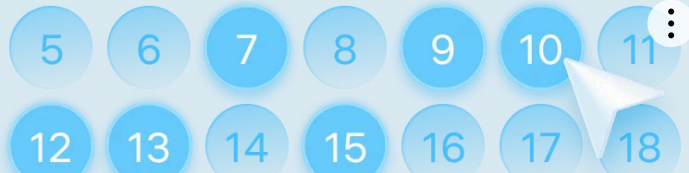




Все важнейшие
IT-события здесь

**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.

Частному лицу для начала торговли на бирже частному инвестору необходим брокерский

РЕКЛАМА

**Реакции уже на Хабре**

Разблокируй синих слизней

Банк (ранее известный как тинькофф), работая в среде исполнения javascript node.js.

```

// Проверка наличия торгующих инструментов (instrument) в базе
// } catch (error) {
//   logger.error('Ошибка ${stock}:', error.message);
// }
// }

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

// // Тест корректности размера лотов:
// const figi = 'BBG004730N88'; // Пример ФИГИ
// const price = await tinkoffClient.getQuote(figi);

[Running] node "d:\Synology\SilverFir-TradingBot\src\instruments.js"
2024-11-08 10:11:27 [INFO]: Запуск функции "test"

2024-11-08 10:11:27 [INFO]: Запуск функции "instruments"

2024-11-08 10:11:27 [INFO]: Банк БТБ (VTBR) это BBG004730ZJ9 или 8e2b0325-0292-4654-8a18-4f63ed3b0e09.
2024-11-08 10:11:27 [INFO]: Мечел (MTLR) это BBG004568598 или eb4ba863-e85f-4f80-8c29-f2627938ee58.
2024-11-08 10:11:27 [INFO]: ОБК (UWGN) это BBG000H03V85 или a8dc188b-cae8-40c0-99e8-f561f4339751.
2024-11-08 10:11:28 [INFO]: РуссНефть (RNFT) это BBG00F9XX7H4 или c7485564-ed92-45fd-a724-1214aa202904.
2024-11-08 10:11:28 [INFO]: ЕвроТранс (EUTR) это TCS00A100ZV2 или 02b2ea14-3c4b-47e8-9548-45a8dbcc8f8a.
2024-11-08 10:11:28 [INFO]: Сургутнефтегаз - привилегированные акции (SNGSP) это BBG0045681M2 или a797f14a-8513-4b84-b15e-a3b98dc4cc00.
2024-11-08 10:11:28 [INFO]: Газпром (GAZP) это BBG004730RPE или 962e2a95-02a9-4171-abd7-aa198db643a.
2024-11-08 10:11:28 [INFO]: Роснефть (ROSN) это BBG004731354 или fd417230-19cf-4e7b-9623-f7c9ca18ec6b.
2024-11-08 10:11:28 [INFO]: Сбер Банк (SBER) это BBG004730N88 или e6123145-9665-43e0-8413-cd61b8aa9b13.
2024-11-08 10:11:29 [INFO]: Сегежа (SGZH) это BBG0100R9963 или 7bedd86b-478d-4742-a28c-29d27f8dbc7d.
2024-11-08 10:11:29 [INFO]: Аэрофлот (AFLT) это BBG0045683M7 или 1c69e020-f3b1-455c-af6f-5f9073e0d1f5.
2024-11-08 10:11:29 [INFO]: ВК (VKCO) это TCS00A106VF0 или b71bd174-c72c-41b0-a66f-5f9073e0d1f5.
2024-11-08 10:11:29 [INFO]: РУСАЛ (RUAL) это BBG008F2T3T2 или f866872b-8f68-4b6e-930f-749fe9aa79c0.
2024-11-08 10:11:29 [INFO]: Татнефть (TATN) это BBG004RVFFC0 или 88468f6c-c67a-4fb4-a006-53eed803883c.

[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

Функция `test()` :

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



Реакции уже на Хабре

Разблокируй синих слизней

- 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 данных 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.

Файл `instruments.js` это ещё одна часть бота, которая позволяет частному инвестору 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.



Реакции уже на Хабре

Разблокируй синих слизней

```
// 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_F
  //   id: "BBG004730N88" // Идентификатор инструмента
  // };
  // const response = await tinkoffClient.callApi('InstrumentsService/GetInstrumentBy'
  // logger.info(`InstrumentsService/GetForecastBy: ${JSON.stringify(response, null, 2)}`);

  // // Получить список акций InstrumentsService/Shares
  // const testPayload = {
  //   "instrumentStatus": "INSTRUMENT_STATUS_BASE", // https://russianinvestments.g
  //   "instrumentExchange": "INSTRUMENT_EXCHANGE_UNSPECIFIED"
  // };
  // const response = await tinkoffClient.callApi('InstrumentsService/Shares', testPay
  // // Отображение ответа от API
  // logger.info(`Ответ: ${JSON.stringify(response, null, 2)}`); // выводится только 3

}

async function instruments() {
  logger.info(`Запуск функции ${JSON.stringify(logFunctionName())}\n`);
  // const response = await tinkoffClient.callApi('InstrumentsService/TradingSchedules
```



Реакции уже на Хабре

Разблокируй синих слизней

```
// 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) { // securitiesToMonitor
//     const securitiesToMonitorTikerArrayPayload = {
//         "query": stock,
//         "instrumentKind": "INSTRUMENT_TYPE_SHARE"
//     };
//     try {
//         const FindInstrument = await tinkoffClient.callApi('InstrumentsService/FindInstrument', securitiesToMonitorTikerArrayPayload);
//         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 купить
// 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}] = ${quote} шт.`);
//     } catch (error) {
//         logger.error(`Ошибка ${stock}:`, error.message);
//     }
// }

// Получение понятного имени инструмента
```

**Реакции уже на Хабре**

Разблокируй синих слизней

```
        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 (5 минут)
//         const typeOfPrice = "TYPE_OF_PRICE_CLOSE"; // Тип цены (например, закрытия)
//         const indicators = await tinkoffClient.getTechIndicators(instrumentUid, indicatorType, interval, typeOfPrice);
//         logger.info(`Индикатор ${indicatorType} для ${instrument.nameCombination}: ${JSON.stringify(indicators)}`);
//     } catch (error) {
//         logger.error(`Ошибка ${stock}: ${error.message}`);
//     }
// }
```

**Реакции уже на Хабре**

Разблокируй синих слизней

```

// // 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'); // Купи
// tinkoffClient.placeMarketOrder('BBG004730N88', 1, 'ORDER_DIRECTION_SELL'); // Про

// // 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: ${SilverFirBot

// // Получить прогнозов инвестдомов по инструменту InstrumentsService/GetForecastBy
// 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, 2

}

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

test().catch(logger.error);
instruments().catch(err => logger.error(err));

```

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



Реакции уже на Хабре

Разблокируй синих слизней

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

Хабы: Open source, Финансы в IT, JavaScript, Node.JS

Редакторский дайджест

Присылаем лучшие статьи раз в месяц

**176**

Карма

135.2

Рейтинг

Михаил Шардин @empenoso

Разработчик

Подписаться



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

РЕКЛАМА

FURLA**СТОКМАНН**

*Реклама. АО "Стокманн", ОГРН 1037717018228, Москва,
Краснопресненская наб., 12, офис 1001, stockmann.ru

**Реакции уже на Хабре**

Разблокируй синих слизней

Публикации

[ЛУЧШИЕ ЗА СУТКИ](#)[ПОХОЖИЕ](#)**Devastator82**

14 часов назад

Заметки попаданца о Руанде 2000-х



13 мин



5.5K

Из песочницы

**+98**

24



14

**ru_vds**

21 час назад

Как долго можно извлечь данные из RAM после отключения компьютера?



Средний



11 мин



11K

Обзор

Перевод

**+53**

50



14

**Loxmatiyamamont**

23 часа назад

Будущее PostgreSQL: как 64-битный счетчик транзакций решает проблему масштабирования



Средний



7 мин



5K

**+37**

34



14

**Finam_Broker**

22 часа назад

\$20 000 000 для хикикомори: история успеха Такаши Котегавы, подловившего банк Mizuho на ошибке

**Реакции уже на Хабре**

Разблокируй синих слизней

 +33 14 3**techno_mot**

22 часа назад

Самые дерзкие фишинговые схемы 2024 года, которые сработали



Простой



10 мин



8.7K

Обзор

 +29 19 6**gserge**

4 часа назад

Менеджмент менеджмента: как во «Фланте» внедрили принятие решений эфемерными рабочими группами



Простой



7 мин



545

 +26 5 0**kuksepa**

23 часа назад

AWX: упрощаем работу с Ansible



Простой



12 мин



3K

 +24 36 1**OldFashionedEngineer**

2 часа назад

Защита схемы от переполюсовки, что может быть проще?



Средний



15 мин



1.3K

Тutorial

 +22 19 10**Реакции уже на Хабре**

Разблокируй синих слизней

«Как не надо проектировать ракеты» или путь разработки ракеты на ЖРД. Часть 4 — Как не сделать парашют



Простой



9 мин



2.1K

Тutorial

+17

13

1



GordienkoAnd

2 часа назад

Как настраивать сети: выделенный и облачный серверы



Средний



21 мин



4.1K

Обзор

+16

9

0

Получи ответ на свой вопрос в сфере патентования

Турбо

Показать еще

ИСТОРИИ



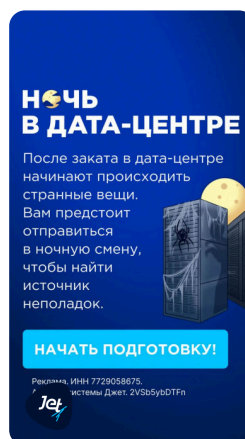
Новый хаб для



Если Elixir — твоя



Спасибо, КЭП



Неполадки в



Космическая



Магия



Реакции уже на Хабре

Разблокируй синих слизней

ЗАКАЗЫ

Telegram Бот для Просмотра видео за вознаграждение

8000 руб./за проект · 1 отклик · 21 просмотр

Разработатть скрипты на TS для игрового сервера GTA 5

60000 руб./за проект · 1 отклик · 18 просмотров

Разработка на Angular

1500 руб./в час · 6 откликов · 41 просмотр

Доработать функционал сайта

3000 руб./за проект · 10 откликов · 77 просмотров

Редизайн карточки товара сайта на Битрикс

12000 руб./за проект · 6 откликов · 61 просмотр

[Больше заказов на Хабр Фрилансе](#)

МИНУТОЧКУ ВНИМАНИЯ



Опрос

Исследуем новые миры: Хабр и ЭКОПСИ изучают IT-рынок РБ



Турбо

Карманная Лlama: запускаем языковую модель в браузере



Интересно

Конференции, хакатоны, митапы: что посетить в декабре

РАБОТА

[JavaScript разработчик](#)

146 вакансий

[Node.js разработчик](#)

55 вакансий



Реакции уже на Хабре

Разблокируй синих слизней

41 вакансия

[Все вакансии](#)

БЛИЖАЙШИЕ СОБЫТИЯ



25 – 26 апреля

IT-конференция Merge Tatarstan 2025

Казань

Разработка

Маркетинг

Другое

[Больше событий в календаре](#)

Хабр



Реакции уже на Хабре

Разблокируй синих слизней

🌐 Настройка языка

Техническая поддержка

© 2006–2024, Habr



Реакции уже на Хабре

Разблокируй синих слизней