



empenoso

25 мар в 03:20

## Поиск ликвидных облигаций с использованием Python

Простой

5 мин

13K

Python\*, API\*, Open source\*, Финансы в IT

Кейс

Сезон Open source

На Московской бирже торгуется более 2500 облигаций, но большая часть из них неликвидна - в стакане почти нет предложений и сделок совершается крайне мало. Это затрудняет покупку и продажу таких бумаг. При этом известные мне публичные сервисы не суммируют объемы торгов за период, поэтому сложно быстро найти облигации с высокой ликвидностью.

Пять лет назад написал Node.js-скрипт, затем адаптировал его для Google Таблиц, а теперь разрабатываю Python версию. При помощи сообщества на GitHub эта Python версия идёт к созданию полноценной библиотеки с расширенными возможностями: автоматический поиск ликвидных облигаций, расчет денежных потоков, сбор новостей по эмитентам и вычисление оптимального объема покупки. Все это направлено на помощь простым инвесторам, вроде нас с вами, чтобы оперативно находить выгодные инвестиционные инструменты и принимать решения на основе актуальной информации.

РЕКЛАМА

**GPU от 29 ₽/час**

Все ресурсы GPU только ваши!

```

18 class MOEX:
216 def search_volume(self, security_id: str, threshold_value: int) -> dict[str, int]:
220     # этот день и день вчераш
227     board_id = self.board_id(security_id)
228     if not board_id:
229         self.log.info(
230             f"⚠ Не удалось получить board_id для {security_id}. Поиск объема прерван."
231         )
232         return {"low_liquid": 1, "value": 0}
233
234     url = (
235         f"https://iss.moex.com/iss/history/engines/stock/markets/bonds/boards/{board_id}/securities/{sec
236         f"iss.meta=off&iss.only=history&history.columns=SECID,TRADEDATE,VOLUME,NUMTRADES&limit=20&from={
237     )
238     # numtrades - Минимальное количество сделок с бумагой
239     # VOLUME - оборот в количестве бумаг (Объем сделок, шт)
240     self.log.info(
241         f"🔗 {foo_name}. Ссылка для поиска объема сделок {security_id}: {url}"
242     )
243     try:
244         time.sleep(self.API_DELAY)
245
246         response = requests.get(url)
247         response.raise_for_status()
248         json_data = response.json()
249         history_data = json_data["history"]["data"]
250
251         count = len(history_data)
252         volume_sum = 0
253         low_liquid = 0
254         for i in range(count):
255             volume = history_data[i][2]
256             volume_sum += volume
257             if threshold_value > volume: # если оборот в конкретный день меньше
258                 low_liquid = 1

```

## Критерии выбора ликвидных облигаций на Московской Бирже

Ликвидность это один из ключевых параметров, поскольку даже высокодоходная бумага бесполезна, если её невозможно купить. В моём скрипте для поиска облигаций используются несколько основных критериев:

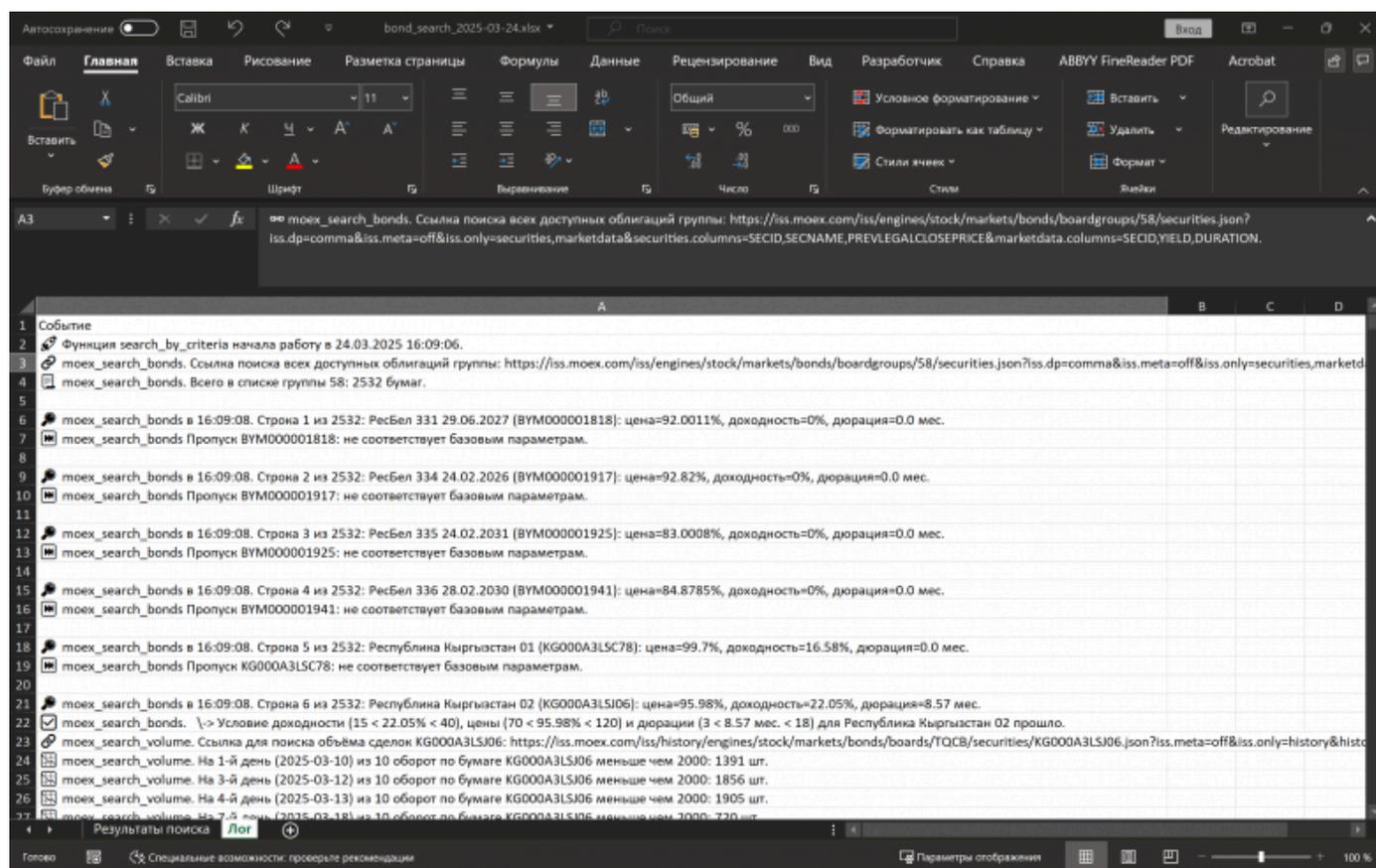
### Доходность

Эффективная доходность облигации — один из главных параметров. В фильтре задаётся диапазон, например, от 15% до 30%. Важно учитывать, что этот показатель не включает налог с купонов и комиссии брокера.



**GPU от 29 ₽/час**

Все ресурсы GPU только ваши!



## Текущая цена

Облигации торгуются по разным ценам относительно номинала, все цены облигаций указаны в процентах, и этот параметр позволяет фактически выбрать стратегию получения дохода:

- Если облигация торгуется сильно ниже номинала, основная доходность будет сформирована к моменту погашения (за счёт разницы между ценой покупки и номиналом).
- Если облигация торгуется близко к номиналу, основная доходность складывается из купонных выплат в течение срока жизни.



**GPU от 29 ₽/час**

Все ресурсы GPU только ваши!

**Пример:** облигация ЕвроТранс БО-001Р-03 на 10 марта 2025 года (код RU000A1061K1, [ссылка](#)):

- Текущая цена: 86% от номинала
- Купонная доходность: 13,6% годовых
- Доходность к погашению: 26,78% годовых. Доходность к погашению предполагает, что вы держите облигацию до погашения и что все купонные выплаты будут произведены в срок.

Откуда такая разница? Дело в том, что облигация сейчас торгуется ниже номинала, а при погашении инвестор получит 100% номинальной стоимости. То есть, кроме купонов, инвестор дополнительно зарабатывает на разнице в цене. Именно поэтому параметр текущей цены помогает выбрать, когда получать основную доходность — постепенно в течение срока или разово в момент погашения.

## Дюрация

Показатель дюрации позволяет выбрать облигации с нужным сроком жизни. Например, если мне нужна бумага на ближайшие 3–18 месяцев, фильтр исключает слишком краткосрочные или долгосрочные варианты.



**GPU от 29 ₽/час**

Все ресурсы GPU только ваши!

Наличие полной информации о будущих купонных выплатах или наличие оферты.

Также я исключаю флоатеры, поскольку Московская биржа не передаёт по ним данные о будущих платежах.

## Ликвидность

Ликвидность - основной критерий, ради которого создавался этот инструмент. В скрипте анализируются:

- Минимальное число сделок за каждый из последних 15 дней - чтобы исключить облигации, которые могут внезапно «замереть».
- Общий объём сделок за 15 дней - параметр, который позволяет выявлять бумаги с устойчивым спросом. Этот скрипт позволяет гибко подстраивать фильтры и находить действительно ликвидные облигации, подходящие под конкретно Вашу стратегию инвестирования.

## Как работает скрипт

Скрипт использует API Московской биржи для получения актуальных данных об облигациях. Данные скачиваются для ознакомительных целей и это позволяет оперативно находить ликвидные облигации.



**GPU от 29 ₽/час**

Все ресурсы GPU только ваши!

**Ограничение запросов.** С сентября 2024 года API Московской биржи начало периодически разрывать соединение без объяснения причин. Это продолжалось около полугода, поэтому в коде был установлен лимит — не более 50 запросов в минуту. Сейчас, в марте 2025, эта проблема больше не наблюдается, но ограничение оставлено для стабильности.

Московская биржа периодически меняет формат данных, что требует оперативного обновления скрипта.

## Пошаговое руководство по запуску

Если вы не разбираетесь в программировании, но хотите воспользоваться этим Python-скриптом, следуйте инструкции:

### Шаг 1. Скачайте скрипт

1. Откройте ссылку: [GitHub проекта](#).
2. Нажмите "Code" → "Download ZIP".
3. Разархивируйте ZIP в удобную папку.



**GPU от 29 ₽/час**

Все ресурсы GPU только ваши!

Если Python не установлен:

- Windows: скачайте и установите [Python с официального сайта](#). В установке отметьте "Add Python to PATH".
- MacOS: скачайте [Python для macOS](#) и установите.

### Шаг 3. Установите зависимости проекта

1. Откройте папку с проектом.
2. Дважды кликните файл `install_requirements.bat` (Windows) или `install_requirements.command` (MacOS).

### Шаг 4. Запустите скрипт

Дважды кликните файл `1_bonds_search_by_criteria.py`.

Во время выполнения отображается лог выполнения.



**GPU от 29 ₽/час**

Все ресурсы GPU только ваши!

Будет создан файл с текущей датой: `bond_search_2025-03-25.xlsx`

## Преимущества open source и шаги к Python-библиотеке

Идея - набор из четырёх скриптов для личного использования. Разработал их как частный инвестор, понимая какие задачи стоят передо мной:

1. Поиск ликвидных облигаций
2. Автоматический расчёт денежных потоков
3. Сбор новостей по эмитентам
4. Расчёт оптимального объёма покупки

### От одиночного скрипта к полноценной библиотеке

Так как проект открыт, к нему подключилось сообщество. Одним из первых с pull request пришёл Imasikl, который заинтересовался темой облигаций и предложил преобразовать набор скриптов в полноценную Python-библиотеку. Уже было внесено множество улучшений:

М. 0005



**GPU от 29 ₽/час**

Все ресурсы GPU только ваши!

- #11 – Добавлена диаграмма взаимодействия пользователя со скриптом (автор: lmasikl, одобрено 17 марта)
- #7 – Отформатирован код с дефолтными настройками Ruff (автор: lmasikl, одобрено 14 марта)
- #6 – Внесен в приложение скрипт расчета оптимального объема покупки облигаций (автор: lmasikl, одобрено 13 марта)
- #5 – Перенесен в приложение скрипт поиска новостей (автор: lmasikl, одобрено 12 марта)
- #4 – Перенесен в приложение скрипт поиска облигаций (автор: lmasikl, одобрено 10 марта)
- #3 – Начат переход к полноценной библиотеке (автор: lmasikl, одобрено 6 марта)

## Февраль 2025

- #2 – Исправлены некорректные значения в рублях (автор: gogvajbobo, одобрено 21 февраля)
- #1 – Выполнен рефакторинг кода (автор: lmasikl, одобрено 20 февраля)

Добавлен план схемы работы:



**GPU от 29 ₽/час**

Все ресурсы GPU только ваши!



**GPU от 29 ₽/час**

Все ресурсы GPU только ваши!

## Почему open source — это важно?

Открытый код даёт возможность сообществу вносить улучшения, исправлять ошибки и расширять функциональность. Гибкость библиотеки позволяет каждому настроить поиск облигаций под собственные нужды, создавая индивидуальные стратегии отбора.

## Как практически использовать эту библиотеку

Допустим, у нас есть 300 000 рублей, которые мы хотим вложить в облигации. Чтобы минимизировать риски, разделим сумму на 10 разных облигаций.



**GPU от 29 ₽/час**

Все ресурсы GPU только ваши!

Первая часть скрипта анализирует рынок и отбирает бумаги с хорошей ликвидностью,

подходящие под заданные критерии (доходность, дюрация, цена и т. д.).

## 2. Проверка эмитента

Запускаем вторую часть скрипта — он собирает последние новости по эмитентам. Если обнаружены негативные публикации (например, судебные иски или финансовые проблемы компании), такие облигации исключаем из списка.

## 3. Расчёт денежных потоков

Далее, используя третью часть скрипта, можно заранее рассчитать будущие выплаты по купонам и спрогнозировать точную доходность портфеля.

## 4. Расчёт объёма покупки

Последний скрипт поможет рассчитать, сколько именно облигаций можно приобрести с учётом доступного капитала и НКД. Это позволяет эффективно распределить средства и избежать недостатка ликвидности.

Раз в месяц достаточно просматривать портфель, анализировать новые облигации через первый скрипт и при необходимости докупать бумаги. Такой алгоритм можно повторять бесконечно, постепенно увеличивая капитал.

## Заключение

Использование этого скрипта позволяет частному инвестору систематизировать процесс подбора облигаций, минимизировать риски и упростить управление портфелем.



**GPU от 29 ₽/час**

Все ресурсы GPU только ваши!

Этот проект развивается благодаря усилиям энтузиастов и разработчиков, заинтересованных в автоматизации инвестирования. Если у вас есть идеи по улучшению функциональности или вы хотите протестировать новые возможности, присоединяйтесь к обсуждению на GitHub!

Любые предложения, правки и новые модули помогут сделать библиотеку ещё более мощным инструментом для инвесторов.

**Автор:** Михаил Шардин

 [Моя онлайн-визитка](#)

 [Telegram «Умный Дом Инвестора»](#)

25 марта 2025 г.

**Теги:** [облигации](#), [московская биржа](#), [moex](#), [Сезон Open source](#), [алгоритмическая торговля](#)

**Хабы:** [Python](#), [API](#), [Open source](#), [Финансы в IT](#)

## Редакторский дайджест ×

Присылаем лучшие статьи раз в месяц

**191**

Карма

**11.1**

Рейтинг

**Михаил Шардин** [@empenoso](#)

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

[Подписаться](#)

[Хабр Карьера](#) [Сайт](#) [Сайт](#) [Github](#)

**GPU от 29 ₽/час**

Все ресурсы GPU только ваши!

Комментарии 43

## Публикации

ЛУЧШИЕ ЗА СУТКИ

ПОХОЖИЕ



eleday

21 час назад

### Рукописный редактор на Python: инструкция для тех, кто хочет «рисовать» код



Средний



10 мин



4.9K

Тutorial

+38

33

7



GPU от 29 ₽/час

Все ресурсы GPU только ваши!



alizar

20 часов назад

## Читерство в законе. Как пройти собеседование и устроиться на работу в дивном новом мире



Простой



6 мин



6.5K

Мнение



+33



75



31



lynikol

16 часов назад

## На входе аудио, на выходе — саммари. Собираем локальный транскрибатор из бесплатного софта



Средний



18 мин



2K

Тutorial



+32



58



30



Dingzhibo

17 часов назад

## Как не переплатить за автоматизацию? Разбираем, когда стоит подключать ML



11 мин



1.4K



+25



13



0



PatientZero

13 часов назад

## Я не люблю NumPy



Средний



10 мин



4.1K

Обзор

Перевод



+24



41



4



GPU от 29 ₽/час

Все ресурсы GPU только ваши!

**vegeed**

20 часов назад

## SQL-линтер на Rust, HTML-минификатор для .NET и эмулятор DOS в браузере: кто получил гранты Yandex Open Source

👍 Простой 🕒 7 мин 👁 1.8K📌 +24📖 9💬 2**AlexeySu**

18 часов назад

## Ошибка компилятора или неожиданный эффект шаблонов в C++?

👍 Сложный 🕒 3 мин 👁 2.4KИз песочницы📌 +22📖 19💬 5**ru\_vds**

16 часов назад

## Веб-разработка на ванильном HTML, CSS и JavaScript: стилизация и сайты

👍 Средний 🕒 22 мин 👁 1.1KТutorialПеревод📌 +21📖 37💬 2**k0mar0v**

22 часа назад

## Ericsson SH888: культовый телефон конца 90-х. Что внутри?

🕒 7 мин 👁 2.6K📌 +21📖 6💬 7**b\_alchinov****GPU от 29 ₽/час**

Все ресурсы GPU только ваши!

маршрутизатора для облачных сервисов

Средний 5 мин 4.3K

Обзор

+21

3

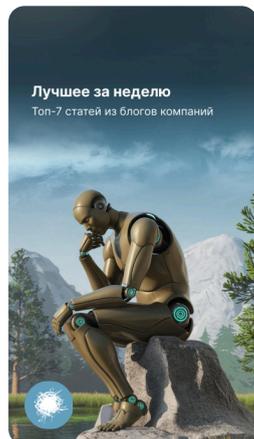
4

## Кто вы, мистер багхантер? Что показал опрос на Хабре

Турбо

Показать еще

### ИСТОРИИ



Топ-7 годных статей из блогов компаний



Старт гонки разработчиков



Торопись в сезон Open source



Открыт приём в Школу анализа данных



С Днём радио!

### ВАКАНСИИ

#### Ведущий разработчик С (Open source)

от 300 000 ₽ · СберТех · Москва · Можно удаленно

#### Python разработчик Senior/Lead (Django, DRF)

от 250 000 до 350 000 ₽ · Hello, Doc! · Можно удаленно



GPU от 29 ₽/час

Все ресурсы GPU только ваши!

Python разработчик

от 1 500 до 3 000 \$ · DevTeam.Space · Москва · Можно удаленно

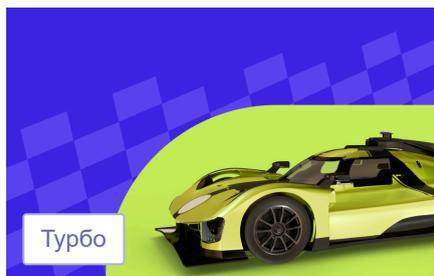
## Python разработчик Senior

от 200 000 до 300 000 Р · Туроператор «Русь» · Москва  
 Больше вакансий на [Хабр Карьере](#)

### МИНУТОЧКУ ВНИМАНИЯ



Как избежать ошибок при бизнес-миграции и сохранить данные?



Гонка разработчиков backend, frontend, mobile уже началась! Займи место на старте



Экономим деньги со скидками в Промокодусе

### РАБОТА

[Data Scientist](#)

51 вакансия

[Python разработчик](#)

67 вакансий

[Django разработчик](#)

20 вакансий

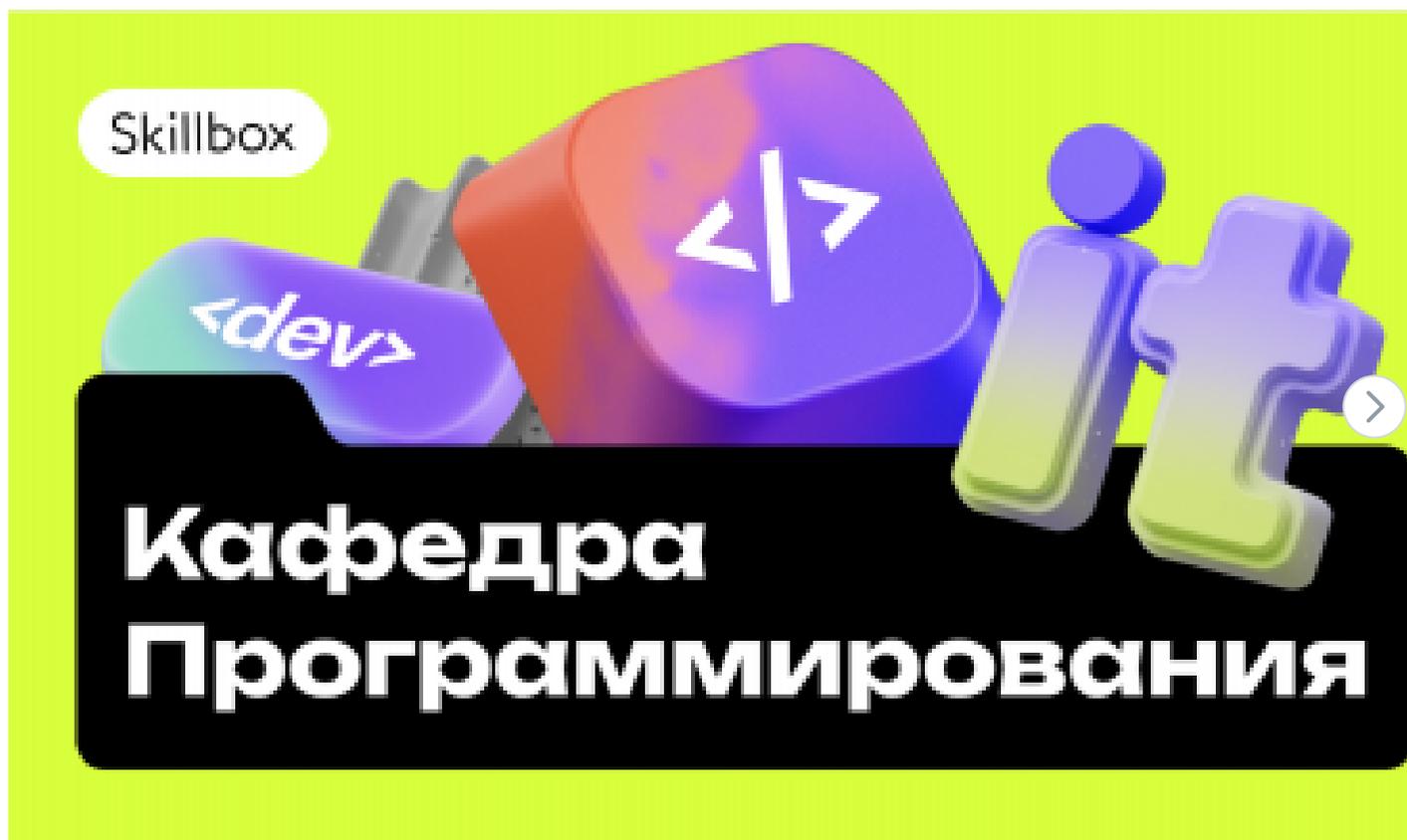
[Все вакансии](#)

### БЛИЖАЙШИЕ СОБЫТИЯ



**GPU от 29 Р/час**

Все ресурсы GPU только ваши!



17 апреля – 29 мая

## Серия бесплатных офлайн-конференций «Кафедра Программирования от Skillbox»

Москва

Разработка

Больше событий в календаре

Хабр



GPU от 29 ₽/час

Все ресурсы GPU только ваши!

техническая поддержка

© 2006–2025, Habr



**GPU от 29 ₽/час**

Все ресурсы GPU только ваши!