

Хабр



КАК СТАТЬ АВТОРОМ



Бэкендеры, вам сюда



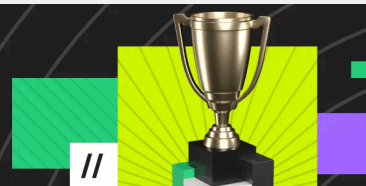
Войти

РЕКЛАМА

cloud.ru

Облачный IT-турнир

Участвуй и докажи свой уровень



empenoso

8 ноя в 08:33

## Инструменты робота, торгующего на Московской бирже через API брокера



Сложный



8 мин



55

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



Облачный IT-турнир

Участвуй и получай призы

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

```

101 // } catch (error) {
102 //   logger.error('Ошибка ${stock}:', error.message);
103 // }
104 // }
105
106 // Получение понятного имени инструмента
107 for (const stock of config.securitiesToMonitorfigiArray) {
108   try {
109     const name = await tinkoffClient.getName(stock);
110     const nameUid = name.uid;
111     logger.info(`${name.nameCombination} это ${stock} или ${nameUid}.`);
112   } catch (error) {
113     logger.error('Ошибка ${stock}:', error.message);
114   }
115 }
116
117 // // Тест корректности размера лотов:
118 // const figi = 'BBG004730N88'; // Пример ФИГИ
119 // const price = await tinkoffClient.getQuote(figi);

```

```

[Running] node "d:\Synology
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 ВТБ (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 (UMGN) это BBG0008ND3V85 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 TCS00A1002V2 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 BBG004730RP0 или 962e2a95-02a9-4171-abd7-aa198dbe643a.
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-affa-45f8b0049234.
2024-11-08 10:11:29 [INFO]: ВК (VKCO) это TCS00A100VFB или b71bd174-c72c-41b0-a66f-5f9073beddfe.
2024-11-08 10:11:29 [INFO]: РУСАЛ (RUAL) это BBG000F2T3T2 или f866872b-8f68-4b6e-930f-740f0e0a79c0.
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



**Облачный IT-турнир**

Участвуй и получай призы

**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.



**Облачный IT-турнир**

Участвуй и получай призы

отслеживать и взаимодействовать с платформой tinker, 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
```



**Облачный IT-турнир**

Участвуй и получай призы

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

**Облачный IT-турнир**

Участвуй и получай призы

```
// 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
```

**Облачный IT-турнир**

Участвуй и получай призы

```

//      } 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);

```



**Облачный IT-турнир**

Участвуй и получай призы

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](#)

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

## Редакторский дайджест

Присылаем лучшие статьи раз в месяц

**165**

Карма

**33**

Рейтинг

**Михаил Шардин** [@empenoso](#)

Разработчик

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

РЕКЛАМА

citilink.ru

Плати частями в Сплит

**82 590 ₽**

Плати частями в Сплит

**64 999 ₽****Облачный IT-турнир**

Участвуй и получай призы



## Публикации

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

19 часов назад

### Ликуй, разработчик — всего 14 лет и QNX снова открыли

4 мин

11K

+56

32

20

**AKlimenkov**

12 часов назад

### Мои простые правила хорошего текста: личное руководство по стилю

5 мин

1.7K

+43

49

14

**Konstantin-2**

16 часов назад

### Гексагональная сетка для игр с круглой Землей

Средний

7 мин

2.8K

Тutorial

+42

37

5

**GeeksCat**

19 часов назад

### Как создавалась антология Риддика

Простой

27 мин

3.4K

Ретроспектива

**Облачный IT-турнир**

Участвуй и получай призы

**Bright\_Translate**

20 часов назад

## Почему программное обеспечение движется только вперёд

**Простой**

6 мин



4K

Обзор

Перевод

**+28**

36



3

**dalerank**

10 часов назад

## Тяжелый H[header]

**Простой**

16 мин



2K

**+25**

17



13

**Finam\_Broker**

ФИНАМ

16 часов назад

## \$800 000 на фейковой аналитике: история самого юного биржевого мошенника

**Простой**

5 мин



6.9K

Ретроспектива

**+22**

20



5

**tataranovich**

9 часов назад

## Миллиомметр своими руками

**Простой**

2 мин



1.9K

**+21**

25



0

**SvetlanaPalitsyna**

15 часов назад

**Облачный IT-турнир**

Участвуй и получай призы

**Средний**

8 мин



848

Тutorial

+19

7

0



ElenaAladina

17 часов назад

## Не усложняй: простые фреймворки, чтобы быстрее принимать решения и приоритизировать задачи

3 мин

4K

+12

37

5

## «Омг, я буду работать среди станков» и другие мифы об айтишниках в металлургии

Турбо

Показать еще

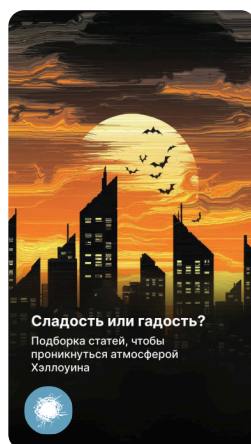
### ИСТОРИИ



**Выложены новые фото "Орла"**



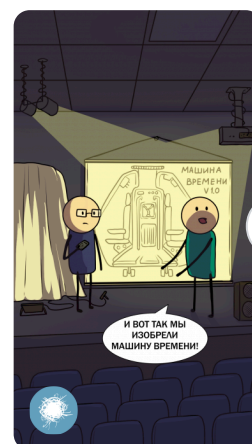
**Спрошу у лида**



**Сладость или гадость?**



**Топ-7 годных статей из блогов компаний**



**Как продвинуть машину времени?**



Облачный IT-турнир

Участвуй и получай призы

Провести глубокий технический SEO аудит и оптимизацию всего сайта

3000 руб./в час · 1 отклик · 3 просмотра

### Обработка альбомов с одеждой

8000 руб./за проект · 3 отклика · 35 просмотров

### Next. Js проект привести к pixel perfect,

30000 руб./за проект · 11 откликов · 71 просмотр

### Нужен бот

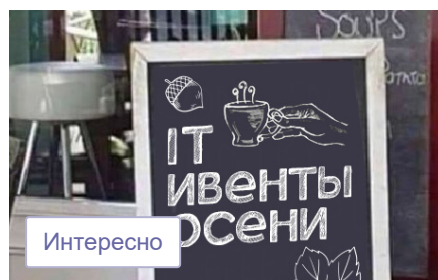
15000 руб./за проект · 32 отклика · 197 просмотров

### Помочь с интеграцией VK ID на сайт

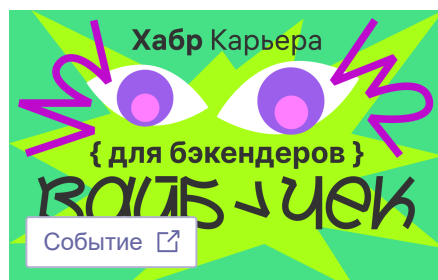
1000 руб./за проект · 3 отклика · 67 просмотров

[Больше заказов на Хабр Фрилансе](#)

## МИНУТОЧКУ ВНИМАНИЯ



Конференции, хакатоны, вебинары: новое осеннее меню



Бэкендеры, выбирайте команду по вайбам



Составляем портрет охотника на баги — с вашей помощью

## РАБОТА

### React разработчик

50 вакансий

### Node.js разработчик

50 вакансий

### JavaScript разработчик

170 вакансий



### Облачный IT-турнир

Участвуй и получай призы

## БЛИЖАЙШИЕ СОБЫТИЯ



8 октября – 4 декабря

**ТурбоХакатон «Решения для электроэнергетики на базе искусственного интеллекта»**

Онлайн

Разработка

Другое

[Больше событий в календаре](#)

## Хабр

**Облачный IT-турнир**

Участвуй и получай призы

Техническая поддержка

© 2006–2024, Habr



### Облачный IT-турнир

Участвуй и получай призы