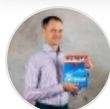


Во время посещения сайта вы соглашаетесь с использованием файлов [cookie](#)

Хорошо



Михаил Шардин ★

личный блог



27 ноября 2024, 04:35

+ Подписаться

## Отслеживание позиций торгового робота Московской биржи через CSV файл

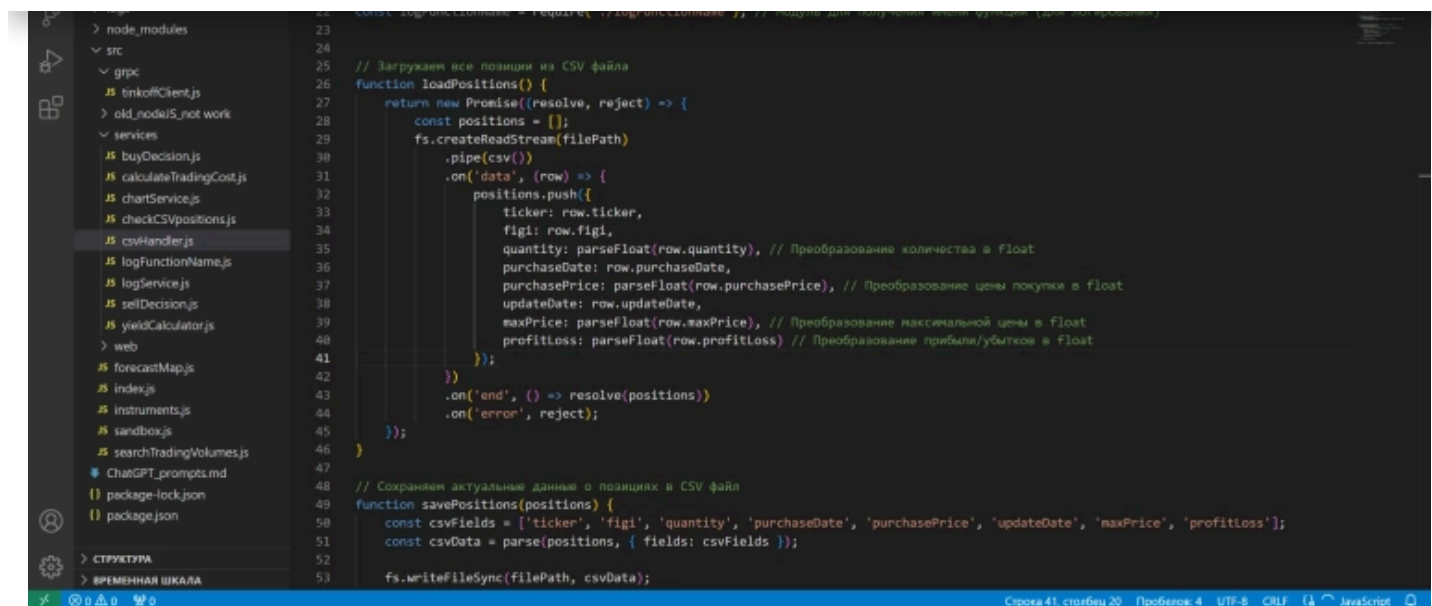
Нахожусь в процессе написания механизма торгового робота, работающего на Московской бирже через API одного из брокеров. Брокеров имеющих своё АПИ для МосБиржи катастрофически мало — мне известно только о трёх. При этом, когда я стал публиковать модули робота (и полностью [выложу готовый механизм робота 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](#).

Откройте счёт в ВТБ Мои Инвестиции

Введите текст комментария



```
23 // Загружаем все позиции из CSV файла
24
25 function loadPositions() {
26   return new Promise((resolve, reject) => {
27     const positions = [];
28     fs.createReadStream(filePath)
29       .pipe(csv())
30       .on('data', (row) => {
31         positions.push({
32           ticker: row.ticker,
33           figi: row.figi,
34           quantity: parseFloat(row.quantity), // Преобразование количества в float
35           purchaseDate: row.purchaseDate,
36           purchasePrice: parseFloat(row.purchasePrice), // Преобразование цены покупки в float
37           updateDate: row.updateDate,
38           maxPrice: parseFloat(row.maxPrice), // Преобразование максимальной цены в float
39           profitLoss: parseFloat(row.profitLoss) // Преобразование прибыли/убытков в float
40         });
41       })
42       .on('end', () => resolve(positions))
43       .on('error', reject);
44   });
45 }
46
47 // Сохраняем актуальные данные о позициях в CSV файл
48 function savePositions(positions) {
49   const csvFields = ['ticker', 'figi', 'quantity', 'purchaseDate', 'purchasePrice', 'updateDate', 'maxPrice', 'profitLoss'];
50   const csvData = parse(positions, { fields: csvFields });
51   fs.writeFileSync(filePath, csvData);
52 }
53
```

## Проблемы с записью позиций в Node.js

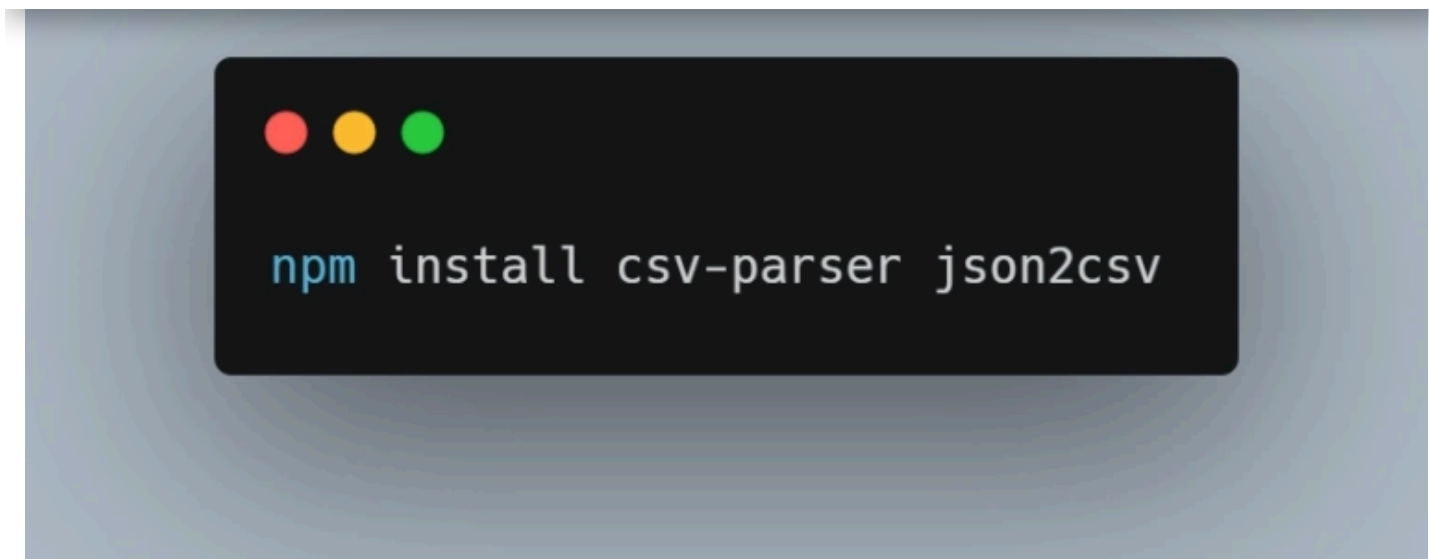
Вообще именно этот модуль пришлось пару раз переписывать, потому что не смог сразу отладить его. Проблема была в том, что вызов модуля записи и обновления позиций осуществлялся сразу из нескольких мест и одни результаты перезаписывали другие. Но удалось разобраться и теперь всё протестировано и работает.

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:

Откройте счёт в ВТБ Мои Инвестиции



## 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.

1. 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

Откройте счёт в ВТБ Мои Инвестиции

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.
1. Экспортированные модули: функции `loadPositions`, `updatePosition` и `removePosition` для использования в других частях робота.

**Полный код `csvHandler.js`:**

Откройте счёт в ВТБ Мои Инвестиции

```
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 (для логирования)

// Загружаем все позиции из CSV файла
function loadPositions() {
  return new Promise((resolve, reject) => {
    const positions = [];
    fs.createReadStream(filePath)
      .pipe(csv())
      .on('data', (row) => {
        positions.push({
          ticker: row.ticker,
          figi: row.figi,
          quantity: parseFloat(row.quantity), // Преобразование количества в float
          purchaseDate: row.purchaseDate,
          purchasePrice: parseFloat(row.purchasePrice), // Преобразование цены покупки в float
          updateDate: row.updateDate,
          maxPrice: parseFloat(row.maxPrice), // Преобразование максимальной цены в float
          profitLoss: parseFloat(row.profitLoss) // Преобразование прибыли/убытков в float
        });
      })
      .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', 'maxPrice', 'profitLoss'];
  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);
    } else {
      // Обновляем, если позиция уже существует
      positions[index] = newPosition;
    }

    savePositions(positions);
  });
}
```

Откройте счёт в ВТБ Мои Инвестиции

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.

### 1. 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.

### 1. Обработка ошибок

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

## Основные функции:

### 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.

Откройте счёт в ВТБ Мои Инвестиции

- Локальные позиции загружаются из CSV-файла.
  - Позиции сервера извлекаются через API Tinkoff.
1. Обнаружение расхождений: Для каждой позиции в CSV-файле:
  2. Код вычисляет общее количество в лотах (`csvPosition.quantity * lotSize`).
  3. Сравнивает с балансом на сервере. Ошибки регистрируются, если:
  4. Количества не совпадают.
  5. Позиция в CSV-файле отсутствует на сервере.
1. 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`:

Откройте счёт в ВТБ Мои Инвестиции



```

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 API Tinkoff
Invest
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', accountId);

    // Логирование полученных позиций с сервера
    logger.info('Все открытые позиции счета ${secrets.AccountID}: \n ${JSON.stringify(response,
null, '\t')}\n\n');

    // Возвращаем только позиции с ценными бумагами (securities)
    return response.securities.map(sec => ({
      figi: sec.figi,
      balance: parseFloat(sec.balance) // Преобразуем баланс в float
    }));
  } catch (error) {
    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: ${csvTotal},
Сервер: ${serverPosition.balance}');
          throw new Error('Найдено несоответствие позиций. Остановка торговли.');
```

Откройте счёт в ВТБ Мои Инвестиции



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

## Итоги

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

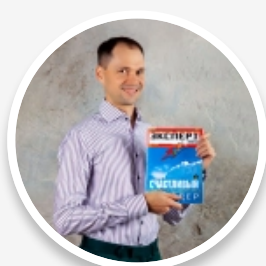
Автор: [Михаил Шардин](#)

27 ноября 2024 г.

торговые роботы

6.6K

11 12 21



**Михаил Шардин**

Пермь

257 2 502

с 23 января 2019

+HreHDn1F5CZjN...

+ Подписаться

12 КОММЕНТАРИЕВ

Сначала старые



**T-800**

27 ноября 2024, 06:50



В Квике можно не только через текстовые \*.tri файлы работать, но и через Trans2Quik.dll  
А вообще, удачи в вашем начинании.

+1



**akumidv**

27 ноября 2024, 07:39



Откройте счёт в ВТБ Мои Инвестиции

Сохранение не потокобезопасное — будут ошибки при синхронной записи и считать. Очередь поможет.

И новые функции лучше разрабатывать через тесты — тестирование позволит делать менее связанный код, его потом легче поддерживать и модифицировать. Ну а тесты позволят прогонять функционал и быть более уверенным, что работает.

Из простых решений еще можно ESLint добавить в проект — он будет помогать ошибки находить и стиль поддерживать.

Сам JS для бота сложный выбор. Проще питон подучить и на нем делать. Там хотя бы pandas и numba с Jit копиляцией есть и мультипроцессорность понятная.

PS логи и данные в проекте — зло.

**IliaM**

27 ноября 2024, 07:43



QUIK конечно поразительная программа, в эпоху санкций, работающая на Windows. За 20 лет, могли бы и кроссплатформенность добавить. А вам удачи.

**SergeyJu**

27 ноября 2024, 10:07



У квика всегда были конкуренты, которые так и не распространились.

У гуты-банка, у Цериха, у Атона, у Альфы, у Айти-инвеста, был еще нет-трейдер и это неполный список. И, насколько я знаю, из под линукса на квике торгуют.

Что касается авторского проекта, если рассматривать его как рабочий макет, то все здорово. Осталось понять косяки, учесть принципиальные недостатки и написать заново.

Автор, как я понимаю, пишет быстро, так что все в его руках.



— Показать 3 ответа

Ещё 3 комментария

Откройте счёт в ВТБ Мои Инвестиции



ОТПРАВИТЬ

## Читайте на SMART-LAB

ЗА ПОЛУГОДИЕ, ЗАКОНЧИВШЕЕСЯ 30 ИЮНЯ 2025 ГОДА (НЕАУДИРОВАННЫЙ)

(в миллионах российских рублей, за исключением прибыли на акции)

	Примечание	За полугодие, закончившееся	
		30 июня 2025 года	30 июня 2024 года
ВЫРУЧКА	15	171 233	201 884
СЕБЕСТОИМОСТЬ РЕАЛИЗАЦИИ	17	(144 244)	(160 257)
<b>ВАЛОВАЯ ПРИВЛЕЧЬ</b>		<b>26 989</b>	<b>41 627</b>
Коммерческие, общесистемные и административные расходы	18	(35 722)	(38 177)
Прочие операционные доходы	19	1 363	385
Прочие операционные расходы		(837)	(115)
Изменение справедливой стоимости инвестиционной недвижимости		328	-
<b>ОПЕРАЦИОННЫЙ УБЫТОК (ПРИВЛЕЧЬ)</b>		<b>(7 559)</b>	<b>3 720</b>
Финансовые доходы	20	585	443
Финансовые расходы	21	(26 432)	(14 573)
<b>УБЫТОК ДО НАЛОГА НА ПРИВЛЕЧЬ</b>		<b>(33 686)</b>	<b>(12 394)</b>
Доходы по налогу на прибыль		8 527	2 036
<b>Итого чистый убыток и общий совокупный убыток на капитал</b>		<b>(25 159)</b>	<b>(10 358)</b>

## М.Видео - есть ли в мультивселенной компания, где миноритариям что-то достанется?

М.Видео меняет формат допэмиссии. Ранее акционеры одобрили допэмиссию до 500 млн новых акций по закрытой подписке — о...



Mozgovik

03:51

## ДАЙДЖЕСТ ПО РЕЙТИНГОВЫМ ДЕЙСТВИЯМ В ВЫСОКОДОХОДНОМ СЕКТОРЕ, ПОРТФЕЛЕ PROBONDS ВДО И РОЗНИЧНЫХ ИНВЕСТИЦИОННЫХ ОБЛИГАЦИЙ ЗА...

● ПАО «ЕВРОТРАНС» АКРА подтвердило кредитный рейтинг на уровне A-(RU) ПАО «ЕвроТранс» — один из крупнейших независимых топливных операторов на рынке Московского региона....



Иволга Капитал

13.09.2025

## Банк России снизил ключевую ставку до 17% — что делать? Мнение аналитиков

Сегодня на заседании Банк России снизил ключевую ставку на 1 п.п., до 17%. Теоретически, для рынков акций и облигаций — это определенно благоприятное событие, но на практике рынок...



БКС Мир инвестиций

13.09.2025

Установите приложение Смартлаба:



RuStore



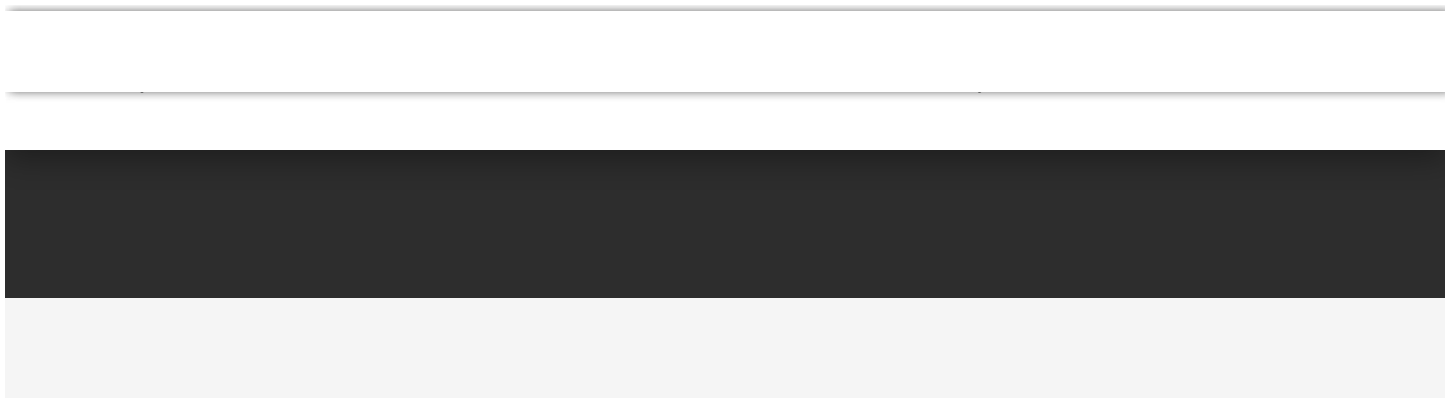
AppGallery



App Store



Откройте счёт в ВТБ Мои Инвестиции



Откройте счёт в ВТБ Мои Инвестиции