

Хабр



КАК СТАТЬ АВТОРОМ



Новые знания здесь



Войти



empenoso

27 ноя 2024 в 05:23

## Отслеживание позиций торгового робота Московской биржи через CSV файл

Простой

7 мин

2.8К

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

Кейс

Нахожусь в процессе написания механизма торгового робота, работающего на Московской бирже через API одного из брокеров. Брокеров имеющих своё АПИ для МосБиржи катастрофически мало — мне известно только о трёх. При этом, когда я стал публиковать модули робота (и полностью [выложу готовый механизм робота на GitHub](#)), то стал получать непонимание — например, мне писали в комментариях — зачем придумывать велосипед, когда уже есть QUIK — популярная российская платформа для биржевых торгов. В Квике уже есть готовый функционал «импорт транзакций из файла» или таблица «карман транзакций». В тех же комментариях предлагали даже рассмотреть использование платформы 1С для робота, но оказалось, что торговля все равно будет осуществляться через импорт `.tri`-файла в Квик.

Лично мне Квик не очень нравится тем, что это программа для Windows. Хочется иметь механизм торгового робота, который был бы кроссплатформенным и легким — это позволит использовать его даже на «слабом» сервере. К тому же, [много лет назад](#), когда Квик был единственной альтернативой для частного лица, невозможно было внутри одной Windows без использования виртуальной машины запустить несколько копий программы технического анализа с разными системами - для того, чтобы каждая из этих копий отправляла свои сигналы на покупку и продажу в соответствующий Квик. Это было нужно для разных торговых стратегий.

По субъективным причинам я стал писать [торгового робота в среде исполнения JavaScript Node.js](#), но для тестирования на истории пришлось [использовать Python и его библиотеки](#).

РЕКЛАМА



Получи грант за код

Конкурс open source проектов

```

src > services > # csvHandler.js > loadPositions > on[data] callback
15 const path = require('path'); // Модуль для работы с путями файлов и директории
20 const filePath = path.join(__dirname, '../data/positions.csv'); // Путь к файлу CSV
21 const logger = require('./logService'); // Подключаем модуль для логирования
22 const logFunctionName = require('./logFunctionName'); // Модуль для получения имени функции (для логирования)
23
24
25 // Загружаем все позиции из CSV файла
26 function loadPositions() {
27   return new Promise((resolve, reject) => {
28     const positions = [];
29     fs.createReadStream(filePath)
30       .pipe(csv())
31       .on('data', (row) => {
32         positions.push({
33           ticker: row.ticker,
34           figi: row.figi,
35           quantity: parseFloat(row.quantity), // Преобразование количества в float
36           purchaseDate: row.purchaseDate,
37           purchasePrice: parseFloat(row.purchasePrice), // Преобразование цены покупки в float
38           updateDate: row.updateDate,
39           maxPrice: parseFloat(row.maxPrice), // Преобразование максимальной цены в float
40           profitLoss: parseFloat(row.profitLoss) // Преобразование прибыли/убытков в float
41         });
42       })
43       .on('end', () => resolve(positions))
44       .on('error', reject);
45   });
46 }
47
48 // Сохраняем актуальные данные о позициях в CSV файл
49 function savePositions(positions) {
50   const csvFields = ['ticker', 'figi', 'quantity', 'purchaseDate', 'purchasePrice', 'updateDate', 'maxPrice', 'profitLoss'];
51   const csvData = parse(positions, { fields: csvFields });
52
53   fs.writeFileSync(filePath, csvData);

```

Модуль, считывающий позиции из файла

## Проблемы с записью позиций в Node.js

Вообще именно этот модуль пришлось пару раз переписывать, потому что не смог сразу отладить его. Проблема была в том, что вызов модуля записи и обновления позиций осуществлялся сразу из нескольких мест и одни результаты перезаписывали другие. Но удалось разобраться и теперь всё протестировано и работает.

Дополнительно использую библиотеки `csv-parser` и `json2csv` — это популярные инструменты Node.js для обработки данных CSV, каждая из которых служит различным целям:

- **csv-parser**, это легкая и быстрая библиотека для анализа файлов CSV. Она основана на потоках, что делает ее очень эффективной для обработки больших наборов данных.
- **json2csv**, это утилита для преобразования данных JSON в формат CSV. Идеально подходит для экспорта данных из приложений в структуру, удобную для CSV, может работать как синхронно, так и асинхронно.

Установка этих библиотек:



Получи грант за код

Конкурс open source проектов

```
npm install csv-parser json2csv
```

## Мой модуль csvHandler.js

Этот код определяет модуль для взаимодействия с CSV-файлом для управления финансовыми торговыми позициями. Служит для загрузки, сохранения, обновления и удаления финансовых позиций, хранящихся в CSV-файле.

### Ключевые библиотеки:

- `fs` : для операций файловой системы, таких как чтение и запись файлов.
- `csv-parser` : для анализа CSV-файлов в объекты JavaScript.
- `json2csv` : для преобразования объектов JavaScript в формат CSV для сохранения.
- `path` : для управления путями к файлам.
- **Интеграция**: включает пользовательские модули для ведения журнала ( `logService` ) и получения имен функций для лучшей отладки.

### Функциональность

1. Обработка пути к файлу: использует модуль `path` для поиска CSV-файла, хранящего данные о позиции: `../../data/+positions.csv` .
2. Функции управления позицией:

`loadPositions()`:

1. Считывает CSV-файл и анализирует его в массив объектов позиции.
2. Преобразует числовые поля (`quantity`, `purchasePrice`, `maxPrice`, `profitLoss`) в числа с плавающей точкой для вычислений.
3. Возвращает обещание, которое разрешается с проанализированными данными или отклоняется в случае ошибки.



**Получи грант за код**

Конкурс open source проектов

1. Преобразует массив объектов позиции обратно в формат CSV с помощью `json2csv`.
2. Перезаписывает CSV-файл обновленными данными.

```
removePosition(figi):
```

1. Удаляет позицию из CSV-файла на основе ее `figi` (уникального идентификатора).
2. Загружает все позиции, отфильтровывает указанную и перезаписывает файл.

```
updatePosition(newPosition):
```

1. Добавляет новую позицию или обновляет существующую в CSV-файле:
  2. Если `figi` существует, обновляет соответствующую позицию.
  3. В противном случае добавляет новую позицию.
  4. Сохраняет обновленный список обратно в CSV-файл.
3. Экспортированные модули: функции `loadPositions`, `updatePosition` и `removePosition` для использования в других частях робота.

### Полный код `csvHandler.js`:

```
const fs = require('fs');
const csv = require('csv-parser');
const { parse } = require('json2csv');
const path = require('path'); // Модуль для работы с путями файлов и директорий
const filePath = path.join(__dirname, '../data/positions.csv'); // Путь к файлу CSV
const logger = require('./logService'); // Подключаем модуль для логирования
const logFunctionName = require('./logFunctionName'); // Модуль для получения имени функции

// Загружаем все позиции из CSV файла
function loadPositions() {
  return new Promise((resolve, reject) => {
    const positions = [];
    fs.readFile(filePath, 'utf8', (err, data) => {
      if (err) {
        logger.error('Ошибка при загрузке CSV файла: ' + err);
        reject(err);
      } else {
        const rows = data.split('\n').map(row => {
          const cols = row.split(',');
          const position = {
            figi: cols[0],
            name: cols[1],
            price: cols[2],
            volume: cols[3],
            time: cols[4]
          };
          positions.push(position);
        });
        resolve(positions);
      }
    });
  });
}
```



**Получи грант за код**

Конкурс open source проектов

`.on('data', (row) => {`

```
        positions.push({
            ticker: row.ticker,
            figi: row.figi,
            quantity: parseFloat(row.quantity), // Преобразование количества в
            purchaseDate: row.purchaseDate,
            purchasePrice: parseFloat(row.purchasePrice), // Преобразование цен
            updateDate: row.updateDate,
            maxPrice: parseFloat(row.maxPrice), // Преобразование максимальной
            profitLoss: parseFloat(row.profitLoss) // Преобразование прибыли/убытка
        });
    })
    .on('end', () => resolve(positions))
    .on('error', reject);
});
}

// Сохраняем актуальные данные о позициях в CSV файл
function savePositions(positions) {
    const csvFields = ['ticker', 'figi', 'quantity', 'purchaseDate', 'purchasePrice', 'updateDate'];
    const csvData = parse(positions, { fields: csvFields });

    fs.writeFileSync(filePath, csvData);
}

// Удаляем позицию из CSV файла (после продажи)
function removePosition(figi) {
    loadPositions().then(positions => {
        const updatedPositions = positions.filter(position => position.figi !== figi);
        savePositions(updatedPositions);
    });
}

// Добавляем новую позицию или обновляем существующую в CSV файле
function updatePosition(newPosition) {
    loadPositions().then(positions => {
        const index = positions.findIndex(pos => pos.figi === newPosition.figi);

        if (index === -1) {
            // Добавляем, если не нашли существующую позицию
            positions.push(newPosition);
        }
    });
}
```

**Получи грант за код**

Конкурс open source проектов

```
    }

    savePositions(positions);
  });
}

module.exports = { loadPositions, updatePosition, removePosition };
```

## Мой модуль checkCSVpositions.js

Этот модуль важен для обеспечения согласованности данных между локальным CSV-файлом и текущими позициями, полученными из T-Bank Invest API. Он проверяет наличие несоответствий, которые могут привести к ошибкам в торговых операциях, и останавливает робота, если обнаруживаются несоответствия.

## Основные функции

### 1. Интеграция с внешними системами

- T-Bank Invest API: взаимодействует с API для извлечения торговых позиций в реальном времени.
- CSV File Management: использует локальный CSV-файл для хранения и управления представлением бота о торговых позициях.

### 2. Проверка согласованности

- Сравнивает позиции из CSV-файла с позициями с сервера T-Bank Invest API.
- Проверяет как количество, так и наличие позиций для обнаружения несоответствий.

### 3. Обработка ошибок

- Регистрирует подробные ошибки при обнаружении несоответствий.
- Останавливает торговые операции для предотвращения дальнейших действий на основе неверных данных.



**Получи грант за код**

Конкурс open source проектов

## 1. `getServerPositions()`

- Извлекает все открытые позиции из T-Bank Invest API.
- Извлекает позиции с ценными бумагами и преобразует баланс в float для сравнения.
- Регистрирует ответ сервера для отладки и аудита.

## 2. `checkForDiscrepancies()`

- Загружает данные CSV: считывает локальную запись позиций бота с помощью `csvHandler`.

Сравнивает позиции:

- Для каждой позиции CSV ищет соответствующую позицию на сервере с помощью FIGI (уникальный идентификатор).
- Извлекает размер лота для точного сравнения количества.
- Если обнаружены расхождения в количестве или отсутствующие позиции, регистрирует ошибки и останавливает торговлю.
- Статус журнала: подтверждает, когда все позиции совпадают, и позволяет продолжить торговлю.

## Рабочий процесс

### 1. Извлечение позиций:

- Локальные позиции загружаются из CSV-файла.
- Позиции сервера извлекаются через API Tinkoff.

### 2. Обнаружение расхождений:

Для каждой позиции в CSV-файле:

3. Код вычисляет общее количество в лотах ( `csvPosition.quantity * lotSize` ).

4. Сравнивает с балансом на сервере



**Получи грант за код**

Конкурс open source проектов

5. Количества не совпадают.

6. Позиция в CSV-файле отсутствует на сервере.

### 3. Безопасность робота:

- Любые обнаруженные расхождения вызывают ошибку, останавливающую торговые операции.
- Не позволяет роботу совершать сделки на основе устаревших или неверных данных.

### Полный код checkCSVpositions.js:

```
const logger = require('./logService'); // Логирование в файл и консоль
const logFunctionName = require('./logFunctionName'); // Получение имени функции

const secrets = require('../../config/secrets'); // Ключи доступа и идентификаторы
const config = require('../../config/config'); // Параметры
const csvHandler = require('./csvHandler'); // Работа с CSV файлами

const TinkoffClient = require('../grpc/tinkoffClient'); // Модуль для взаимодействия с
const API_TOKEN = secrets.TbankSandboxMode;
const tinkoffClient = new TinkoffClient(API_TOKEN);

// Функция для получения всех позиций с сервера
async function getServerPositions() {
  try {
    const accountId = {
      accountId: secrets.AccountID
    };
    const response = await tinkoffClient.callApi('OperationsService/GetPositions',

    // Логировем полученные позиции с сервера
    logger.info(`Все открытые позиции счета ${secrets.AccountID}: \n ${JSON.stringify

    // Возвращаем только позиции с ценными бумагами (securities)
    return response.securities.map(sec => ({
      figi: sec.figi,
```



**Получи грант за код**

Конкурс open source проектов



```
logger.error(`Ошибка при получении позиций с сервера: ${error.message}`);
throw error;
}
}

// Функция для проверки расхождений
async function checkForDiscrepancies() {
  try {
    // Загружаем текущие позиции из CSV файла
    var csvPositions = await csvHandler.loadPositions();

    // Получаем позиции с сервера
    const serverPositions = await getServerPositions();

    // Проверяем каждую позицию из CSV
    for (const csvPosition of csvPositions) {
      // Находим соответствующую позицию с сервера
      const serverPosition = serverPositions.find(pos => pos.figi === csvPosition.figi);

      if (serverPosition) {
        const lotSize = await tinkoffClient.getLot(csvPosition.figi);
        logger.info(`Количество бумаг в лоте ${csvPosition.figi}: ${lotSize} шт`);
        const csvTotal = csvPosition.quantity * lotSize;

        // Сравниваем количество позиций
        if (csvTotal !== serverPosition.balance) {
          // Если есть расхождение, логируем ошибку и останавливаем торгового робота
          logger.error(`Ошибка: Несоответствие по FIGI ${csvPosition.figi}. CSV баланс: ${csvTotal}, серверный баланс: ${serverPosition.balance}`);
          throw new Error('Найдено несоответствие позиций. Остановка торговли');
        }
      } else {
        logger.error(`Ошибка: Позиция с FIGI ${csvPosition.figi} отсутствует на сервере`);
        throw new Error('Найдено несоответствие позиций. Остановка торговли. ');
      }
    }

    logger.info('Все позиции совпадают. Торговля продолжается. ');
  } catch (error) {
    logger.error(`Ошибка при проверке позиций: ${error.message}`);
    // Останавливаем торгового робота (добавьте здесь вашу логику остановки)
  }
}
```

**Получи грант за код**

Конкурс open source проектов

```
// Экспортируем функции
module.exports = {
  checkForDiscrepancies
};

// checkForDiscrepancies().catch(logger.error);
```

## Итоги

Проект полностью представлен на [Гитхабе](#). Новые модули будут загружаться по мере написания и тестирования.

**Автор:** Михаил Шардин

 [Моя онлайн-визитка](#)

 [Telegram «Умный Дом Инвестора»](#)

27 ноября 2024 г.

**Теги:** [московская биржа](#), [мосбиржа](#), [моех](#), [моехalgo](#), [tbank](#), [t-bank invest api](#), [алгоритмическая торговля](#)

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

## Редакторский дайджест



Присылаем лучшие статьи раз в месяц



Оставляя свою почту, я принимаю [Политику конфиденциальности](#) и даю согласие на получение рассылок

**212****63.3**

Карма

Рейтинг

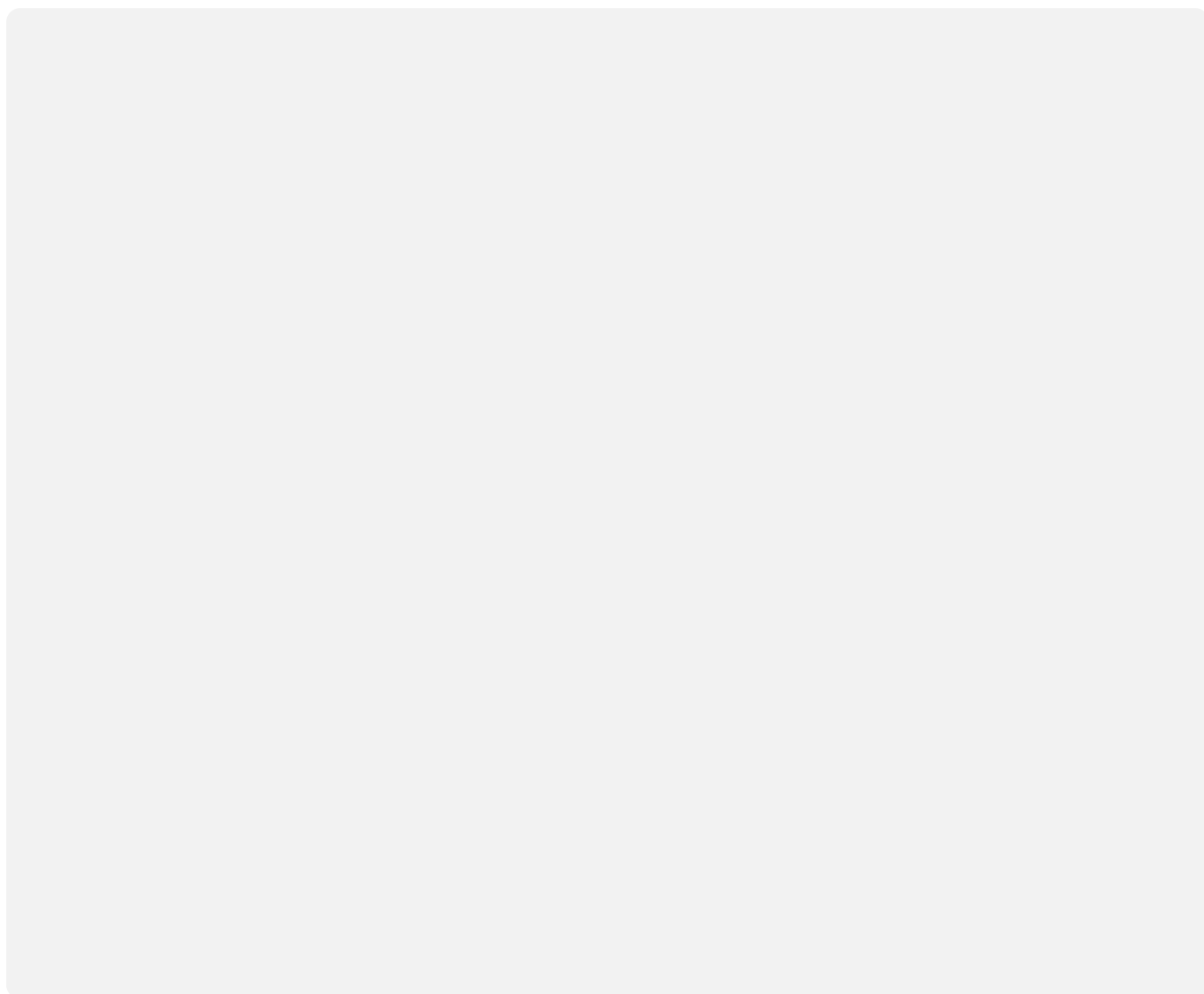
**Михаил Шардин** [@empenoso](#)

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

**Получи грант за код**

Конкурс open source проектов

[Сайт](#) [Сайт](#) [GitHub](#)



 [Комментарии 2](#)

## Публикации

[ЛУЧШИЕ ЗА СУТКИ](#) [ПОХОЖИЕ](#)



**Tirarex**

14 часов назад



**Получи грант за код**

Конкурс open source проектов

 Простой 9 мин 11K

Тutorial

 +41 83 34

Erwinmal

18 часов назад

## Сэндвич, сэр? История британских бутербродов от аристократических салонов до вокзальных буфетов

 Простой 13 мин 3K

Ретроспектива

 +41 20 2

oneastok

19 часов назад

## Умное зеркало на Raspberry Pi: пошаговое руководство

 Простой 4 мин 4.5K

Обзор

Перевод

 +22 56 14

iLushkersky

14 часов назад

## Жизнь на Марсе? (снова)

 Простой 3 мин 2.3K +17 5 10

TrexSelectel

16 часов назад

## Nintendo Virtual Boy: неожиданное возрождение виртуальной реальности из 90-х



Получи грант за код

Конкурс open source проектов

 +14 3 3

mio\_anni

19 часов назад

## От мини-ЭВМ и перфокарт к IDE и фреймворкам. Как поменялось программирование за 50 лет — взгляд изнутри

 12 мин  1.9K +12 15 35

RED\_OS\_M

18 часов назад

## Станислав Петров: «Ключевые отличия РЕД ОС М от Android – вовсе не в интерфейсе»

 Средний  8 мин  6.5K

Интервью

 +10 10 43

Albert\_Wesker

19 часов назад

## Миф о быстром и медленном пути выполнения программы

 Средний  11 мин  1.5K

Обзор

Перевод

 +9 16 0

kilokanat

5 часов назад

## Механическая клавиатура LARKeyboard

 Простой  5 мин  519

Тutorial



Получи грант за код

Конкурс open source проектов



beeline\_cloud

10 часов назад

## Научный «дипфейк»? Как галлюцинации нейросетей — и другие проблемы — просачиваются в академические статьи



Простой



8 мин



888

Аналитика



+8



11



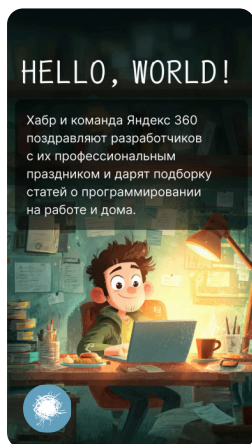
2

## «Раньше играл в Танки, сейчас веду кибертурниры» — истории современных студентов

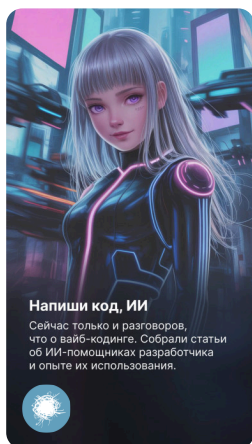
Турбо

Показать еще

### ИСТОРИИ



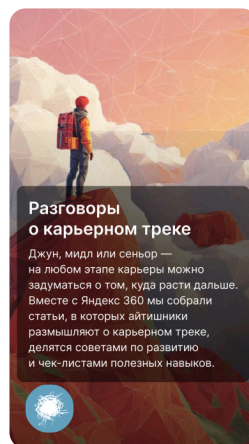
Чай, торт и код: с Днём программиста!



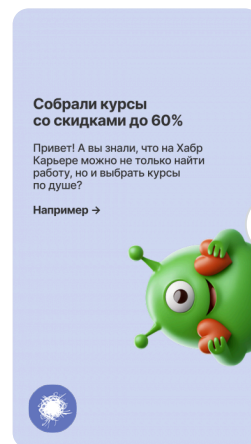
Made in AI



Чего хотят лиды в бигтехе?



Как расти в IT: советы, гайды и опыт сеньоров



Курсы со скидками до 60%

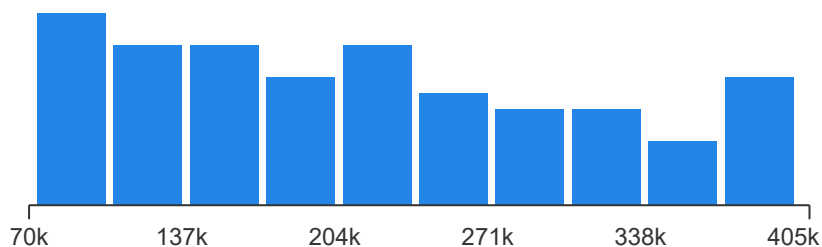
### СРЕДНЯЯ ЗАРПЛАТА В IT



Получи грант за код

Конкурс open source проектов

— средняя зарплата во всех IT-специализациях по данным из 27 443 анкет, за 2-ое пол. 2025 года. Проверьте «в рынке» ли ваша зарплата или нет!



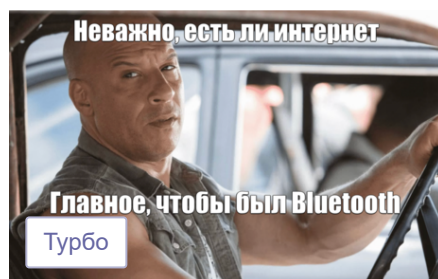
[Проверить свою зарплату](#)

#### МИНУТОЧКУ ВНИМАНИЯ



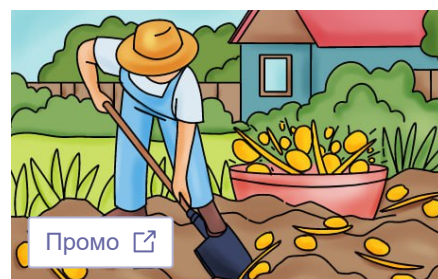
Событие

Посмотри в Календарь, вдруг сегодня есть мероприятие?



Турбо

Bluetooth против плохой связи: кейс каршеринга



Промо

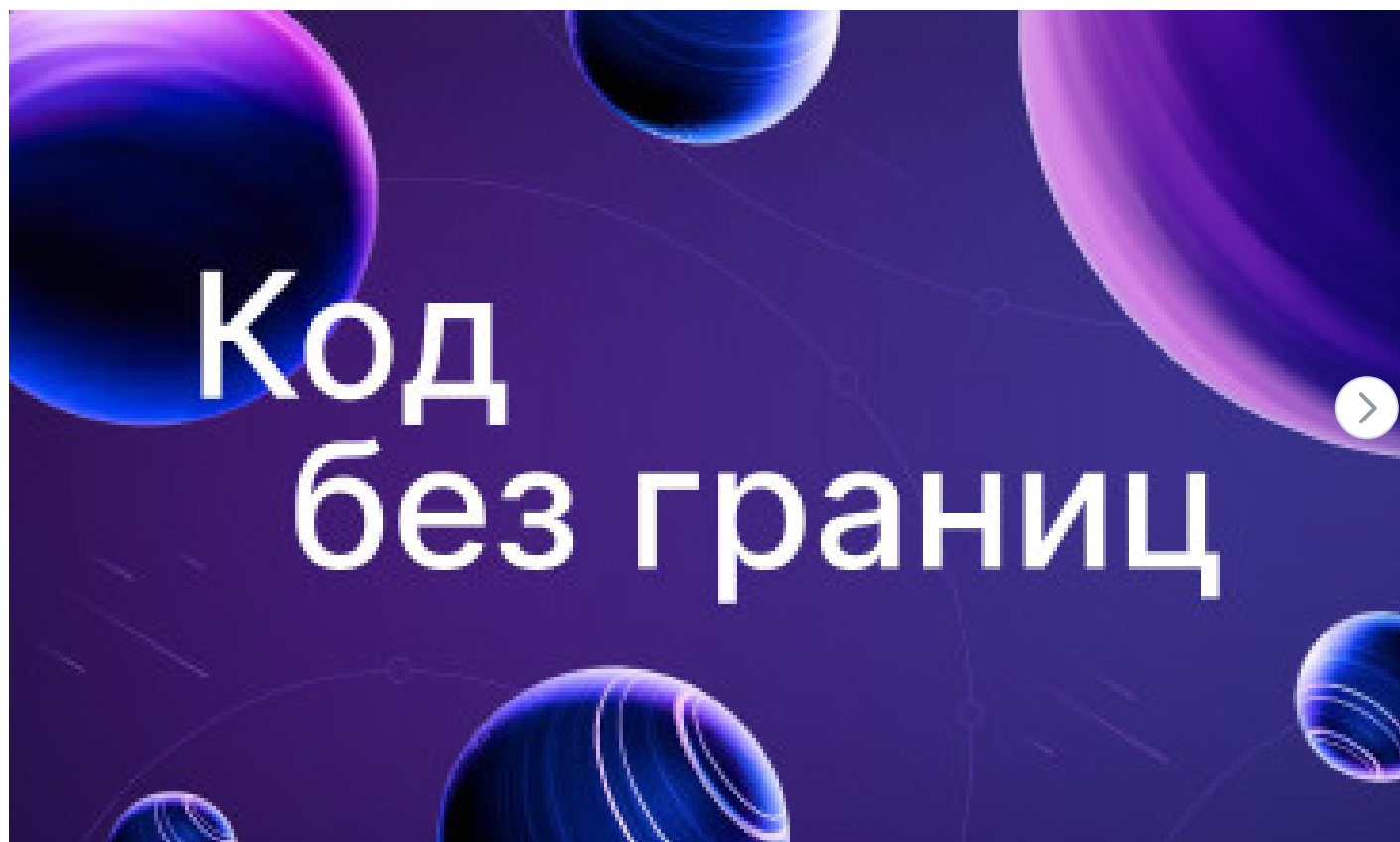
Не нужно копать картошку — хватит скидок в Промокодусе

#### БЛИЖАЙШИЕ СОБЫТИЯ



**Получи грант за код**

Конкурс open source проектов



3 сентября – 31 октября

## Программа грантов для развития open source проектов «Код без границ»

Онлайн

Разработка

Больше событий в календаре

Хабр



Получи грант за код

Конкурс open source проектов



Техническая поддержка

© 2006–2025, Habr



**Получи грант за код**

Конкурс open source проектов