

Все важнейшие
IT-события здесь

5

6

7

8

9

10

11

12

13

14

15

16

17

18



empenoso

5 ноя 2024 в 03:23

Работа в «песочнице» с торговым роботом на Московской бирже



Простой



6 мин



3.8K

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 API 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 использую 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 sandbox.js > sandboxAccount
30 async function sandboxAccount() {
58   const GetSandboxAccounts = await tinkoffClient.callApi('SandboxService/GetAccounts', null, 't');
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 чеpez 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 счета в песочнице
58 const GetSandboxAccounts = await tinkoffClient.callApi('SandboxService/GetSandboxAccounts');
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
2024-11-02 07:25:30 [INFO]: Список счетов 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

```

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

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

// // 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);
```

Быстродействие

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\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

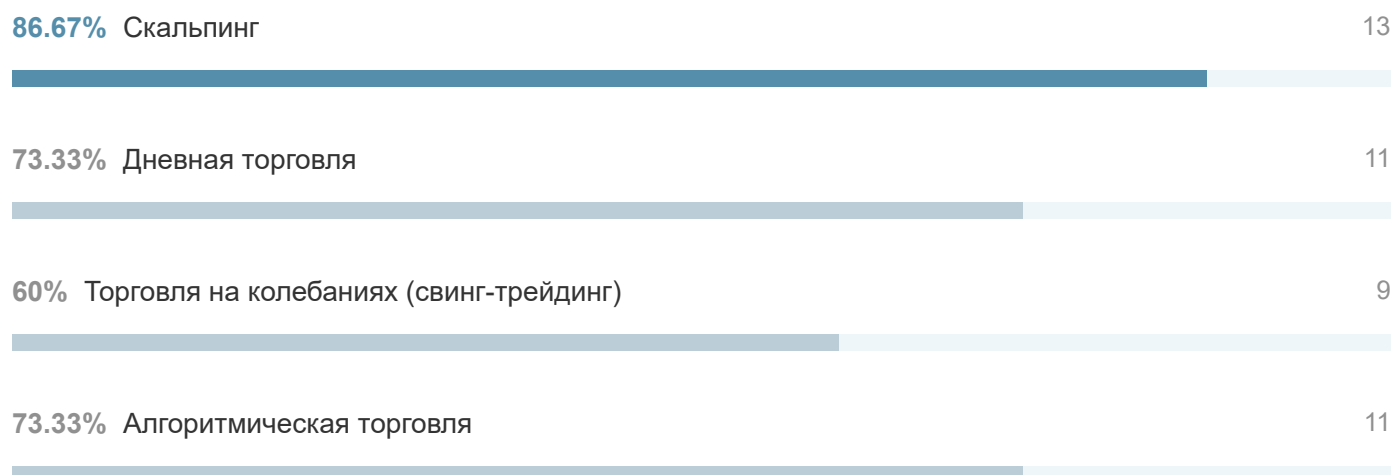
 [Моя онлайн-визитка](#)

 [Telegram «Умный Дом Инвестора»](#)

5 ноября 2024 г.

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

Про какие виды торговли вы слышали?



73.33% Высоочастотная торговля (HFT)

11

60% Торговля на основе событий

9

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

**183****87.1**

Карма

Рейтинг

Михаил Шардин [@empenoso](#)

[Автоматизация](#) / [Данные](#) / [Финансы](#) / [Умные дома](#)

[Подписаться](#)

[Сайт](#) [Сайт](#) [Github](#)

 Комментарии 11

Публикации

ЛУЧШИЕ ЗА СУТКИ

ПОХОЖИЕ



rssdev10

22 часа назад

Почему въехав по «визе талантов» в США я с радостью вернулся в Россию



Средний



32 мин



34K

Мнение



+172



110



500



melnik909

18 часов назад

Вы не знаете CSS. Мои вопросы о CSS с ответами. Часть 2



Средний



7 мин



2K

Обзор

 +38 31 1**DAN_SEA**

16 часов назад

Генерация случайных чисел



Средний



10 мин



2.4K

Обзор

 +31 23 33**OrkBiotecnologist**

22 часа назад

VPS за 139 рублей — дом для вашего резюме на основе Hugo



Простой



7 мин



8.3K

Тutorial

 +28 45 12**nebykoff_anton**

23 часа назад

До и после: оптимизация изображений для Lighthouse и не только



Простой



8 мин



947

Тutorial

 +28 24 2**Корcheniy**

23 часа назад

Имитатор касаний. Ч2: Железная часть



Средний



9 мин



1.7K

Кейс

 +24 20 12

**PatientZero**

2 часа назад

Пишем стек TCP/IP с нуля: Ethernet, ARP, IPv4 и ICMPv4

**Простой**

13 мин



806

Тutorial

Перевод

**+18**

24



1

**tertiumnon**

17 часов назад

Минимум книг, которые нужно прочитать начинающему или продолжающему свою кривую обучения программисту

**Простой**

3 мин



8.5K

Обзор

**+17**

188



20

**lbkanter**

59 минут назад

Бэкдор Auto-color: разбор угрозы, технический анализ и способы защиты

**Средний**

4 мин



251

Обзор

**+13**

5



2

**alexander-shustanov**

18 часов назад

В поисках идеального Database-клиента для IDE: Amplicode выбирает DBeaver

**Простой**

6 мин



2.5K

**+13**

13



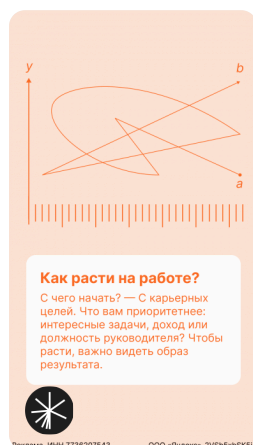
7

2FA и атаки через VPN, ADFS, SSH — как второй фактор защищает ваши данные

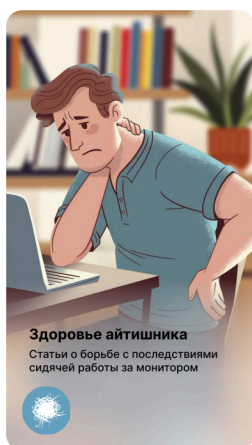
Турбо

Показать еще

ИСТОРИИ



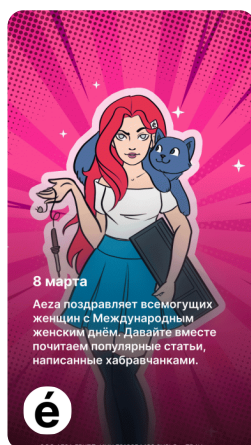
Как расти на работе?



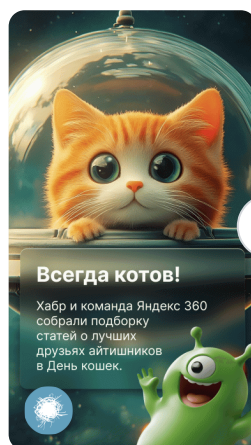
Здоровье айтишника



Угадайте будущее в новом сезоне



С праздником весны!

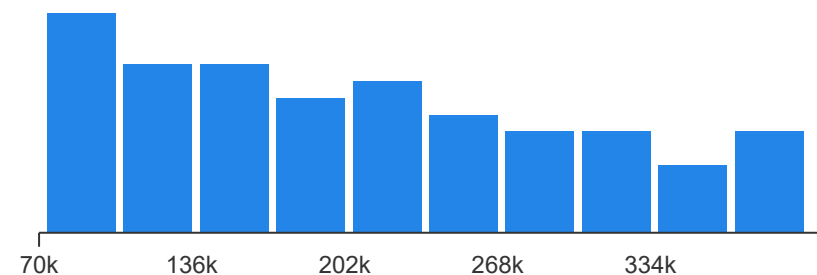


Всегда котов!

СРЕДНЯЯ ЗАРПЛАТА В IT

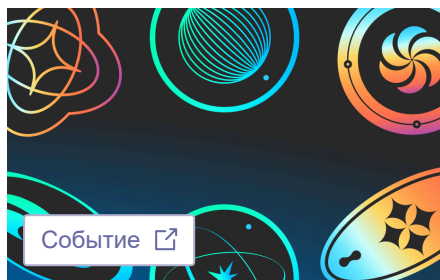
212 662 ₽/мес.

— средняя зарплата во всех IT-специализациях по данным из 33 637 анкет, за 1-ое пол. 2025 года. Проверьте «в рынке» ли ваша зарплата или нет!

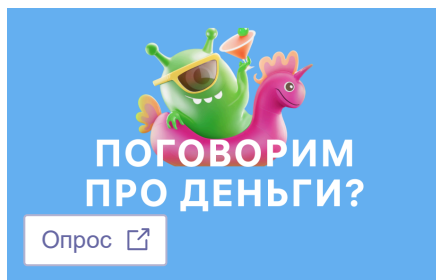


Проверить свою зарплату

МИНУТОЧКУ ВНИМАНИЯ



Как писать про здоровье на
Хабре? Расскажем на вебинаре



Как айтишники подходят к
финансовому планированию?



Девушка с розовыми волосами
и Слизень на планете Рбах

РАБОТА

Node.js разработчик

56 вакансий

React разработчик

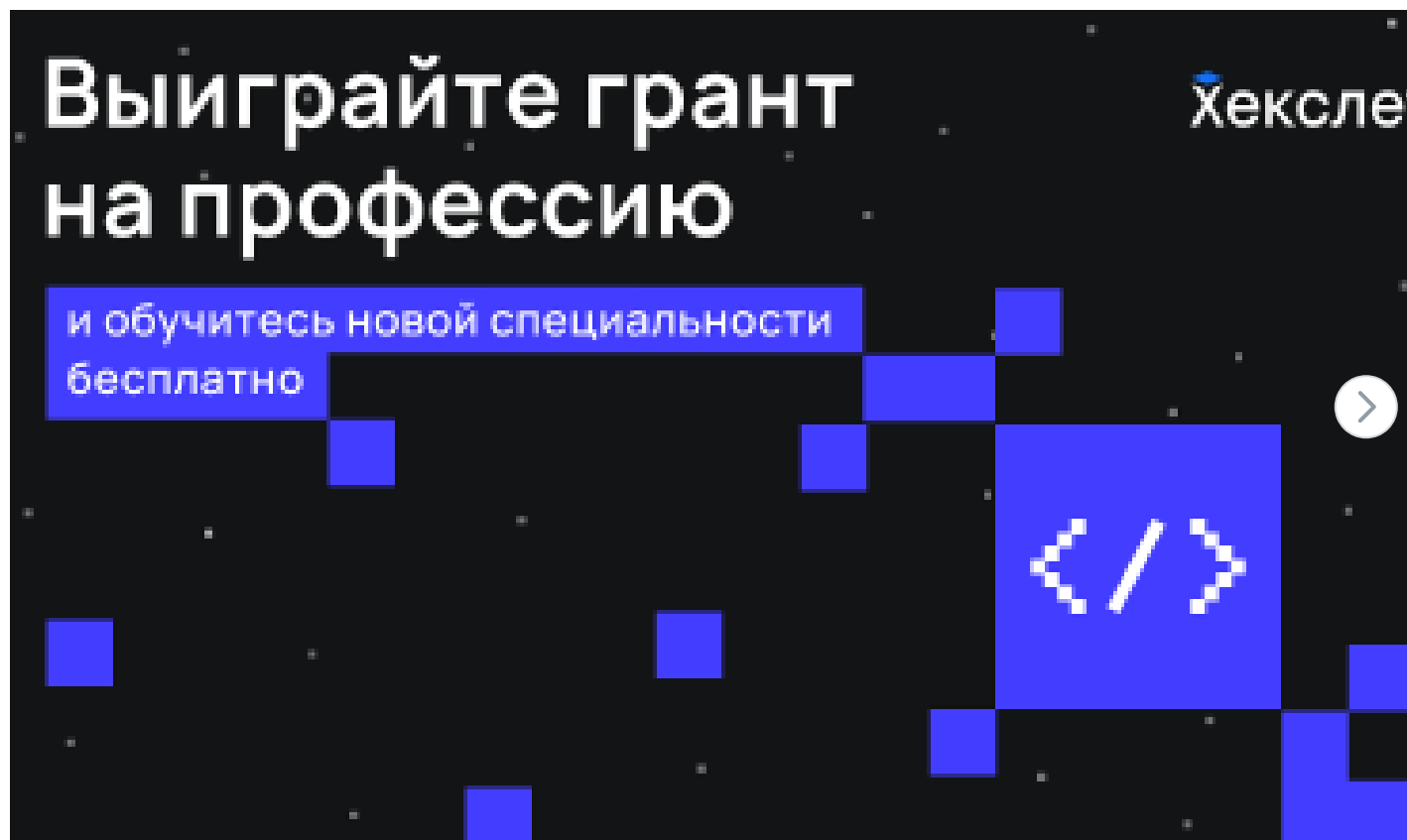
48 вакансий

JavaScript разработчик

134 вакансии

[Все вакансии](#)

БЛИЖАЙШИЕ СОБЫТИЯ



17 февраля – 24 марта

Конкурс «Снежный код» от Хекслета. Три гранта на бесплатное 10-месячное обучение

Онлайн

Разработка

[Больше событий в календаре](#)

Хабр



🌐 [Настройка языка](#)

[Техническая поддержка](#)

© 2006–2025, Habr