

РЕКЛАМА



Путешествие в мир
искусственного
интеллекта

AI помогает **исследовать**
11-13 декабря смотри на AIJ.ru



Реклама. Рекламодатель: ПАО Сбербанк, ИНН 7707083893, ОГРН 102770013219, г. Москва, ул. Вавилова, д. 19. *AIJ, Artificial Intelligence Journey (англ.) - путешествие в мир искусственного интеллекта. Мероприятие AIJ (далее - Мероприятие) проводится 11-13 декабря 2024 года. Участие в Мероприятии AIJ производится на aij.ru. Принять участие могут все желающие участники возрастной категории 12+. Организатор Мероприятия - ПАО Сбербанк (генеральная лицензия Банка России на осуществление банковских операций № 1481 от 11.08.2015).

**empenoso**

8 ноя в 08:33

Инструменты робота, торгующего на Московской бирже через API брокера

**Сложный**

8 мин



1.8K

Open source*, Финансы в IT, JavaScript*, Node.JS*

Кейс

Поскольку хочу использовать для среднесрочной алгоритмической торговли на российском рынке скрипт - робота, то мне необходимо получать от брокера актуальную информацию о текущих ценах и сопутствующую информацию:

- Время работы биржи через `InstrumentsService/TradingSchedules` .
- Основную информацию об инструменте через `InstrumentsService/GetInstrumentBy` .
- Последнюю котировку по инструменту через `MarketDataService/GetLastPrices` .
- Торговые лоты - это определенное количество акций, которые можно купить или продать в рамках одной сделки.
- Свечи по инструменту для разных временных интервалов через `MarketDataService/GetCandles` .
- Технические индикаторы через `MarketDataService/GetTechAnalysis` .
- Понятное имя инструмента через `InstrumentsService/FindInstrument` .

В статье разбираюсь как проделать все эти операции при помощи программного кода.

РЕКЛАМА

**О болях стартапов**

Как избежать их

API от Т-Банк (ранее известный как Тинькофф), работая в среде исполнения JavaScript Node.js.

The screenshot shows a VS Code editor with the file `instruments.js` open. The code defines a function `test` that iterates over a list of stock symbols and calls `tinkoffClient.getName` and `tinkoffClient.getQuote` for each. The terminal output shows the execution of `node "d:\Synology\SilverFir-TradingBot_github\src\instruments.js"`, which runs the `test` function and logs the names and quotes for various stocks like VTBR, MTLR, UMGH, RNFT, EURL, SNGSP, GAZP, ROSN, SBER, SGZK, AFLT, VKCO, RUAL, and TATN. The execution completes with a message: `[Done] exited with code=0 in 2.793 seconds`.

Время запроса почти 3 секунды - это много

SilverFir-TradingBot\src\instruments.js

Этот модуль служит для проверки части функций, которые будут использоваться потом в автоматическом режиме. Что он делает? Импортирует необходимые модули:

- `secrets` и `config` для конфиденциальной информации и настроек конфигурации.
- Службы для рисования диаграмм (`chart`), обработки CSV-файлов (`csvHandler`), решений о покупке/продаже (`buyDecision` и `sellDecision`) и расчета доходности (`yieldCalculator`).
- Служба ведения журнала (`logger`) для отслеживания действий и ошибок.
- `TinkoffClient` , модуль для взаимодействия с Tinkoff Invest API, и `API_TOKEN` для аутентификации.

Основные функции



О болях стартапов

Как избежать их

Цель: Тестирование функциональности API и регистрация данных для конкретных биржевых инструментов.

Примеры операций:

- Получить основную информацию об инструменте - вызывает `InstrumentsService/GetInstrumentBy` для получения информации о определенном инструменте с использованием его идентификатора.
- Получить список всех акций - вызывает `InstrumentsService/Shares` для составления списка акций и регистрации первых нескольких результатов.

Функция `instruments()` :

Цель: Основная функция для извлечения данных и подготовки к торговле.

Примеры операций:

- Получение времени работы биржи - получает и регистрирует часы торговли.
- Найти всю информацию об акциях в списке файла config - отображает всю информацию о каждом из тикеров в JSON формате.
- Последние цены и торговые лоты - извлекает последние цены акций и проверяет размеры лотов (это определенное количество акций, которые можно купить или продать в рамках одной сделки).
- Данные свечей - собирает данные свечей (ценовые точки с течением времени) в различные интервалы (5 минут, час, день).
- Технические индикаторы - извлекает индикаторы, такие как SMA (простая скользящая средняя), для анализа тенденций акций. По выходным данным нет, хотя свечи за это же время присутствуют.
- Разместить рыночный ордер - строки кода для прямого размещения ордеров на покупку/продажу.
- Позиции портфеля - перечисляет текущие активы и вычисляет годовую доходность.

В конце код запускает `test()` и `instruments()` с обработкой ошибок, регистрируя все возникшие проблемы.



О болях стартапов

Как избежать их

отслеживать и взаимодействовать с платформой Tinkoff, обрабатывая все: от анализа цен

акций и тенденций до размещения сделок. Настройка этого бота подходит для среднесрочной торговли на основе данных, используя Node.js для быстрой обработки данных и взаимодействия с API.

```
// Импорт необходимых модулей
const secrets = require('../config/secrets'); // Ключи доступа и идентификаторы
const config = require('../config/config'); // Параметры
const chart = require('../services/chartService'); // Отрисовка графиков
const csvHandler = require('../services/csvHandler'); // Работа с CSV файлами
const buyDecision = require('../services/buyDecision'); // Функции покупки
const sellDecision = require('../services/sellDecision'); // Функции продажи
const yieldCalculator = require('../services/yieldCalculator'); // Расчёт годовой доходности

const logger = require('../services/logService'); // Логирование в файл и консоль
const logFunctionName = require('../services/logFunctionName'); // Получение имени функции

const TinkoffClient = require('../grpc/tinkoffClient'); // модуль для взаимодействия с API
const API_TOKEN = secrets.TbankSandboxMode;
const tinkoffClient = new TinkoffClient(API_TOKEN);

async function test() {
  logger.info(`Запуск функции ${JSON.stringify(logFunctionName())}\n`);

  // // Получить основную информацию об инструменте InstrumentsService/GetInstrumentBy
  // const testPayload = {
  //   idType: "INSTRUMENT_ID_TYPE_FIGI", // Тип идентификатора INSTRUMENT_ID_TYPE_
  //   id: "BBG004730N88" // Идентификатор инструмента
  // };
  // const response = await tinkoffClient.callApi('InstrumentsService/GetInstrumentBy
  // logger.info(`InstrumentsService/GetForecastBy: ${JSON.stringify(response, null,

  // // Получить список акций InstrumentsService/Shares
  // const testPayload = {
  //   "instrumentStatus": "INSTRUMENT_STATUS_BASE", // https://russianinvestments.
  //   "instrumentExchange": "INSTRUMENT_EXCHANGE_UNSPECIFIED"
  // };
  // const response = await tinkoffClient.callApi('InstrumentsService/Shares', testPa
  // // Отображение ответа от API
  // logger.info(`Ответ: ${JSON.stringify(response, null, 2)}\n`); // выводится только
```



О болях стартапов

Как избежать их

```

async function instruments() {
    logger.info(`Запуск функции ${JSON.stringify(logFunctionName())}\n`);

    // // Получение времени работы биржи
    // const response = await tinkoffClient.callApi('InstrumentsService/TradingSchedule
    // logger.info(`Получение времени работы биржи: ${JSON.stringify(response, null, 2)}
    // await tinkoffClient.getExchangeOpen();

    // // Найти всю информацию об акциях в списке файла config
    // for (const stock of config.securitiesToMonitorTikerArray) { // securitiesToMonit
    //     const securitiesToMonitorTikerArrayPayload = {
    //         "query": stock,
    //         "instrumentKind": "INSTRUMENT_TYPE_SHARE"
    //     };
    //     try {
    //         const FindInstrument = await tinkoffClient.callApi('InstrumentsService/F
    //         logger.info(`Ищем тикер ${stock}: \n${JSON.stringify(FindInstrument, null, 2)}
    //     } catch (error) {
    //         logger.error(`Ошибка ${stock}:`, error.message);
    //     }
    // }

    // // Получить последнюю цену для акций из списка в файле config
    // for (const stock of config.securitiesToMonitorFigiArray) {
    //     try {
    //         const quote = await tinkoffClient.getQuote(stock);
    //         const name = await tinkoffClient.getName(stock);
    //         logger.info(`Цена акции ${name.nameCombination} [${stock}]: ${quote} руб
    //     } catch (error) {
    //         logger.error(`Ошибка ${stock}:`, error.message);
    //     }
    // }

    // // Получение торговых лотов - это определенное количество акций, которые можно к
    // for (const stock of config.securitiesToMonitorFigiArray) {
    //     try {
    //         const quote = await tinkoffClient.getLot(stock);
    //         const name = await tinkoffClient.getName(stock);
    //         logger.info(`Торговый лот акции ${name.nameCombination} [${stock}] = ${q
    //     } catch (error) {

```



О болях стартапов

Как избежать их

```
// Получение понятного имени инструмента
for (const stock of config.securitiesToMonitorFigiArray) {
  try {
    const name = await tinkoffClient.getName(stock);
    const nameUid = name.uid;
    logger.info(`${name.nameCombination} это ${stock} или ${nameUid}.`);
  } catch (error) {
    logger.error(`Ошибка ${stock}:`, error.message);
  }
}

// // Тест корректности размера лотов:
// const figi = 'BBG004730N88'; // Пример ФИГИ
// const price = await tinkoffClient.getQuote(figi);
// const quantity = await config.getPurchaseQuantity(price, figi);
// logger.info(`Тест количества лотов ${figi} для покупки: ${quantity}`);

// // Получение свечей по инструменту
// for (const stock of config.securitiesToMonitorFigiArray) {
//   try {
//     const name = await tinkoffClient.getName(stock);
//     const candles5Min = await tinkoffClient.getCandles(stock, "CANDLE_INTERVAL_5_MINUTES");
//     logger.info(`5-минутные свечи для ${name.nameCombination}: ${JSON.stringify(candles5Min)}`);
//     const candlesHour = await tinkoffClient.getCandles(stock, "CANDLE_INTERVAL_1_HOUR");
//     logger.info(`Часовые свечи для ${name.nameCombination}: ${JSON.stringify(candlesHour)}`);
//     const candlesDay = await tinkoffClient.getCandles(stock, "CANDLE_INTERVAL_1_DAY");
//     logger.info(`Дневные свечи для ${name.nameCombination}: ${JSON.stringify(candlesDay)}`);
//   } catch (error) {
//     logger.error(`Ошибка ${stock}:`, error.message);
//   }
// }

// // Получение технических индикаторов по инструменту
// for (const stock of config.securitiesToMonitorFigiArray) {
//   try {
//     const instrument = await tinkoffClient.getName(stock);
//     const instrumentUid = instrument.uid;
//     const indicatorType = "INDICATOR_TYPE_SMA"; // Пример типа индикатора (SMA)
//     const interval = "INDICATOR_INTERVAL_FIVE_MINUTES"; // Пример интервала
```



О болях стартапов

Как избежать их

```

//      } catch (error) {
//          logger.error(`Ошибка ${stock}: ${error.message}`);
//      }
// }

// // Создание графиков пересечения свечей и индикатора для акций из списка в файле
// for (const stock of config.securitiesToMonitorFigiArray) {
//     try {
//         const charts = chart.generateCandlestickChart(stock);
//     } catch (error) {
//         logger.error(`Ошибка ${stock}:`, error.message);
//     }
// }

// // Функция для отправки рыночного ордера
// tinkoffClient.placeMarketOrder('BBG004730N88', 1, 'ORDER_DIRECTION_BUY'); // Куп
// tinkoffClient.placeMarketOrder('BBG004730N88', 1, 'ORDER_DIRECTION_SELL'); // Пр

// // Получить все открытые позиции счёта
// const GetSandboxPositions = await tinkoffClient.getPortfolio();
// logger.info(`Все открытые позиции счёта ${secrets.AccountID}: \n ${JSON.stringify

// // Расчёт годовой доходности от Торгового робота
// const SilverFirBotYield = await yieldCalculator.calculateAnnualYield();
// logger.info(`Годовая доходность от Торгового робота SilverFir Bot: ${SilverFirBo

// // Получить прогнозов инвестдомов по инструменту InstrumentsService/GetForecastB
// const ForecastPayload = {
//     "instrumentId": "1c69e020-f3b1-455c-affa-45f8b8049234" // У Аэрофлот (AFLT),
// };
// const response = await tinkoffClient.callApi('InstrumentsService/GetForecastBy',
// logger.info(`InstrumentsService/GetForecastBy: ${JSON.stringify(response, null,

}

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

```

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



О болях стартапов

Как избежать их

Многие строки закомментированы, но это не потому что они не рабочие, а потому что они используются для тестов той или иной функции.

Итоги

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

Автор: [Михаил Шардин](#)

11 ноября 2024 г.

Теги: бот, московская биржа, мосбиржа, moexalgo, tbank, T-Bank Invest API

Хэбы: [Open source](#), [Финансы в IT](#), [JavaScript](#), [Node.JS](#)

Редакторский дайджест

Присылаем лучшие статьи раз в месяц

**170**

Карма

80.1

Рейтинг

Михаил Шардин @empenoso

Разработчик

[Подписаться](#)

[Сайт](#) [Сайт](#) [Github](#) [Telegram](#)

РЕКЛАМА

СТОКМАНН**%****РАСПРОДАЖА***Реклама. АО "Стокманн", ОГРН 1037717018228, Москва, Краснопресненская наб., 12, офис 1001, stockmann.ru**О болях стартапов**

Как избежать их

Публикации

ЛУЧШИЕ ЗА СУТКИ

ПОХОЖИЕ



anatolii-potapov

17 часов назад

T-Lite и T-Pro – открытые русскоязычные опенсорс-модели на 7 и на 32 млрд параметров



Сложный



9 мин



13K

Репортаж



+118



80



27



CyberPaul

17 часов назад

Советский одноплатник. Уникальная ЭВМ «Электроника С5»



Простой



7 мин



4.9K

Ретроспектива



+53



22



37



PatientZero

18 часов назад

Реверс-инжиниринг формата данных кабельного канала Sega



Простой



9 мин



2.1K

Обзор

Перевод



+38



20



1



ru_vds

12 часов назад

Прошивки OpenWrt: атака на цепочку поставок



О болях стартапов

Как избежать их

 +36 22 1**melnik909**

16 часов назад

Готовимся к вопросам по вёрстке на интервью Frontend-разработчика: «Какие знаешь псевдо-классы?»



Средний



9 мин



1.9K

 Обзор +29 33 2**breakmirrors**

15 часов назад

Не нажимайте эту кнопку: почему макросы Office все еще опасны



Средний



9 мин



2.6K

 Обзор +28 24 4**GeorgKDeft**

21 час назад

Обогрев при помощи ветряка без электричества



Простой



14 мин



18K

 Аналитика Перевод +28 33 71**spring_aio**

14 часов назад

Горькая правда о программировании с использованием ИИ



Простой



11 мин



3.7K

 Обзор Перевод**О болях стартапов**

Как избежать их

**PeterZaidel**

18 часов назад

ACM RecSys — 2024: тренды и доклады с крупнейшей конференции по ML в рекомендательных системах



Средний



17 мин



542

Обзор



+23



9



0

**FirstJohn**

17 часов назад

Домашние эксперименты с радиолампами. Часть 2. Практика



10 мин



1.1K



+20



15



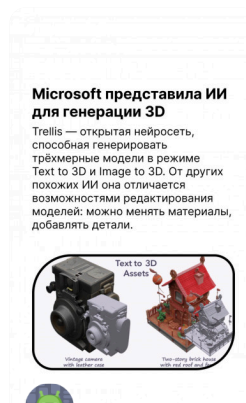
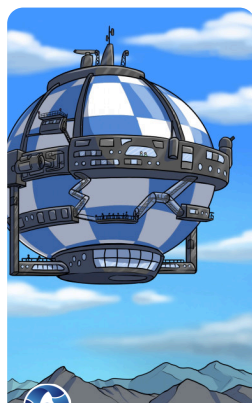
40

О чем поговорить с легендарным инженером на ИТ-конференции мечты?

Опрос

Показать еще

ИСТОРИИ



О болях стартапов

Как избежать их

Тестировщиков

аналитического

мира AI

представила ии для

компаний

bala

зОВЕТ

ПОЛЕТА

ГЕНЕРАЦИИ 3D

ВАКАНСИИ

NodeJS Бэкэнд разработчик

от 140 000 до 220 000 Р · Complex Cloud Solutions · Москва · Можно удаленно

.Net Developer Remote

от 5 000 \$ · Freuders · Можно удаленно

JavaScript FullStack разработчик

от 80 000 до 130 000 Р · Rocket · Смоленск · Можно удаленно

Frontend-разработчик JavaScript

от 40 000 до 90 000 Р · div. · Ставрополь

JavaScript Fullstack

от 2 000 до 7 000 \$ · OnClass · Можно удаленно

[Больше вакансий на Хабр Карьере](#)

МИНУТОЧКУ ВНИМАНИЯ



Кто завоевал сердца
айтишников в 2024 году



Конференции, хакатоны,
митапы: что посетить в декабре



Боли стартапов и акселерация:
выводим MVP на рынок

РАБОТА

JavaScript разработчик

152 вакансии



О болях стартапов

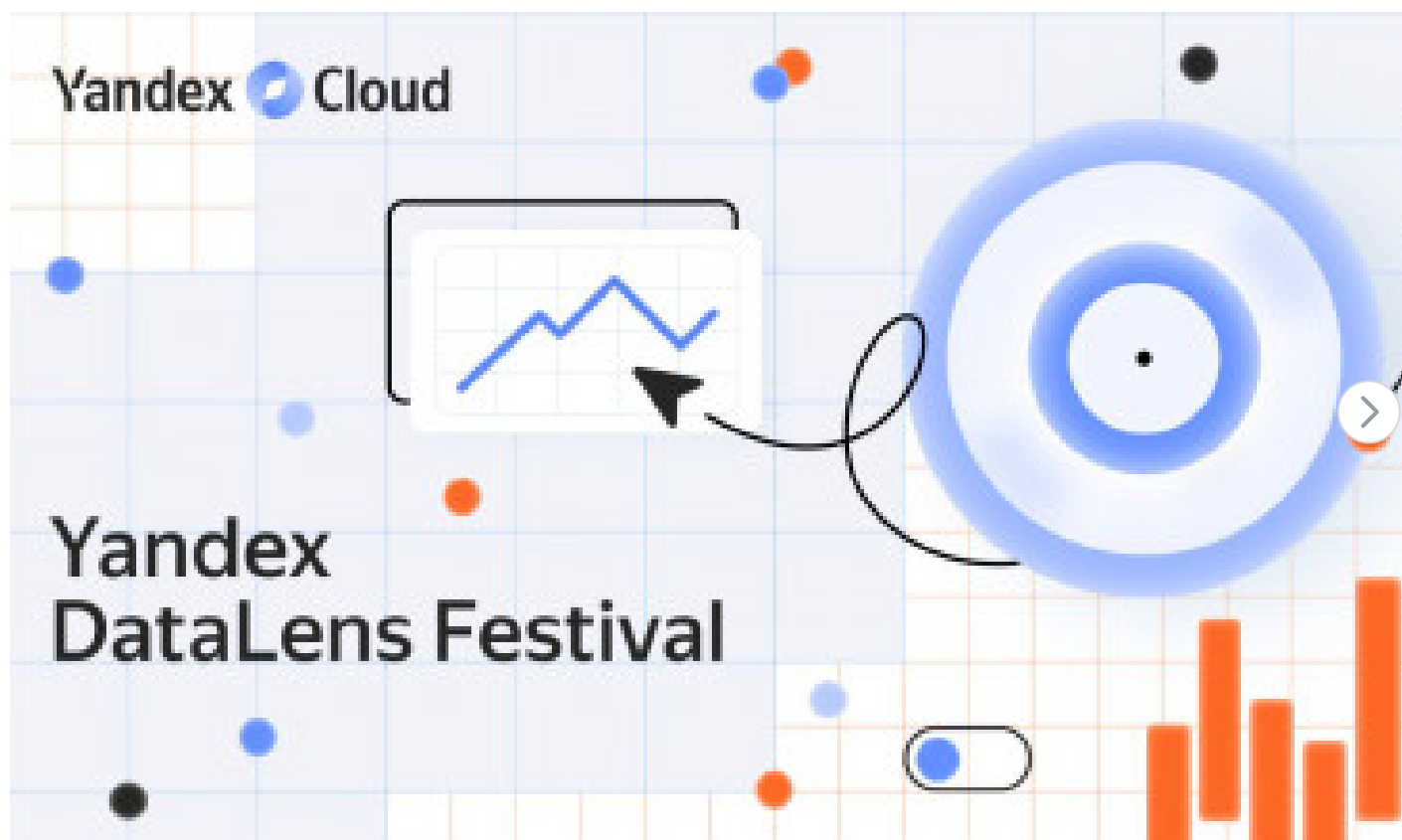
Как избежать их

[Node.js разработчик](#)

56 вакансий

[Все вакансии](#)

БЛИЖАЙШИЕ СОБЫТИЯ



2 – 18 декабря

Yandex DataLens Festival 2024

Москва • Онлайн

Разработка

Менеджмент

Аналитика

[Больше событий в календаре](#)**О болях стартапов**

Как избежать их



🌐 Настройка языка

Техническая поддержка

© 2006–2024, Habr



О болях стартапов

Как избежать их