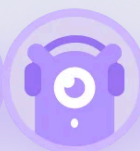


1000+

вакансий с удалёнкой



Хабр Карьера



empenoso

4 часа назад

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

🔥 Сложный

🕒 6 мин

👁 644

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 АПИ (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:

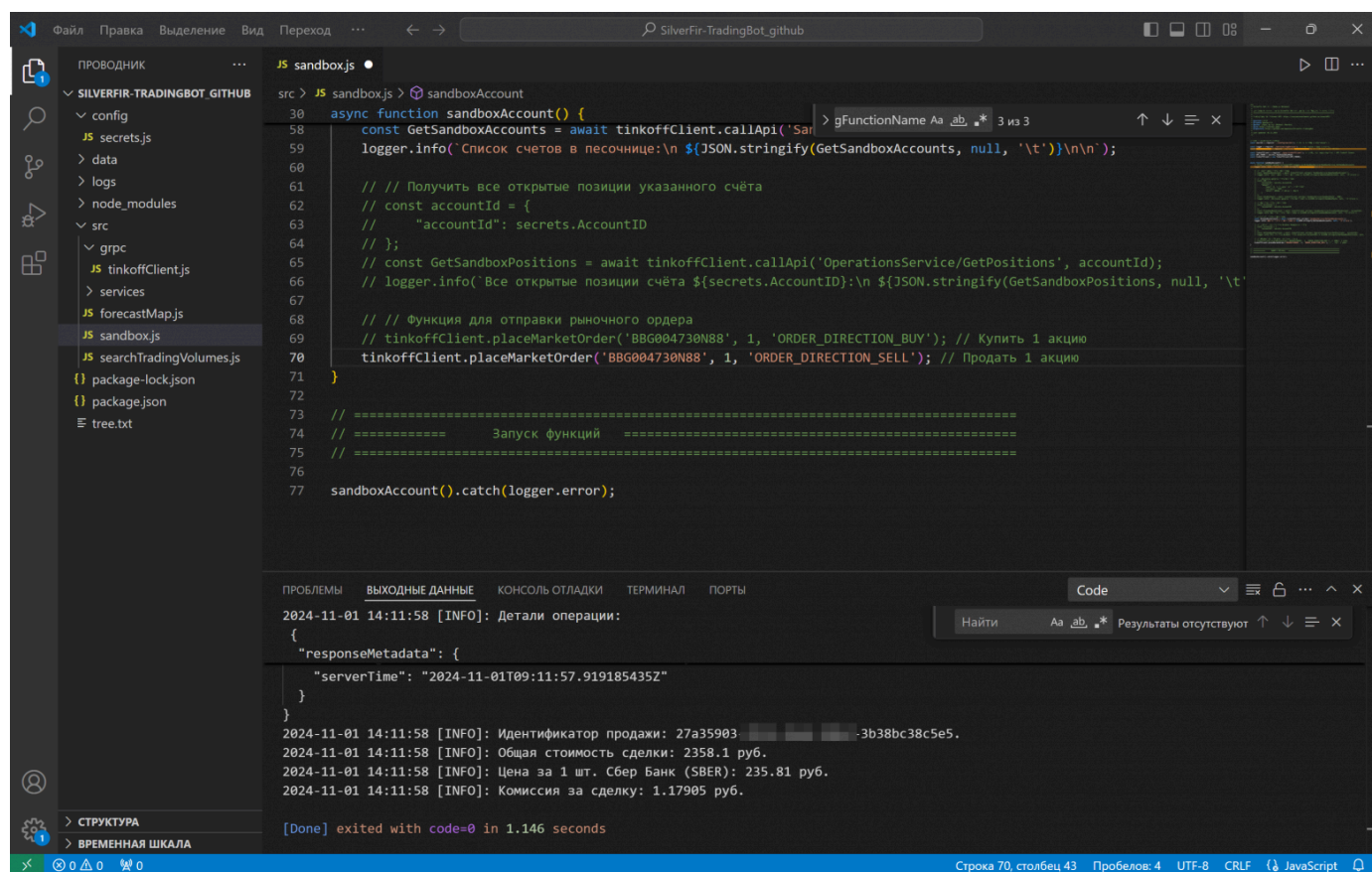
1. Открыть счёт.
2. Пополнить баланс счёта рублями через специальный запрос.
3. Посмотреть все свои открытые счета в песочнице.
4. Купить 1 акцию.
5. Продать 1 акцию.

РЕКЛАМА

**Облачный IT-турнир**

Участвуй и получай призы

7. Закрыть счет.



```
src > JS sandbox.js
30 async function sandboxAccount() {
58   const GetSandboxAccounts = await tinkoffClient.callApi('Sandbox/GetAccounts', {
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);
```

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



**Облачный IT-турнир**

Участвуй и получай призы

- `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



**Облачный IT-турнир**

Участвуй и получай призы

```

src > JS sandboxjs > 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
PROBLEMY 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

```

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', {

```



**Облачный IT-турнир**

Участвуй и получай призы

```

// 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);

```



**Облачный IT-турнир**

Участвуй и получай призы

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": "BBG001730N88"
```

**Облачный IT-турнир**

Участвуй и получай призы

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.



**Облачный IT-турнир**

Участвуй и получай призы

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

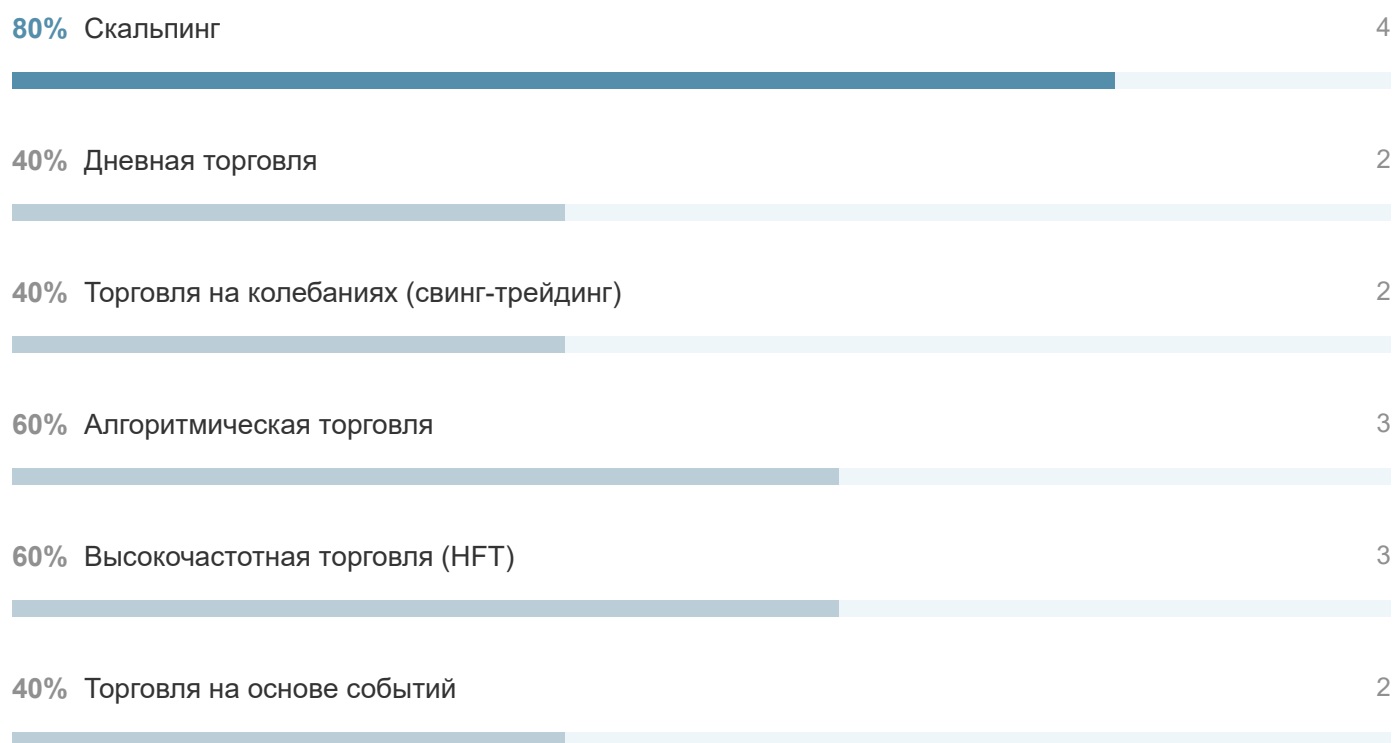
Проект полностью представлен на Гитхабе: <https://github.com/empenoso/SilverFir-TradingBot>. Новые модули будут загружаться по мере написания и тестирования.

Автор: [Михаил Шардин](#)

5 ноября 2024 г.

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

### Про какие виды торговли вы слышали?



Проголосовали 5 пользователей. Воздержавшихся нет.



**Облачный IT-турнир**

Участвуй и получай призы



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

## Редакторский дайджест



Присылаем лучшие статьи раз в месяц

**164****32.1**

Карма

Рейтинг

**Михаил Шардин** [@empenoso](#)

Разработчик

[Подписаться](#)[Сайт](#) [Сайт](#) [Github](#) [Telegram](#) [Комментарии 2](#)

## Публикации

[ЛУЧШИЕ ЗА СУТКИ](#)[ПОХОЖИЕ](#)**Androgenom****Облачный IT-турнир**

Участвуй и получай призы


 Средний  10 мин  9.6K

Из песочницы

 +119

 63

 121

 **RiddleRider**  
20 часов назад

## Ошибка ценой в 50 лет: как Intel потеряла корону в процессорной гонке


 Простой  9 мин  38K

Обзор

 +94

 51

 144

 **shiru8bit**  
15 часов назад

## Все мои Ардуинки: часть первая, восьмибитные на ATmega


 Простой  18 мин  3.8K

Обзор

 +32

 36

 7

 **alizar**  
19 часов назад

## Ядерные реакторы для космоса. Возвращение

 Средний  7 мин  4.5K

Обзор

 +29

 12

 9

 **Seleditor**  
15 часов назад

## Процессор Itanium в Linux не забыли: архитектуру IA-64



**Облачный IT-турнир**

Участвуй и получай призы

 +27 6 11

slava\_rumin

14 часов назад

Мам, а ты можешь принести твоего хлеба на год вперед? Или почему мое производство больше ничего не испечет



Простой



13 мин



7.2K

Интервью

 +22 31 62

Seleditor

20 часов назад

Selectel School 2.0: как мы перезапустили курс для системных инженеров



5 мин



1.8K

Кейс

 +20 20 3

tarlakovsky

18 часов назад

После паузы Voyager 1 вышел на связь с командой миссии



Простой



3 мин



5K

Перевод

 +16 10 10

PatientZero

17 часов назад

Как мы нашли уязвимость в SQLite при помощи LLM



Средний



9 мин



2.6K



Облачный IT-турнир

Участвуй и получай призы



Giox\_Nostr

10 часов назад

# Prolog: три портрета через 50 лет



Простой



25 мин



1.4K

Из песочницы



+13



9



8

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

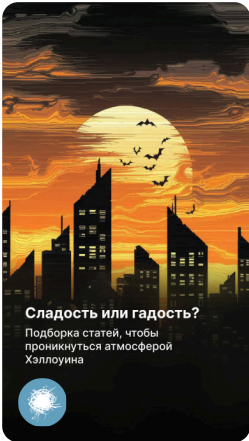
Турбо

Показать еще

### ИСТОРИИ



Спрошу у лида



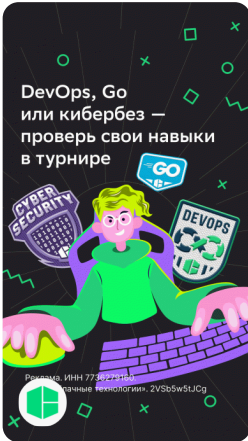
Сладость или гадость?



Актуальные зарплаты аналитиков



Топ-7 годных статей из блогов компаний



Облачный IT-турнир



Как маш

### ВАКАНСИИ

NodeJS Бэкэнд разработчик



Облачный IT-турнир

Участвуй и получай призы

от 176 000 до 240 000 ₽ · Complex Cloud Solutions · Можно удаленно

## JavaScript Developer

от 80 000 до 180 000 ₽ · AppsTrain.io · Можно удаленно

## JavaScript FullStack разработчик

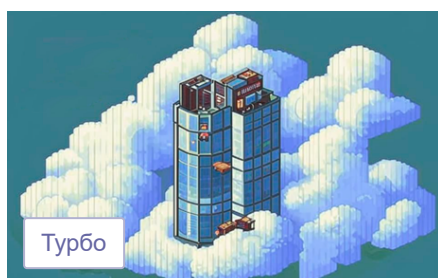
от 80 000 до 130 000 ₽ · Rocket · Смоленск · Можно удаленно

## JavaScript FullStack developer

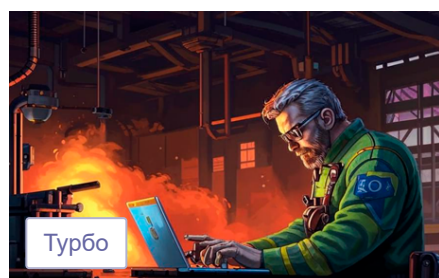
до 220 000 ₽ · Wanted. · Санкт-Петербург

[Больше вакансий на Хабр Карьере](#)

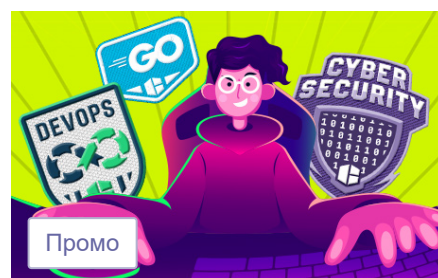
### МИНУТОЧКУ ВНИМАНИЯ



**Курс на автоматизацию и облака: о трендах DevOps в России**



**Как работает айтишникам на металлургическом комбинате**



**DevOps, Go или кибербез — проверь свои навыки в турнире**

### РАБОТА

#### JavaScript разработчик

176 вакансий

#### React разработчик

58 вакансий

#### Node.js разработчик

52 вакансии

[Все вакансии](#)



#### Облачный IT-турнир

Участвуй и получай призы



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