXX-3b38bc38c5e5",
  "executionReportStatus": "EXECUTION_REPORT_STATUS_FILL",
  "lotsRequested": "1",
  "lotsExecuted": "1",
  "initialOrderPrice": {
    "currency": "rub",
    "units": "2358",
    "nano": 100000000
  },
  "executedOrderPrice": {
    "currency": "rub",
    "units": "235",
    "nano": 810000000
  },
  "totalOrderAmount": {
    "currency": "rub",
    "units": "2358",
    "nano": 100000000
```

```
    "currency": "rub",
    "units": "1",
    "nano": 179050000
  },
  "executedCommission": {
    "currency": "rub",
    "units": "1",
    "nano": 179050000
  },
  "figi": "BBG004730N88",
  "direction": "ORDER_DIRECTION_SELL",
  "initialSecurityPrice": {
    "currency": "rub",
    "units": "235",
    "nano": 810000000
  },
  "orderType": "ORDER_TYPE_MARKET",
  "message": "",
  "initialOrderPricePt": {
    "units": "0",
    "nano": 0
  },
  "instrumentUid": "e6123145-9665-43e0-XXXX-cd61b8aa9b13",
  "orderRequestId": "",
  "responseMetadata": {
    "trackingId": "d059748a138038d3XXXXX93783d61a99",
    "serverTime": "2024-11-01T09:11:57.919185435Z"
  }
}
2024-11-01 14:11:58 [INFO]: Идентификатор продажи: 27a35903-2134-4aaf-
2024-11-01 14:11:58 [INFO]: Общая стоимость сделки: 2358.1 руб.
2024-11-01 14:11:58 [INFO]: Цена за 1 шт. Сбер Банк (SBER): 235.81 руб
2024-11-01 14:11:58 [INFO]: Комиссия за сделку: 1.17905 руб.

[Done] exited with code=0 in 1.146 seconds
```


Для торгового робота 1,146 секунды от отправки ордера до его исполнения можно считать довольно медленным временем.

В высокочастотной торговле (HFT), где компании конкурируют за время исполнения менее миллисекунды, время обработки ордера

основаны на выполнении тысяч сделок за доли секунды, поэтому 1,146 секунды сделают этого робота неконкурентоспособным.

Напротив, для долгосрочной стратегии, такой как дневной торговый бот или свинг-трейдинг, это время может быть приемлемым. Скорость исполнения остается важной, но не такой критической, как в HFT. В этих случаях компромисс часто склоняется в сторону надежности и экономической эффективности, а не чистой скорости. Задержка в 1 секунду, как правило, не подорвет прибыльность в стратегии, где сделки исполняются с интервалом в несколько минут или даже часов.

Я планирую использовать свинг-трейдинг — это торговая стратегия, ориентированная на захват краткосрочных и среднесрочных ценовых движений, обычно удерживая активы в течение нескольких дней или нескольких недель. Цель — извлечь прибыль из «колебаний» цены, используя рыночный импульс, когда цены колеблются в рамках тренда или между уровнями поддержки и сопротивления.

Итоги

Проект полностью представлен на Гитхабе: https://github.com/empenoso/SilverFir-TradingBot. Новые модули будут загружаться по мере написания и тестирования.

Автор: Михаил Шардин

5 ноября 2024 г.

74

Ищем надёжного коллегу в нашу команду. Заполните форму, ответьте на вопросы тестового задания и мы свяжемся с вами в ближайшее время.

7 2 1 +

Начать дискуссию

Комментарий...

GIF

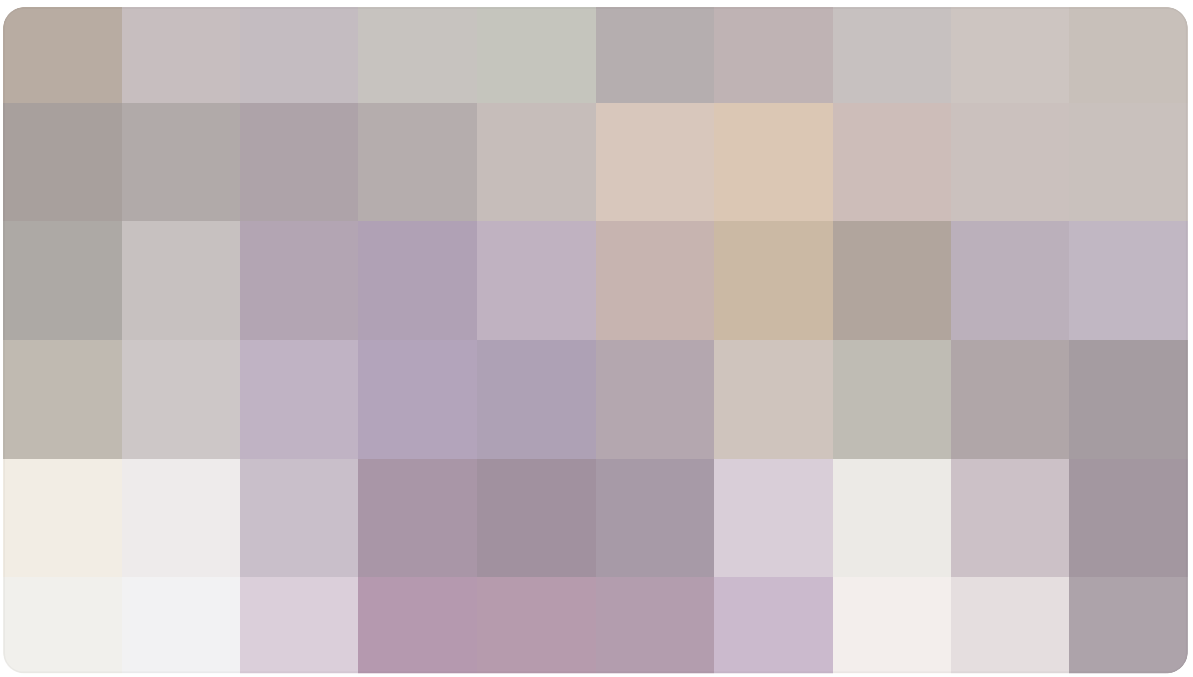
Рекомендации

Артур Томилко
Apple вчера

Подписаться

Bloomberg: из Apple уйдёт старший директор по ИИ Робби Уокер — он отвечал за Siri и работал над новым

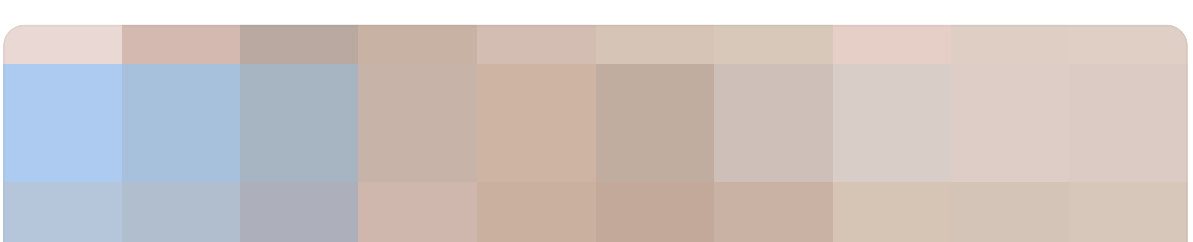
Причины ухода Уокера неизвестны, но они могут быть связаны с задержками внедрения ИИ в продукты компании.

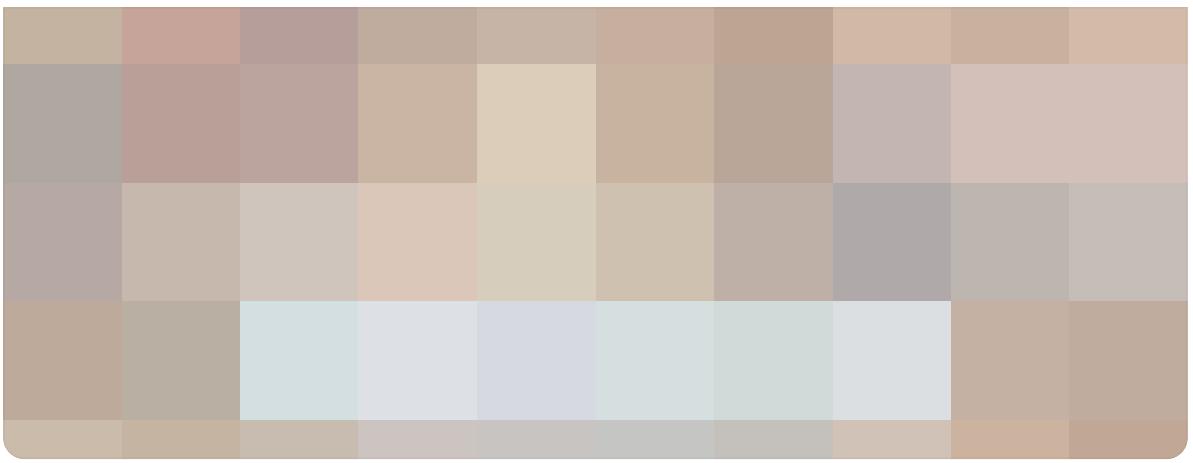


Показать полностью

1 1 +

21 414





10 +

6 3 66K

Михаил Шардин
Инвестиции 26 авг

Подписаться

Как мы строим open-source альтернативу коммерческим платформам алготрейдинга и почему это может изменить рынок

В мире алгоритмической торговли доминируют крупные фонды с их колоссальными ресурсами. Но что, если мы, частные инвесторы и разработчики, можем создать собственный мощный и доступный инструмент? Что, если больше не придётся зависеть от проприетарных платформ или писать с нуля сложную инфраструктуру для тестирования каждой новой идеи?

Сегодня у нас есть Python и такие мощные библиотеки, как Backtrader. Однако голый фреймворк — это лишь половина дела. Чтобы он стал по-настоящему народным инструментом, ему нужна удобная обвязка: готовая структура проекта, автоматический импорт стратегий, наглядные отчёты, тепловые карты для оптимизации и бесшовное подключение к API брокеров — не т...

Показать полностью

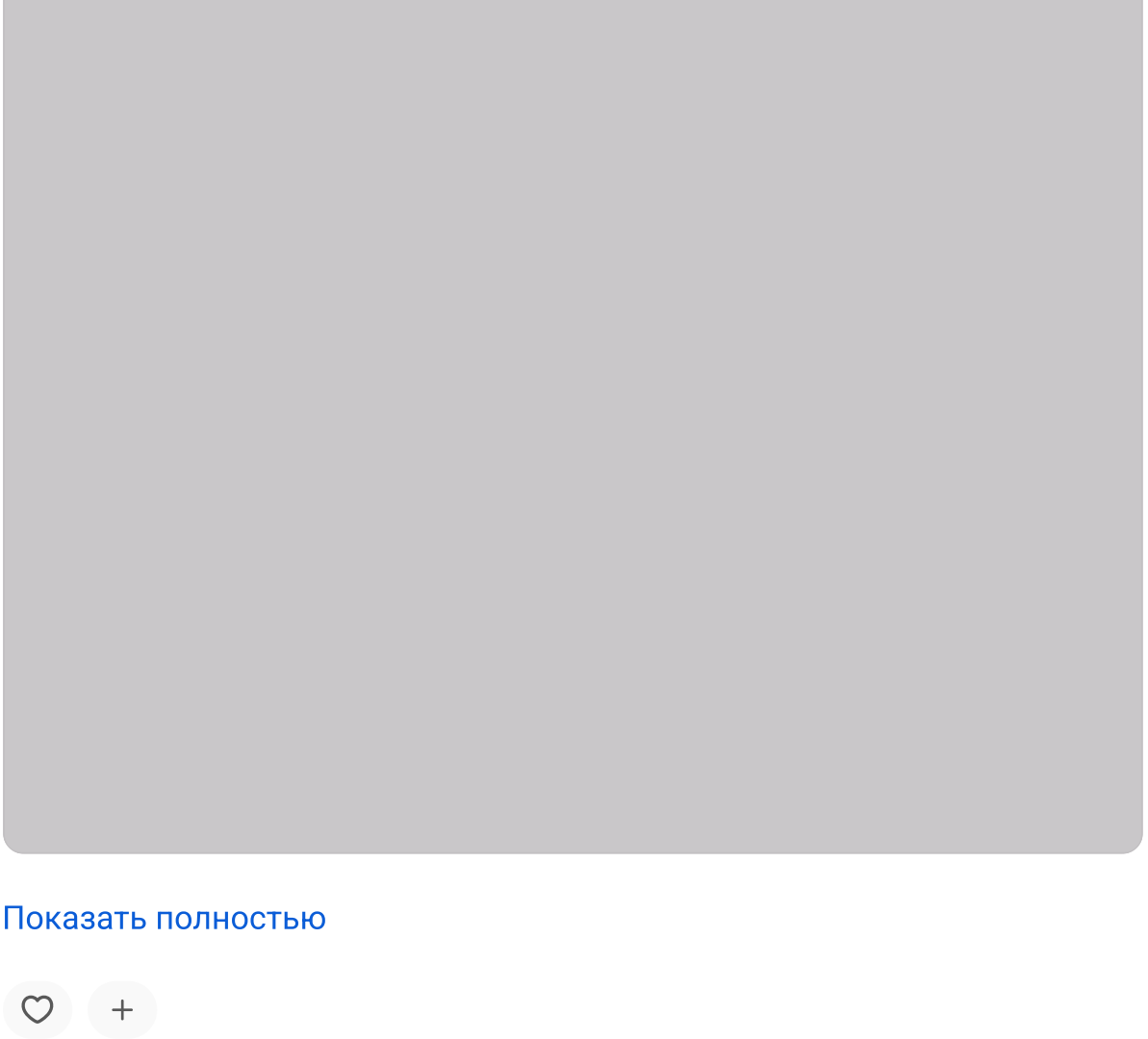
1 157

Макс Корольков
вчера

Подписаться

Схема pay now, buy later в Китае: особенности и риски

Пока во всём мире расцветает BNPL (купи сейчас, плати потом), в Китае всё работает в обратную сторону. Там многие компании из сферы услуг предлагают заплатить наперёд, а пользоваться потом. Эту схему окрестили pay now, buy later 😬



Показать полностью

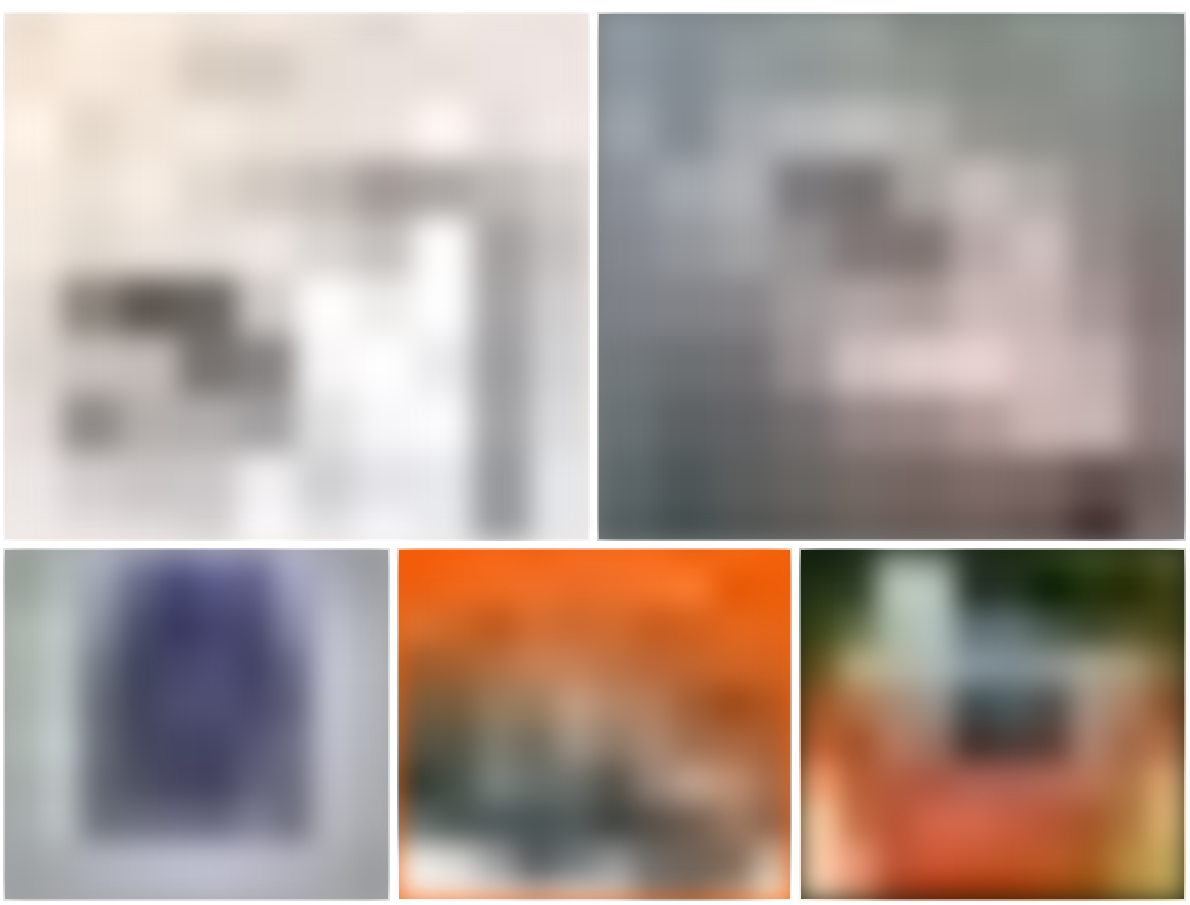
1 1.1K

Maxim Yurin
12 сент

Подписаться

💰 ТОП-5 проектов, которые собрали миллионы на запуск

Люблю поизучать Kickstarter - платформу, где собираются стартапы, которым нужно финансирование. На днях прочитал, как производитель аксессуаров Peak Design открыл сбор денег на выпуск дорожной сумки и получил рекордные $13 млн. Офигеть!



Показать полностью

1 997

MisterBuffett

Как я написал бота для трейдинга на MOEX: уровни BUY и SELL и автоматизация торговли»

Как рассчитать уровни BUY и SELL на бирже

Трейдинг почему-то остаётся хайповой темой только в крипте. Я хочу рассказать про то, как автоматизировал расчёты уровней по акциям и фьючерсам на MOEX и NASDAQ.

Показать полностью

177