

**Михаил Шардин**

7 ОКТ

Подписаться

Как я решил вернуться в алготрейдинг после 10 летнего перерыва

Самое главное, что никакого отношения к слову алко это не имеет. В течении нескольких лет я торговал в связке программы технического анализа AmiBroker + торговый терминал QUIK через .tri и .trr файлы в основном на фьючерсах на срочном рынке. Не могу сказать что это было неудачным опытом, но со временем я узнал про распределение активов (Asset allocation) и понял что очень сложно соревноваться с бенчмарком в виде фондового индекса.

В теории это означает что можно купить индекс и забыть об этом, заниматься своими делами, бизнесом, семьёй - а индекс растёт (ну или падает, смотря какое время) и для этого не нужно прикладывать никаких действий. А за связкой AmiBroker + QUIK постоянно нужно было присматривать, следить не отвалился ли адаптер импорта через .tri файл. А ещё иметь несколько виртуальных машин с установленными копиями Windows на каждой виртуалке на одном физическом компьютере для разных брокеров. Возможна была установка только одной пары AmiBroker + QUIK на одну винду. Всё это мне не особо нравилось.

В 2024 году захотелось что-то лёгкое - без Windows и современное - через API интерфейс. Желательно бесплатное для пользователя. Несколько лет назад я уже пытался узнать появились ли у российских брокеров API для работы с ними, но так и не собрался. Этой осенью я

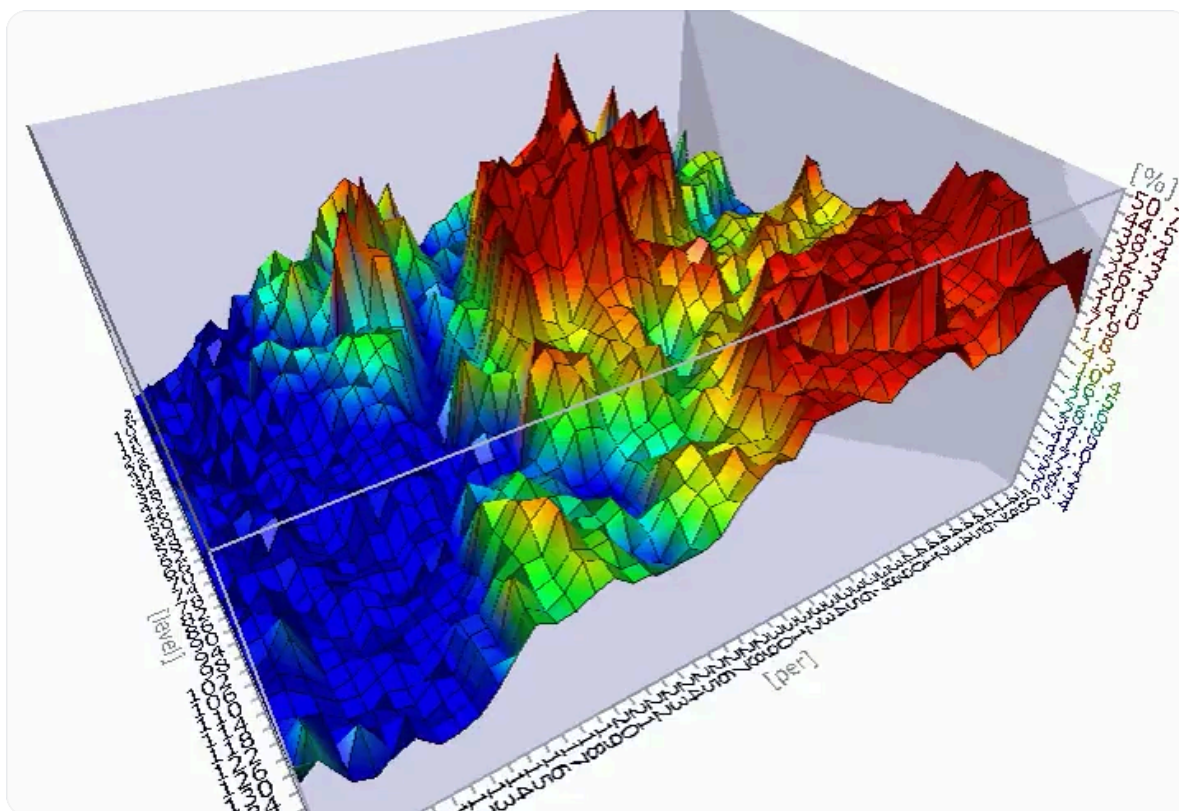
работы с физлицами. Не смог найти никакой сводной таблицы и нашёл только три варианта:

- [ФИНАМ](#)
- [Алор](#)
- [Тинькофф Инвестиции](#)
- (на полноту сведений не претендую и буду рад узнать ещё варианты брокеров с АПИ)

По субъективным причинам я выбрал работать с T-Bank Invest API (это бывший Тинькофф) через среду выполнения JavaScript Node.JS.

Немного моей истории, 2008-2012 годы: AmiBroker + QUIK

AmiBroker – это платформа для технического анализа, которая позволяет пользователям создавать свои собственные индикаторы и скрипты. Она также может интегрироваться с другими программами, такими как MetaTrader, NinjaTrader и даже с торговыми терминалами, включая QUIK.



AmiBroker 3D Optimization Chart

В AmiBroker'e мне очень нравилась его функция 3D Optimization Chart. На мой взгляд она позволяла избежать перепогонки показателей под кривую истории. Оптимизация в бэк-тестере поддерживалась функцией `optimize`. Синтаксис этой функции был следующий:

```
переменная = optimize("Описание", default , min , max , step );
```

`переменная` - это обычная переменная их языка AFL, которой присваивается значение, возвращаемое функцией оптимизации. В режиме оптимизации функция `optimize` возвращает последовательные значения от минимума до максимума (включительно) с шагом. "Описание" - это строка, которая используется для идентификации переменной оптимизации и отображается как имя столбца в списке результатов оптимизации. `default` — это значение по умолчанию, которое оптимизирует возврат функции в режимах исследования, индикатора, комментария, сканирования и обычного бэктеста. `min` — минимальное значение оптимизируемой переменной. `max` — максимальное значение оптимизируемой переменной. `шаг` — это интервал, используемый для увеличения значения от минимума до максимума.

QUIK — это торгово-информационная система, предназначенная для предоставления участникам фондового рынка доступа к биржевым данным в реальном времени. Она используется многими брокерами и трейдерами для совершения операций на фондовых рынках.

Когда AmiBroker подключался к QUIK, он получал данные в реальном времени от торговой системы QUIK и использовал их для построения графиков, создания индикаторов и выполнения других аналитических задач. Пользователь мог также настроить систему так, чтобы она автоматически исполняла сделки через QUIK на основе определенных условий.

Для подключения AmiBroker к QUIK необходимо было установить соответствующий плагин или модуль. Это позволяло платформе AmiBroker получать данные от QUIK и обрабатывать их для дальнейшей работы.

Пример логов работы:

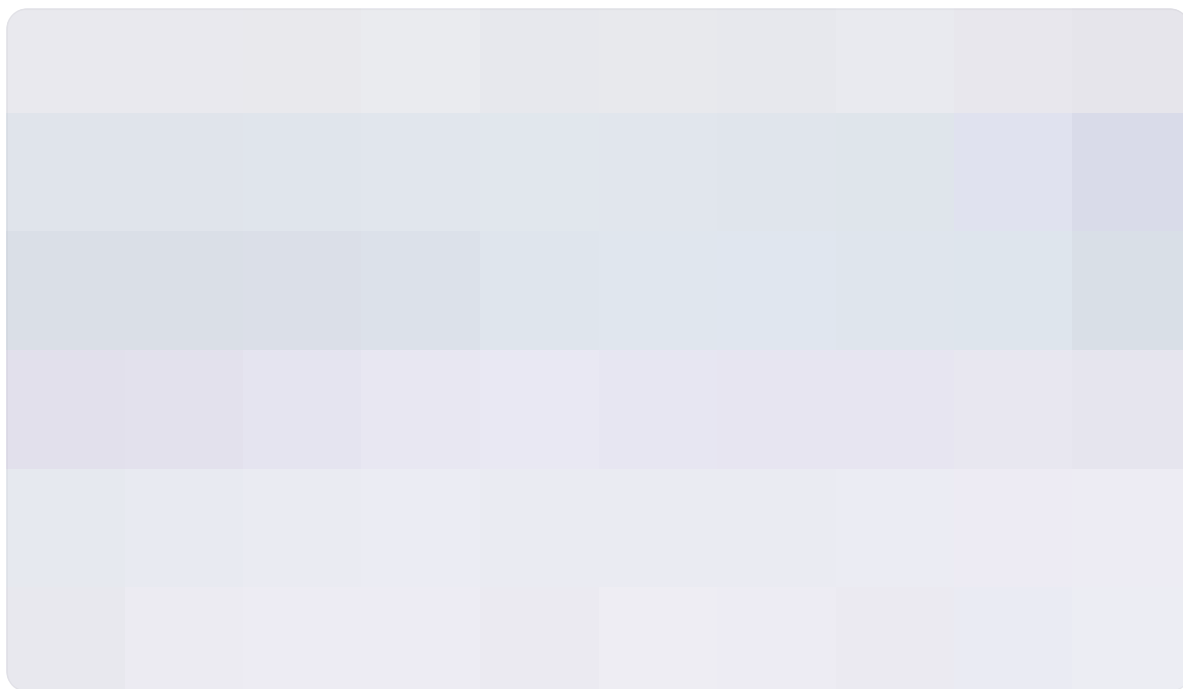
```
09.10.2009 11:00:31: [2208:2212] Получено уведомление о выполнении  
09.10.2009 14:04:30: [2208:1172] Получено уведомление об отправке тран  
09.10.2009 14:04:30: [2208:2212] Получено уведомление о выполнении тра  
09.10.2009 15:10:10: [2208:1172] Завершен процесс отправления транзакц
```

Насколько я понимаю AmiBroker (создан в Польше) уже несколько лет не обновляется: [последний выпуск в 2015 году, а обновления в 2017 году](#).

QUIКом я уже очень давно не пользовался, но по скриншотам на сайте я понял что программа внешне [за последние годы не особенно изменилась](#).

Ещё немного моей истории, 2008-2012 годы: КБ ПАУК и Technical Analysis of STOCKS & COMMODITIES

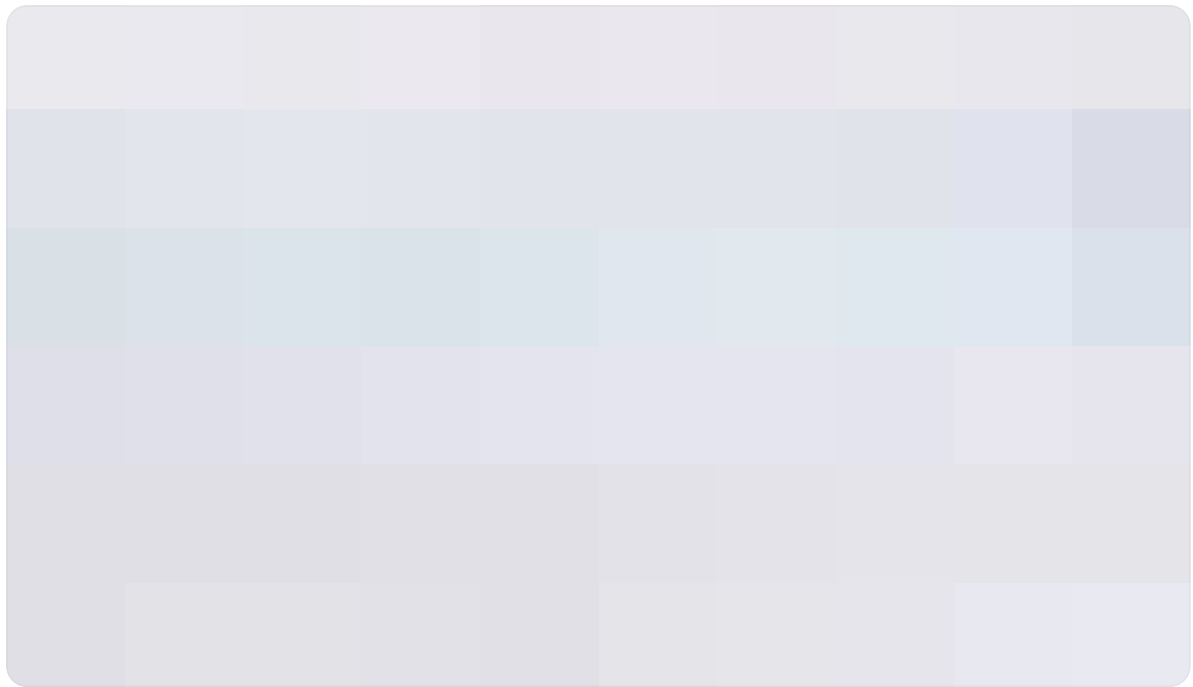
Раньше я был активным пользователем форума КБ ПАУК. Вот даже нашёл файл HTML файл с интересующей меня темой, который пролежал 20 лет:



Форум форекс кб паук <http://forex.kbpauk.ru/> (в 2024 уже не работает)

Я много изучал тему посвященную AmiBroker'у. И спрашивал там

Сейчас, в 2024 году форум недоступен, но если кто-то хочет ностальгии, то [нашёл его архивную копию](#).



Форум форекс кб паук <http://forex.kbpauk.ru/> (в 2024 уже не работает)

Правда архивная копия немного бесполезна, потому что самые интересные разделы открывались только после входа по логину и паролю.

2024 год, сейчас: T-Bank Invest API + неофициальный SDK Node.JS. Описание функций торгового робота



Примерная структура робота

Осенью 2024 года решил написать торгового бота, создать структуру программы которая будет работать через API брокера.

Торговый робот должен содержать:

- Список бумаг - с которым он будет работать.
- Условие покупки.
- Условие продажи.
- CSV файл учёта
- Управление деньгами - процент входа.
- Логирование всех действий.
- Отображение информации, скорее всего через веб-сервер.
- Иметь модуль бектестинга.

T-Bank Invest API + Node.JS:

Решил пока не лезть в срочный рынок и попробовать поработать с самими оборотистыми акциями на Московской бирже. Остаётся только их найти.

В этой статье я распишу как создал два модуля на Node.js:

- Взаимодействие с T-Bank Invest API `tinkoffClient.js`
- Модуль, который ищет акции с самым большим оборотом за три последних месяца `searchTradingVolumes.js`

Проект представлен на Гитхабе: <https://github.com/empenoso/SilverFir-TradingBot>.

SilverFir-TradingBot\src\grpc\tinkoffClient.js



Модуль tinkoffClient.js — это специализированный клиент, разработанный для взаимодействия с T-Bank Invest API, российской брокерской платформой. Основная цель этого модуля — предоставить боту на основе Node.js, доступ к финансовым данным для алгоритмической торговли или анализа рынка.

Вот его функциональность:

Основные функции:

- Авторизация и настройка:
- Модуль использует [API-токены для аутентификации](#).
- Он поддерживает как тестовую (песочницу), так и продовую среду, хотя по умолчанию он использует среду песочницы.
- Заголовки API включают необходимый токен авторизации и тип контента для JSON-коммуникации.
- Универсальный механизм вызова API:
- Метод callApi() служит гибкой утилитой для отправки POST-запросов на конечные точки API Tinkoff.
- Он обрабатывает отправку полезной нагрузки и отчеты об ошибках, предоставляя подробную обратную связь по сбоям (через журналы).
- Извлечение данных "японских свечей":
- Метод getCandles() извлекает данные свечей для заданного финансового инструмента (тикера) за указанные интервалы.
- Он вычисляет временной диапазон на основе предоставленного интервала и извлекает серию свечей из сервиса рыночных

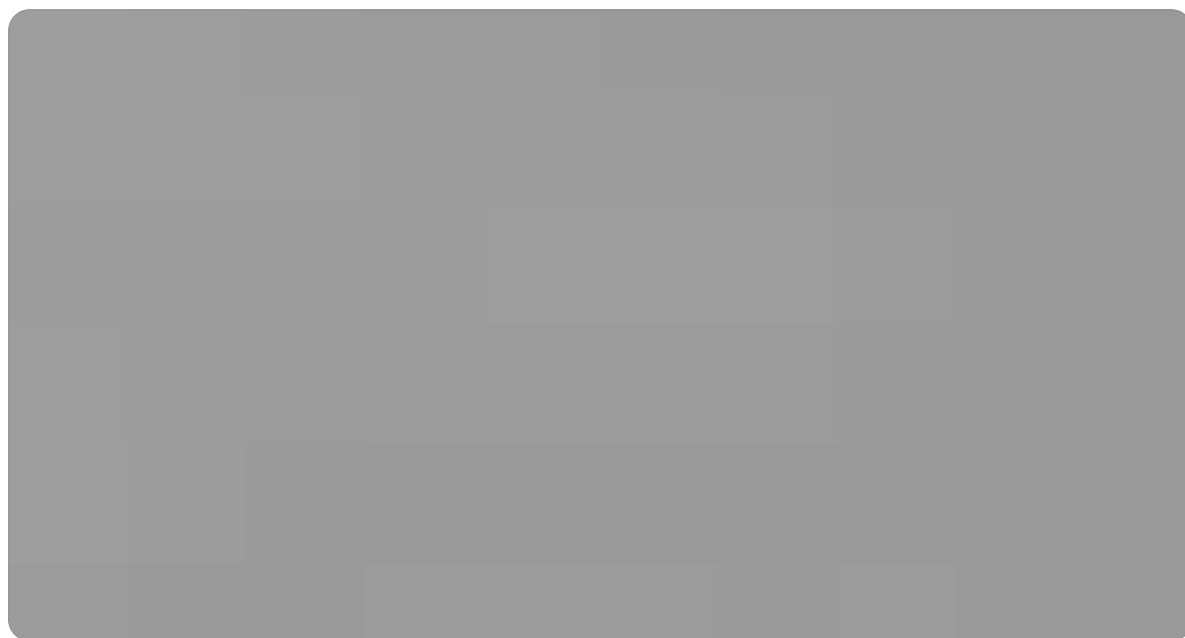
данных. Это необходимо для анализа рынка, технических индикаторов и торговых стратегий.

- Обработывает до 1000 свечей за запрос и обеспечивает регистрацию ошибок, если запрос не выполняется или возвращает пустые данные.

Зависимости и утилиты:

- Moment.js: Используется для форматирования дат и расчета временных диапазонов, что имеет решающее значение при работе с финансовыми данными за различные периоды времени.
- Axios: Обработывает HTTP-запросы к конечным точкам API.
- Службы ведения журналов: реализует настраиваемую систему журналов для вывода файлов и консоли, помогая в отладке и отслеживании операций.

SilverFir-TradingBot\src\searchTradingVolumes.js



Модуль searchTradingVolumes.js является частью SilverFir-TradingBot, разработан для определения акций на Московской бирже (MOEX) с наибольшими объемами торгов за последние три месяца. Нужен чтобы отбросить неликвид.

Основные функции:

- Фильтрация акций с MOEX (Московской биржи):
- Модуль использует T-Bank Invest API для получения полного списка доступных акций.
- Фильтрует результаты, чтобы сосредоточиться конкретно на акциях, торгуемых на REAL_EXCHANGE_MOEX, гарантируя, что будут рассмотрены только акции Мосбиржи.
- Отфильтрованный список включает в себя важную информацию, такую как FIGI (глобальный идентификатор финансового инструмента), тикер, ISIN и название, которые затем регистрируются для отслеживания и анализа.
- Извлечение исторических данных по объему торгов:
- Для каждой акции, идентифицированной на этапе фильтрации, модуль извлекает ежедневные данные свечей (OHLCV - открытие, максимум, минимум, закрытие, объем) с помощью T-Bank Invest API.
- Он извлекает данные за последние три месяца, рассчитывая общий объем торгов за этот период.
- Этот шаг имеет решающее значение для измерения рыночной активности и определения наиболее активно торгуемых акций.
- Определение 15 лучших акций по объему:
- После сбора данных по объему торгов модуль ранжирует все акции на основе их общего объема торгов за трехмесячный период.
- Затем он выбирает 15 лучших акций, которые считаются имеющими самый высокий оборот, что делает их наиболее активными на рынке.
- Ведение журнала и подготовка конфигурации:
- 15 лучших акций регистрируются с соответствующими тикерами и идентификаторами FIGI для дальнейшего использования.

- Эти данные также подготавливаются для включения в конфигурацию бота, что позволяет легко интегрировать их в другие части торговой стратегии.

Зависимости и утилиты:

- Moment.js: используется для манипулирования датами, в частности для генерации временных диапазонов (три месяца) для запросов исторических данных.
- TinkoffClient: это выделенный клиент бота для взаимодействия с API Tinkoff Invest, облегчающий коммуникацию, необходимую для получения данных по акциям и свечам.
- Служба пользовательского ведения журнала: модуль использует систему ведения журнала, которая регистрирует информацию как в консоли, так и во внешних файлах, помогая отслеживать фильтрацию акций, расчеты объема и любые ошибки, возникающие во время взаимодействия с API.

Результат выполнения скрипта searchTradingVolumes.js

```
2024-10-06 07:25:48 [INFO]: Отфильтрованных акций 170 штук.
2024-10-06 07:25:50 [INFO]: Топ 15 акций по объему за последние три ме
{
  "ticker": "GRNT",
  "figi": "TCS10A0JV532",
  "totalVolume": 7092179306
},
{
  "ticker": "VTBR",
  "figi": "BBG004730ZJ9",
  "totalVolume": 5906896857
},
{
  "ticker": "MTLR",
  "figi": "BBG004S68598",
  "totalVolume": 5047115732
},
{
```

```
    "totalVolume": 4423660125
  },
  {
    "ticker": "RNFT",
    "figi": "BBG00F9XX7H4",
    "totalVolume": 1775061013
  },
  {
    "ticker": "EUTR",
    "figi": "TCS00A1002V2",
    "totalVolume": 1270913208
  },
  {
    "ticker": "SNGSP",
    "figi": "BBG004S681M2",
    "totalVolume": 1092911599
  },
  {
    "ticker": "GAZP",
    "figi": "BBG004730RP0",
    "totalVolume": 1027429605
  },
  {
    "ticker": "ROSN",
    "figi": "BBG004731354",
    "totalVolume": 983187493
  },
  {
    "ticker": "SBER",
    "figi": "BBG004730N88",
    "totalVolume": 940111321
  },
  {
    "ticker": "SGZH",
    "figi": "BBG0100R9963",
    "totalVolume": 929574444
  },
  {
    "ticker": "AFLT",
    "figi": "BBG004S683W7",
    "totalVolume": 632806638
  },
  },
```

```
    "figi": "TCS00A106YF0",
    "totalVolume": 595704137
  },
  {
    "ticker": "RUAL",
    "figi": "BBG008F2T3T2",
    "totalVolume": 542242519
  },
  {
    "ticker": "TATN",
    "figi": "BBG004RVFFC0",
    "totalVolume": 538004844
  }
]
```

2024-10-06 07:25:50 [INFO]:

Вставка в `config.js`:

2024-10-06 07:25:50 [INFO]:

`securitiesToMonitorTikerArray`: ["GRNT", "VTBR", "MTLR", "UWGN", "RNFT", "EU

2024-10-06 07:25:50 [INFO]:

`securitiesToMonitorFigiArray`: ["TCS10A0JV532", "BBG004730ZJ9", "BBG004S6

2024-10-06 07:25:50 [INFO]:

Для использования в скрипте `download_md.sh` в одну колонку:

TCS10A0JV532

BBG004730ZJ9

BBG004S68598

BBG008HD3V85

BBG00F9XX7H4

TCS00A1002V2

BBG004S681M2

BBG004730RP0

BBG004731354

BBG004730N88

BBG0100R9963

BBG004S683W7

TCS00A106YF0

BBG008F2T3T2

BBG004RVFFC0

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

Модуль tinkoffClient.js для T-Bank Invest API представлен пока только частично.

Автор: [Михаил Шардин](#)

7 октября 2024 г.



33



1 комментарий



Комментарий...



**Михаил Шардин**

5 нояб

[Подписаться](#)

Тестирование торгового робота на Московской бирже в режиме «песочницы»

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



150

РЕКЛАМА [РАЗМЕСТИТЬ](#)

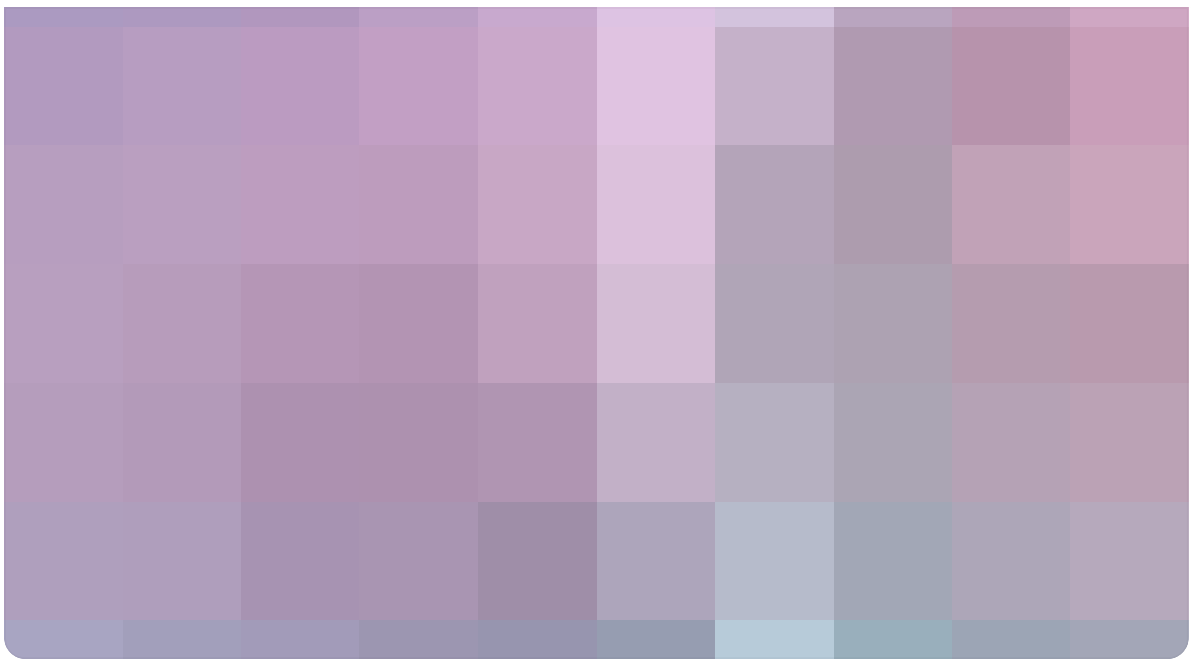
**LIGHT CASH | Инвестиции крипто**

Крипто 30 окт

[Подписаться](#)

Новый ИИ-бот заработал \$1,88 млн за 5 часов

В криптосообществе продолжает набирать популярность тренд на торговлю через ботов, построенных на базе искусственного интеллекта (ИИ). Такие программы помогают своим создателям находить перспективные монеты для инвестиций. Также боты могут автоматизировать торговлю



1.2K



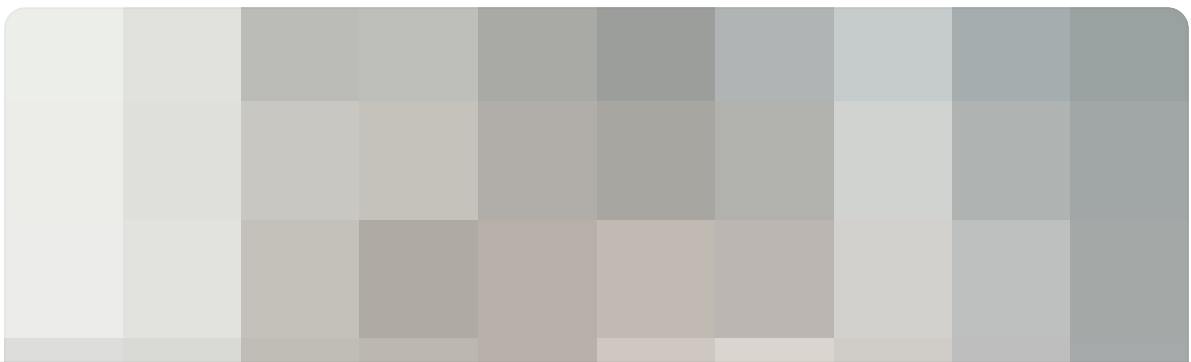
Нейросети - это просто

AI 8 нояб

Подписаться

Генерация изображений в нейросети онлайн. Двойная экспозиция

Рассмотрим варианты генераций изображений в нейросетях онлайн с использованием техники двойной экспозиции. Этот интересный стиль будет полезен как новичкам, так и опытным экспертам. Будем создавать изображения с использованием нейронных сетей, включая **Flux, Midjourney, Dalle 3** и **Stable Diffusion**.





1



2



1.4K

**Алмаз Сираев**

30 окт

Подписаться

Как разработать Telegram бота без вложений?

В этой статье хочу поделиться своим опытом разработки Telegram бота и проблемами, с которыми я столкнулся при разработке. Моя цель - создание бота без финансовых вложений на разработку и дальнейшую поддержку. Сейчас расскажу, что из этого вышло.



1



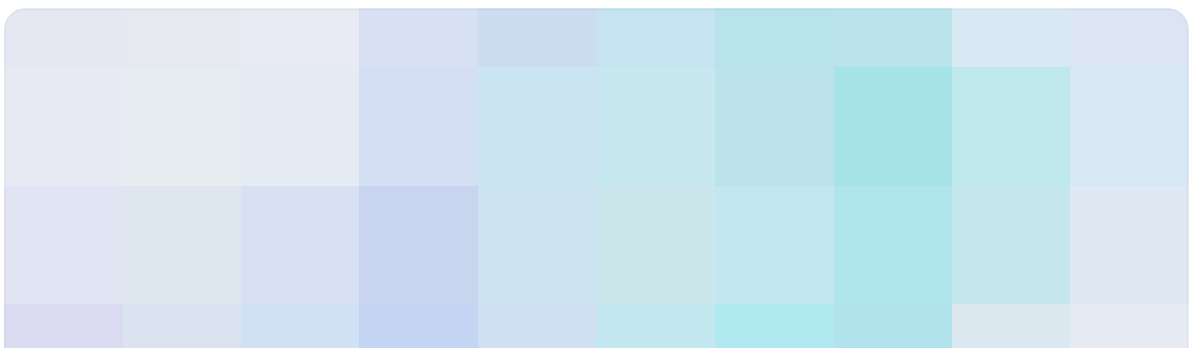
219

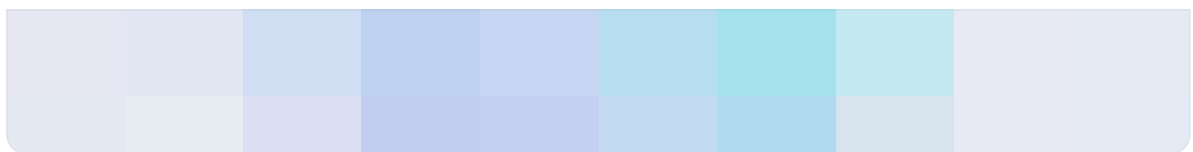
**Виктор Грек**

Сервисы 30 окт

Подписаться

Инвестирование в акции с помощью Chatgpt: как использовать ChatGPT для анализа компаний (с подсказками!)





❤️ 1 +

🔍 📌 ➡️ 🎁

👁️ 547



Просто Бонд 💎

Деньги 9 окт

Подписаться

Комиссии Мосбиржи 2024 – за что платим и как уменьшить



❤️ 2 +

🔍 📌 1 ➡️ 🎁

👁️ 397

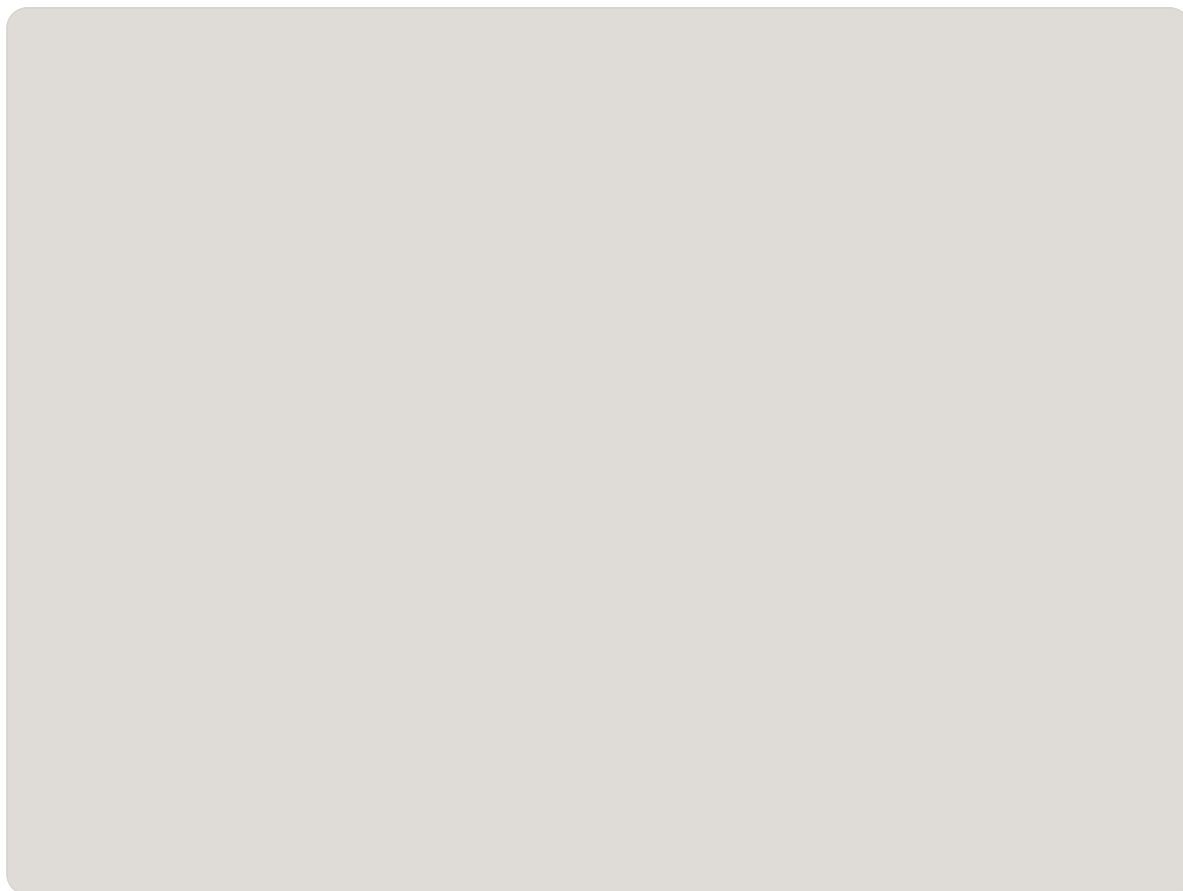


beekoka

Полный курс 7 окт

Подписаться

Как страх за свое будущее подтолкнул меня создать бота-помощника для инвестиций



3



2



1



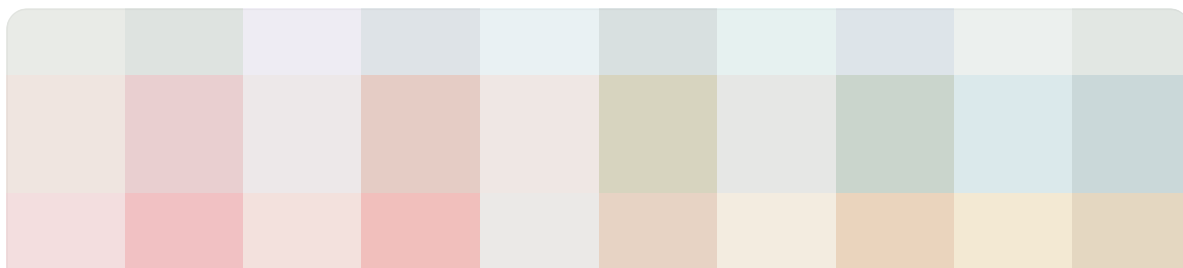
722

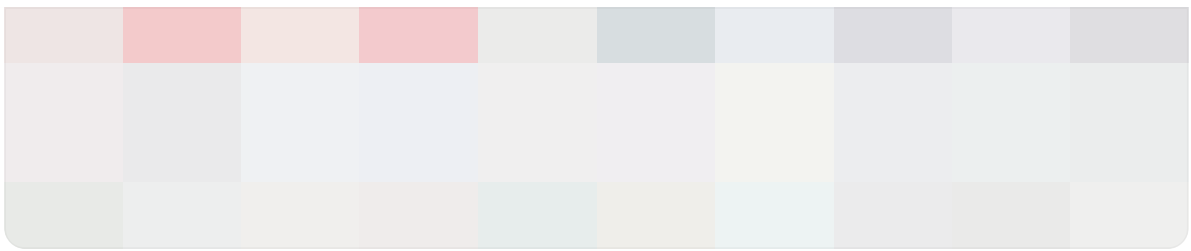
**Цифровой Кочевник Инвестор**

Инвестиции 25 цент

[Подписаться](#)

Пошла популяризация пассивных инвестиций от брокеров





❤️ 2 +

🔍 📌 ➡️ 🎁

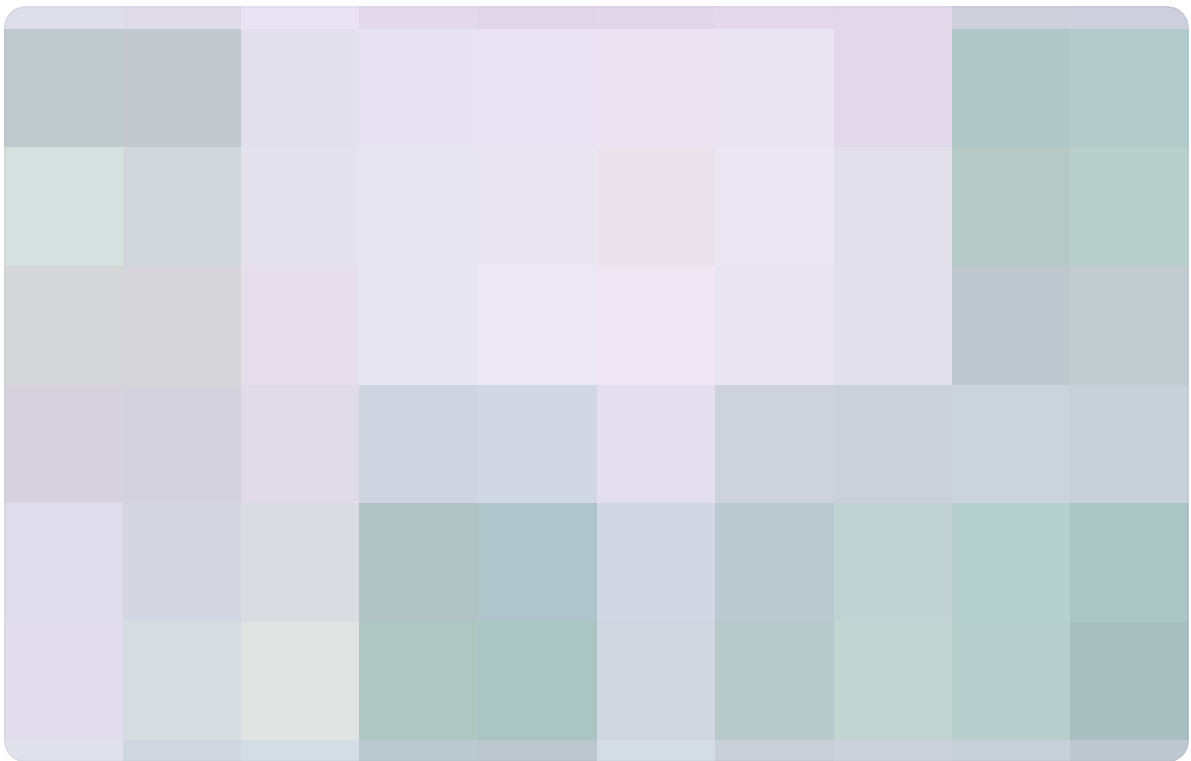
👁️ 776



Лена Никулина. О чат-ботах
Маркетинг 10 цент

Подписаться

Кейс: многофункциональный чат-бот для инвестиционного сообщества. Калькулятор портфеля, тест на экономический профиль, рейтинговая таблица и другие функции



🔥 2 ❤️ 1 +

🔍 4 📌 3 ➡️ 🎁

👁️ 1.7K

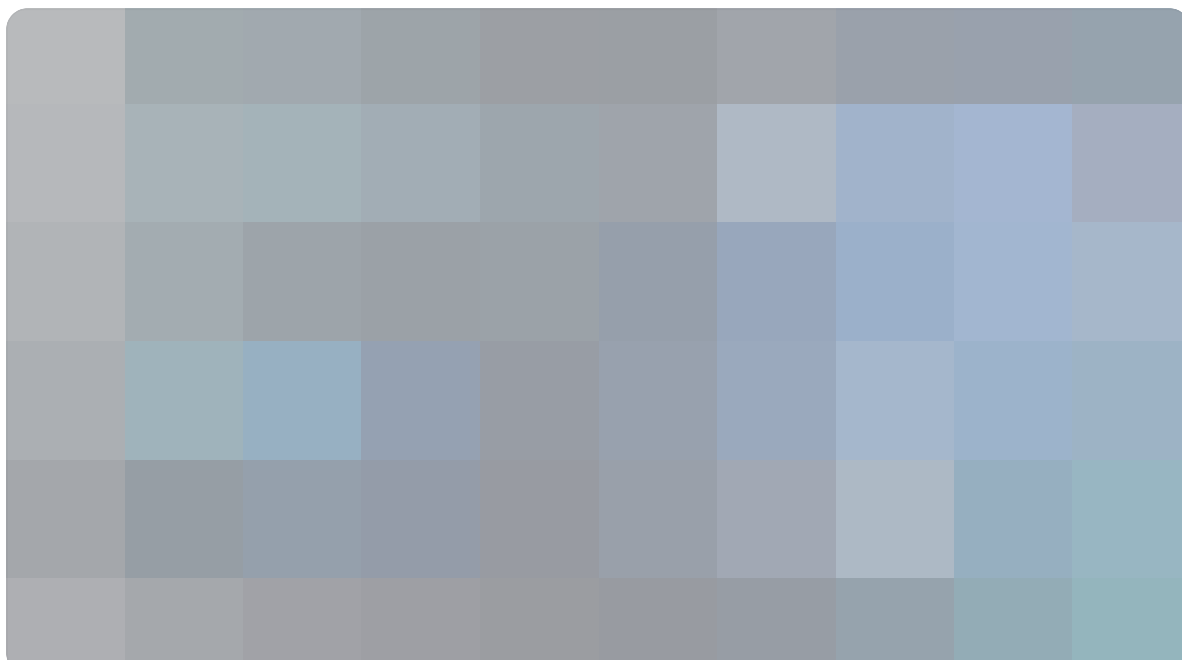
РЕКЛАМА [РАЗМЕСТИТЬ](#)... 

Ян Лазаренок
Дизайн 29 авг

[Подписаться](#)

Многостраничный сайт с личным кабинетом для авторской платформы BUY or SELL — система ИИ-помощников для автоматизации трейдинга и буста финансового результата x10

Как мы сформировали спрос на инновационный для финансового рынка [продукт](#) и сделали интеллектуальных помощников рутинной частью лайфстайла тысяч начинающих и опытных трейдеров





8



1



2



8



589

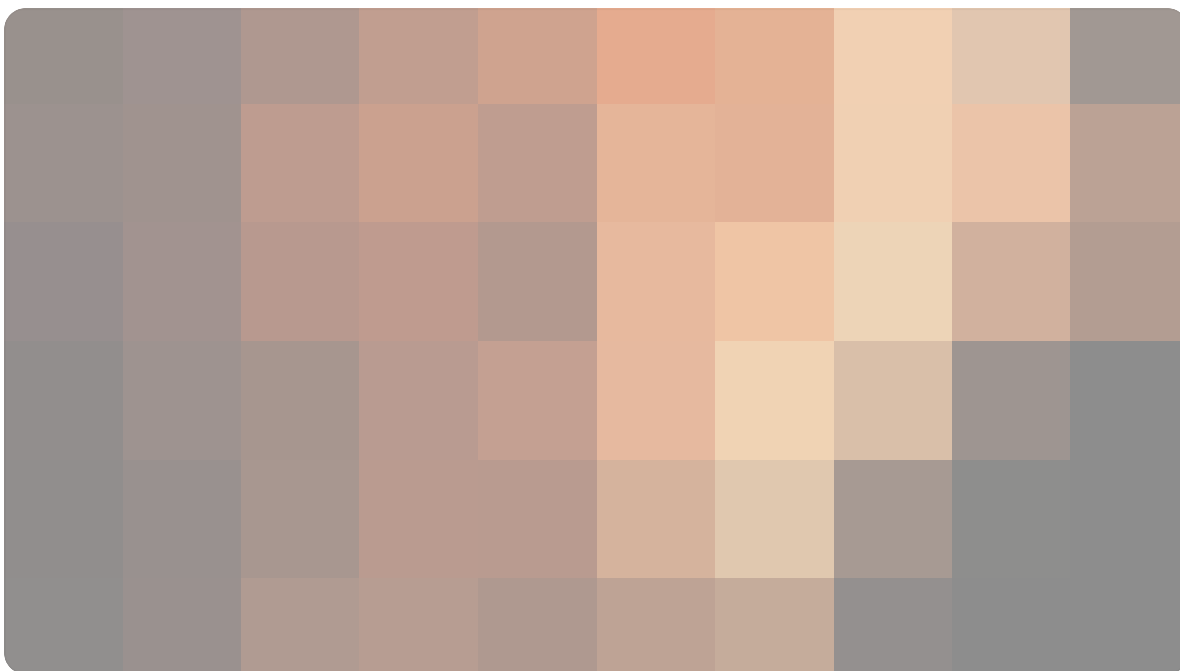
**Игорь Козакевич**

14 авг

Подписаться

Как я участвовал в обмене активами и какой брокер самый адекватный

Инвестиционная палата отчиталась о завершении процесса обмена активами. Отчиталась посредственно, ведь из более 1 миллиона заявок на 35,3 млрд. рублей удалось поменять активы только на 8,1 млрд. рублей. Вины брокера в этом нет, так как я убежден, что он сделал всё возможное и невозможное, чтобы довести этот уровень до максимального 23% успеха.



3



797

