

Хабр



КАК СТАТЬ АВТОРОМ



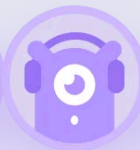
Хабр торт?



Войти

1000+

вакансий с удалёнкой



Хабр Карьера



empenoso

5 ноя 2024 в 03:23

## Работа в «песочнице» с торговым роботом на Московской бирже



Простой



6 мин



3.3K

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

Кейс

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

В статье разбираюсь как используя песочницу:

1. Открыть счёт.
2. Пополнить баланс счёта рублями через специальный запрос.
3. Посмотреть все свои открытые счета в песочнице.
4. Купить 1 акцию.
5. Продать 1 акцию.
6. Получить все открытые позиции указанного счёта.
7. Закрыть счёт.

```

src > JS sandboxjs > sandboxAccount
30  async function sandboxAccount() {
58  const GetSandboxAccounts = await tinkoffClient.callApi('Sandbox/GetAccounts', null, null);
59  logger.info('Список счетов в песочнице:\n ${JSON.stringify(GetSandboxAccounts, null, '\t')}\n\n');
60
61  // // Получить все открытые позиции указанного счёта
62  // const accountId = {
63  //   "accountId": secrets.AccountID
64  // };
65  // const GetSandboxPositions = await tinkoffClient.callApi('OperationsService/GetPositions', accountId);
66  // logger.info('Все открытые позиции счёта ${secrets.AccountID}:\n ${JSON.stringify(GetSandboxPositions, null, '\t')}\n\n');
67
68  // // Функция для отправки рыночного ордера
69  // tinkoffClient.placeMarketOrder('BBG004730N88', 1, 'ORDER_DIRECTION_BUY'); // Купить 1 акцию
70  tinkoffClient.placeMarketOrder('BBG004730N88', 1, 'ORDER_DIRECTION_SELL'); // Продать 1 акцию
71  }
72
73  // ===== Запуск функций =====
74  // =====
75
76
77  sandboxAccount().catch(logger.error);

```

```

ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ПОРТЫ
2024-11-01 14:11:58 [INFO]: Детали операции:
{
  "responseMetadata": {
    "serverTime": "2024-11-01T09:11:57.919185435Z"
  }
}
2024-11-01 14:11:58 [INFO]: Идентификатор продажи: 27a35903-3b38bc38c5e5.
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

```

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 конфиденциальную информацию.
- `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 `BVG004730N88` ). 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.

```

src > JS sandbox.js > sandboxAccount
30 async function sandboxAccount() {
48 // logger.info('Пополнение баланса счёта песочницы:\n ${JSON.stringify(SandboxPayIn, null, '\t')}\n\n');
49
50 // // Закрытие счёта в песочнице
51 // const accountId = {
52 //   "accountId": secrets.AccountID
53 // };
54 // const CloseSandboxAccount = await tinkoffClient.callApi('SandboxService/CloseSandboxAccount', accountId);
55 // logger.info('Закрытие счёта в песочнице:\n ${JSON.stringify(CloseSandboxAccount, null, '\t')}\n\n');
56
57 // Посмотреть счёта в песочнице
58 const GetSandboxAccounts = await tinkoffClient.callApi('SandboxService/GetSandboxAccounts');
59 logger.info('Список счётов в песочнице:\n ${JSON.stringify(GetSandboxAccounts, null, '\t')}\n\n');
60
61 // // Получить все открытые позиции указанного счёта
62 // const accountId = {
63 //   "accountId": secrets.AccountID
64 // };
65 // const GetSandboxPositions = await tinkoffClient.callApi('OperationsService/GetPositions', accountId);
66 // logger.info('Все открытые позиции счёта ${secrets.AccountID}:\n ${JSON.stringify(GetSandboxPositions, null, '\t')}\n\n');
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

```

2024-11-02 07:25:30 [INFO]: Список счётов в песочнице:
{
  "accounts": [
    {
      "status": "ACCOUNT_STATUS_OPEN",
      "openedDate": "2024-10-01T12:19:14.135019Z",
      "accessLevel": "ACCOUNT_ACCESS_LEVEL_FULL_ACCESS"
    }
  ]
}

```

[Done] exited with code=0 in 6.249 seconds

Строка 60, столбец 1 Пробелов: 4 UTF-8 CRLF (JavaScript)

Запрос SandboxService/GetSandboxAccounts

```

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

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

  // // Регистрации счёта в песочнице
  // const OpenSandboxAccount = await tinkoffClient.callApi('SandboxService/OpenSandboxAccount', accountId);
  // logger.info(`Регистрации счёта в песочнице:\n ${JSON.stringify(OpenSandboxAccount, null, '\t')}\n\n`);

  // // 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('SandboxService/SandboxPayIn',
// logger.info(`Пополнение баланса счёта песочницы:\n ${JSON.stringify(SandboxPayIn

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

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

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

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

}

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

sandboxAccount().catch(logger.error);

```

## Быстройдействие

Я не ждал какого-то особо быстрого действия. 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\sandbox.js"
2024-11-01 14:11:57 [INFO]: Запуск функции "sandboxAccount"

2024-11-01 14:11:58 [WARN]: Операция продажи выполнена успешно для Сбер Банк (SBER) (BB
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
  },
  "initialCommission": {
    "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-XXXX-3b38bc38c5e5
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 более одной секунды будет непозволительно долгим. Стратегии HFT 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

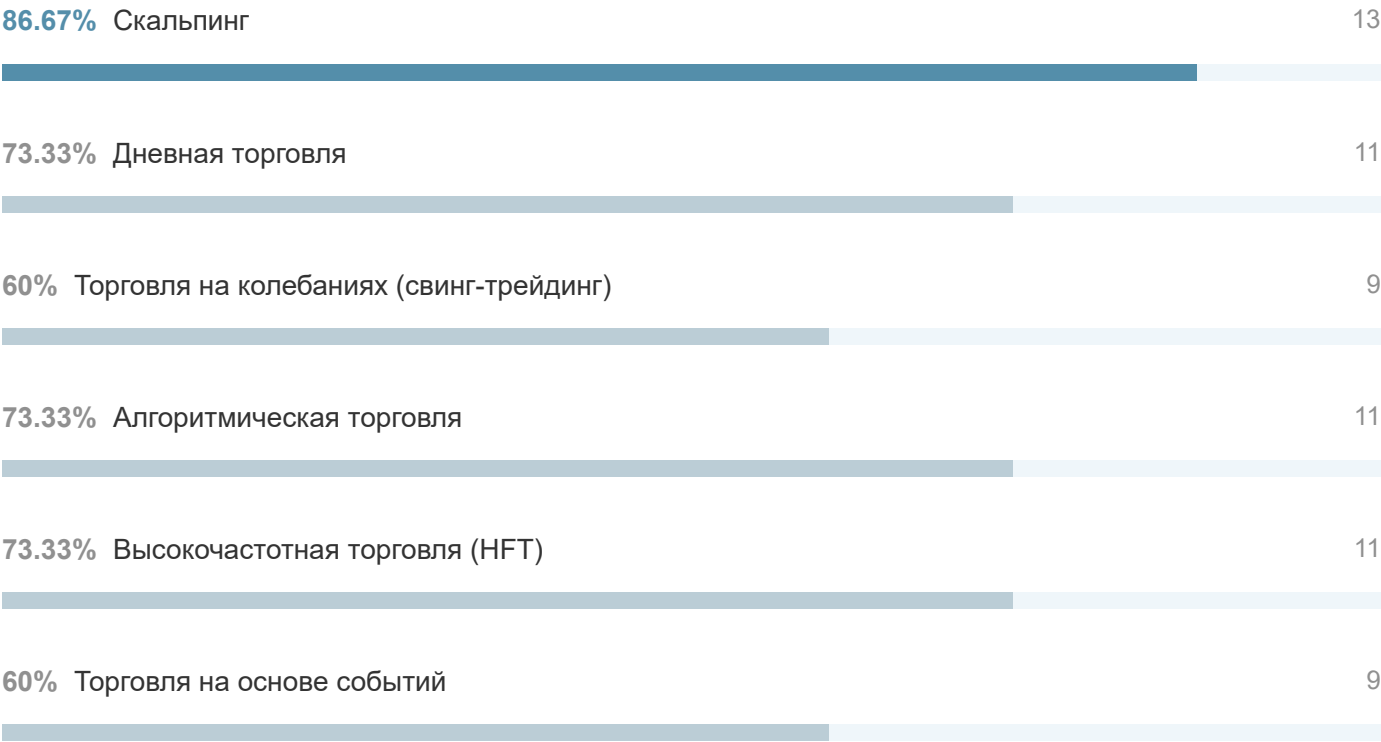
Проект полностью представлен на Гитхабе: <https://github.com/empenoso/SilverFir-TradingBot>. Новые модули будут загружаться по мере написания и тестирования.

Автор: [Михаил Шардин](#)

5 ноября 2024 г.

Только зарегистрированные пользователи могут участвовать в опросе. [Войдите](#), пожалуйста.

### Про какие виды торговли вы слышали?



Проголосовали 15 пользователей. Воздержался 1 пользователь.



**Теги:** [мосбиржа](#), [бот](#), [московская биржа](#), [moexalgo](#), [tbank](#), [t-bank invest api](#)

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

## Редакторский дайджест



Присылаем лучшие статьи раз в месяц

**179**

Карма

**30.4**

Рейтинг

**Михаил Шардин** [@empenoso](#)

Разработчик

[Подписаться](#)

[Сайт](#) [Сайт](#) [Github](#)

Комментарии 11

## Публикации

[ЛУЧШИЕ ЗА СУТКИ](#)[ПОХОЖИЕ](#)**xjr358**

23 часа назад

### IBM 600E Ретро ноутбук за 2000 тенге (400 рублей)

**Простой**

3 мин



5.6K

[Обзор](#)**+37**

12



39

**Erwinmal**

5 часов назад

## Кто поджёт Лос-Анджелес? Свежая конспирология о виноватых НЛО, Пи Дидди, урбанистах и корюшке

**Простой**

14 мин



3.2K

Обзор

**+26**

4



31

**DimDimDimDimDim**

6 часов назад

## Rust 1.84: новый релиз отличного языка программирования. Еще лучше, еще эффективнее, как всегда



6 мин



2.1K

**+17**

8



2

**JBFW**

14 часов назад

## Подключаем длинную линию 1-wire к Ардуино



3 мин



4.1K

**+17**

32



27

**arturdumchev**

48 минут назад

## Заговор разработчиков против корпораций

**Средний**

15 мин



841

Мнение

**+11**

2



0

**DENEVGAR**

4 часа назад

## Распознавание образов в мозге с помощью микроплееров

 Средний  8 мин  903

Из песочницы

 +11

 15

 5



**chlorine**

6 часов назад

## Кэш. Теория кэширования. Устройство и разновидности кэша

 Простой  7 мин  1.8K

Из песочницы

 +11

 60

 16



**mikhailmurzak**

20 часов назад

## Делаем Телеграм-бота в Cursor AI без знания кода

 Простой  5 мин  6.4K

Тutorial

 +11

 73

 15



**burenikov**

2 часа назад

## Стереокамера машинного зрения с поддержкой ИИ на базе FPGA и Arduino Portenta H7

 10 мин  596

Из песочницы

 +10

 10

 0



**andreymbold**

7 часов назад

## Как я развлекался с восходом солнца

 Простой  3 мин  1.4K

◆ +7

📖 5

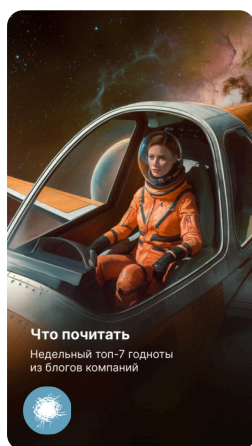
💬 29

## Разбираем подходы к реализации гиперконвергентной среды

Турбо

Показать еще

### ИСТОРИИ



**Что почитать**  
Недельный топ-7 годноты из блогов компаний

**Годнота из блогов компаний**



**Смотрим на небо: выравнивания планет**  
Два редких астрономических явления, когда одновременно в ряд на ночном небе выстраиваются сразу множество планет Солнечной системы, можно будет наблюдать этой зимой.



**Выравнивания планет**



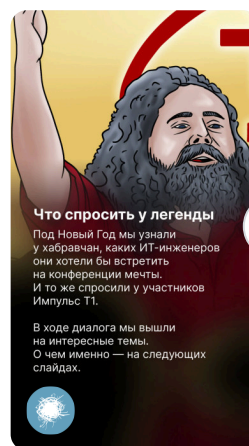
**Что происходит в нейромире**  
Подборка популярных статей о нейросетях и их применении за последние месяцы

**Нейрозима 2025**



**Статьи с новогодним вайбом**  
Поднимут настроение после праздников

**Статьи с новогодним вайбом**



**Что спросить у легенды**  
Под Новый Год мы узнали у хабравчан, каких IT-инженеров они хотели бы встретить на конференции мечты. И то же спросили у участников Импульс Т1.

В ходе диалога мы вышли на интересные темы. О чем именно — на следующих слайдах.

**Кто выступит на конференции мечты**

### ЗАКАЗЫ

Подключить контур эдо

25000 руб./за проект · 1 отклик · 7 просмотров

Доработка расширения переводчик ChatGPT под Google Chrome (C# Backend)

20000 руб./за проект · 14 просмотров

Написать телеграм бот Гадание на картах Таро с результатами из ChatGPT

10000 руб./за проект · 8 откликов · 52 просмотра

Найти ручку для парсинга курса и упростить его

10000 руб./за проект · 3 отклика · 48 просмотров

Доработка web app телеграмм Next JS Python PostgreSQL

50000 руб./за проект · 7 откликов · 47 просмотров  
Больше заказов на [Хабр Фрилансе](#)

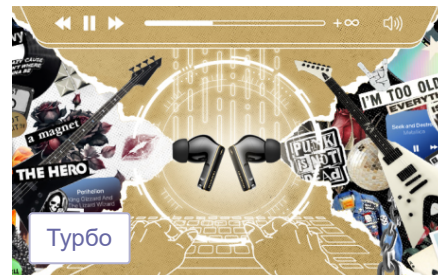
## МИНУТОЧКУ ВНИМАНИЯ



Испытание огнем и ловушками:  
спасти мир с помощью лассо



Гиперконвергентная среда:  
OpenStack или VMware?



Так ли хороши звук и шумодав:  
обзор наушников FreeBuds Pro 4

## РАБОТА

[Node.js разработчик](#)

33 вакансии

[JavaScript разработчик](#)

98 вакансий

[React разработчик](#)

30 вакансий

[Все вакансии](#)

## БЛИЖАЙШИЕ СОБЫТИЯ



30 января

## Зимний тест-драйв Хабра для компаний

Москва

Маркетинг

Другое

[Больше событий в календаре](#)

Хабр



🌐 [Настройка языка](#)

[Техническая поддержка](#)

© 2006–2025, Habr