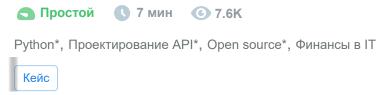




Автоматизируем учёт облигаций: как избавиться от рутины



В преддверии одной большой конференции про облигации я, Михаил Шардин и Екатерина Кутняк решили написать статью о том, почему облигации важны и как можно автоматизировать рутинные действия с ними.



Передаю слово Екатерине:

Сейчас, наверно, уже самые начинающие инвесторы понимают насколько важны облигации в портфеле. Ведь они выполняют целый ряд функций, помимо зарабатывания денег.

Улучшают ликвидность, оптимизирую показатель риск/доходность, придают гибкость при принятии решений. А на сегодняшний день они ещё и самый доходный и предсказуемый инструмент.

Мало кто из частных инвесторов заморачивается ведением своего портфеля. Конечно, это неправильно, но вполне можно понять. Слишком много времени уходит на учет, а его всегда и всем не хватает. Поэтому и доверяют своему торговому приложению. Но не видя картинки в целом, очень трудно принимать как стратегические, так и тактические решения по портфелю.

Сейчас есть приложения, которые помогают вести портфель. Они не бесплатные, но порядок в портфеле того стоит.

Это статья скорее для тех, кто понимает важность учета и пытается сам вести свой преклама дь в Excel. У меня было четыре статьи о том, как вести учет облигаций в Excel. Их



Получи грант за код

Как вести учет облигаций в Excel smart-lab.ru
Как вести учет облигаций в Excel. Расчет доходности smart-lab.ru
Как вести учет облигаций в Excel. Часть 3. Дюрация smart-lab.ru
Как рассчитать дюрацию «на коленке». Часть 4 smart-lab.ru

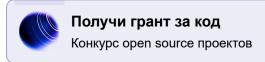
Но всегда есть способ что-то улучшить и оптимизировать. Например сервисы по учету бумаг, такие как Snowball или другие, предлагают помимо прочих опций еще и график купонных выплат. Вещь, безусловно очень важная. Почему?

Чтобы максимально эффективно управлять облигационной частью портфеля, важно учитывать и анализировать все выплаты.

Почему учёт купонов так важен?

- ✓ Позволяет прогнозировать денежные потоки.
- ✓ Упрощает реинвестирование.
- Помогает оценить реальную доходность портфеля.

Например, брокерские приложения принципиально не учитывают выплаченные купоны, что вносит дикую путаницу в умах инвесторов. Они бесконечно жалуются на "красные" портфели, и сетуют на убытки. Забывая о том, что облигации - это еще и полученные купоны.



того, о чем мы расскажем в этой статье.

Итак, у нас есть проблема:

Как учитывать купонные выплаты, если в портфеле несколько облигаций с разными датами выплат? Вручную отслеживать их сложно, особенно если бумаги погашаются и покупаются новые.

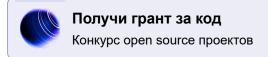
Обычные способы учёта:

- Вести записи вручную в Excel.
- Использовать специализированные приложения (платно)

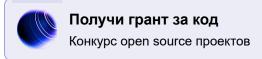
Ручной способ

В Excel трудоёмко, легко что-то упустить. От ручного вбивания каждой выплаты, а перешла к сводным таблицам. Это гораздо нагляднее. В результате, получается отлично отслеживать не только купоны, но и погашения, и оферты.

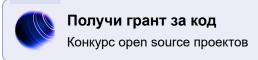
Можно смотреть выплаты по месяцам



А можно раскрыть таблицу, и смотреть по дням в конкретном месяце:



Можно добавить наименование выпусков, и тогда картинка получается максимально информативная:



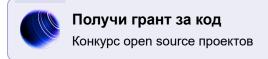
Безусловно, так гораздо нагляднее и удобнее. Но остается одна проблема, чтобы сделать сводные таблицы все еще нужна исходная таблица, содержащая всю необходимую информацию. Вот ее то по-прежнему приходилось делать руками.

Но и у этой проблемы есть техническое решение.

Передаём слово Михаилу:

Автоматизированное получение данных о купонах и выплатах номинала облигаций

Разработал Python скрипт, который автоматически скачивает данные о купонах и выплатах номинала через API Московской биржи для списка облигаций из первой вкладки Excelфайла и записывает результат обратно в этот же файл на вкладку "Ден.поток". Третья вкладка содержит формулы анализа полученных данных.



Расчёт для портфеля на 5 млн рублей

Я выбрал использовать Python скрипт отдельно от встроенный функции Python в Excel, потому что встроенная функция накладывает множество ограничений. Например, у меня был Microsoft Office 2021 - в нём функции Python нет. Пришлось установить Microsoft Office 2024 - только в нём есть встроенный Python в Excel. Но и этого оказалось недостаточно чтобы запускать встроенный Python в Excel. Необходимо иметь Office 365 (Mondo 2016) подписку, чтобы функция активировалась. И будет работать только под Windows. Да и сам



Получи грант за код

Исходя из этого я выбрал написать отдельный Python скрипт - ведь тогда уже не важно какая именно версия Эксель у вас установлена. И даже не важно какая операционная система - Windows или macOS. Да даже если и Linux - Python поддерживает множество операционных систем, а свободный аналог Экселя под Linux называется LibreOffice Calc.

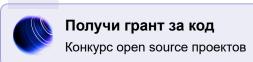
Как это помогает?

- Видно, когда и сколько денег поступит.
- Можно заранее планировать реинвестирование.
- Повышается контроль над кэшем в портфеле.

Все это касается уже существующих портфелей.

Но есть еще один неочевидный плюс для тех, кто прежде чем купить облигационный портфель, планирует его структуру. Можно не прилагая больших усилий найти оптимальное соотношение бумаг, чтобы получить планируемый поток кэша. Это очень удобно именно на этапе выбора структуры.

Как это выглядит на практике?



через отправку pull request (PR) на GitHub.

Конкурс open source проектов

Для того чтобы Вы могли воспользоваться скриптом - сначала скачайте его на свой компьютер с Гитхаба по кнопке < > Code / Download ZIP.

Весь скрипт находится в файле « 2_bonds_search coupons.py ». Вы можете познакомится с его содержимым перед запуском - для этого откройте его, например, блокнотом.

```
# в Скачивание и обработка данных о денежном потоке облигаций в направнения в этот Рути потоке облигаций в направнения на направнения в этот Рути потоке облигаций из Excel-файла bonds.xlsx и направнения в этот же файл. направнения в этот же файл. на направнения в этот же файл.
```

```
import requests
import openpyxl
from datetime import datetime
# Настройка кодировки для корректного вывода русского текста
import sys
sys.stdout.reconfigure(encoding='utf-8')
# Загружаем Excel-файл
file path = "bonds.xlsx"
wb = openpyxl.load_workbook(file_path)
sheet_data = wb["Исходные данные"]
sheet_result = wb["Ден.ποτοκ"]
# Очищаем лист с результатами
sheet_result.delete_rows(1, sheet_result.max_row)
sheet_result.append(["Название", "Идентификатор", "Дата выплаты", "Денежный поток, ₽ (к
# Считываем данные из листа "Исходные данные"
ArraySymbolQuantity = []
for row in sheet data.iter rows(min row=2, max row=sheet data.max row, values_only=True
    if row[0] and row[1]: # Проверяем, что данные не пустые
        ArraySymbolQuantity.append(row)
print(f"Считано {len(ArraySymbolQuantity)} облигаций для обработки.")
CashFlow = []
# Обрабатываем каждую облигацию
for ID, number in ArraySymbolQuantity:
    print(f"\nOбрабатываем {ID}, количество: {number} шт.")
    url = f"https://iss.moex.com/iss/statistics/engines/stock/markets/bonds/bondization
    print(f"3aπpoc κ {url}")
    response = requests.get(url)
    json_data = response.json()
    # Обработка купонов
    for coupon in json_data.get("coupons", {}).get("data", []):
        name = coupon[1].replace('"', '').replace("'", '').replace("\\", '')
      Получи грант за код
      Конкурс open source проектов
```

```
# Преобразуем дату в объект datetime
        coupon_datetime = datetime.strptime(coupon_date, "%Y-%m-%d")
        if coupon_datetime > datetime.now():
            value_rub = (coupon[9] or 0) * number
            CashFlow.append([f"{name} (купон 🌮)", isin, coupon_datetime, value_rub])
            print(f"Добавлен купон: {CashFlow[-1]}")
    # Обработка выплат номинала
    for amort in json_data.get("amortizations", {}).get("data", []):
        name = amort[1].replace('"', '').replace("'", '').replace("\\", '')
        isin = amort[0]
        amort_date = amort[3]
        # Преобразуем дату в объект datetime
        amort_datetime = datetime.strptime(amort_date, "%Y-%m-%d")
        if amort_datetime > datetime.now():
            value_rub = (amort[9] or 0) * number
            CashFlow.append([f"{name} (номинал №)", isin, amort_datetime, value_rub])
            print(f"Добавлена выплата номинала: {CashFlow[-1]}")
# Записываем данные в Excel
for row in CashFlow:
    sheet_result.append(row)
# Устанавливаем формат ячеек
for cell in sheet_result["C"][1:]: # Пропускаем заголовок
    cell.number_format = "DD.MM.YYYY"
for cell in sheet_result["D"][1:]:
    cell.number_format = '# ##0,00 ₽'
# Добавляем запись об обновлении
update_message = f"\nДанные автоматически обновлены {datetime.now().strftime('%d.%m.%Y
sheet_result.append(["", update_message])
print(update_message)
# Сохраняем изменения в файле
wb.save(file_path)
```

Получи грант за код

```
# В конце скрипта
input("Нажмите любую клавишу для выхода...")
```

Хотя нумерация скрипта начинается с двойки, это полностью самостоятельный скрипт, который ищет и скачивает данные о купонах и выплатах номинала через API Московской биржи.

Теперь чтобы запустить скрипт сначала нужно установить Python. Скачайте и установите последнюю версию с официального сайта Python для Вашей операционной системы.

Затем, найдите в меню "Пуск" "Командную строку" или "Терминал" и откройте её. В командной строке введите:

```
pip install requests openpyxl
```

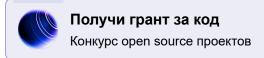
Это установит необходимые для работы скрипта инструменты - без их установки скрипт работать не будет:

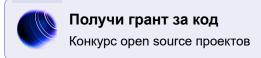
• Requests: библиотека для отправки HTTP-запросов и работы с ответами от серверов.



Получи грант за код

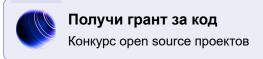
Заполните файл-шаблон « bonds.xlsx » Вашими собственными данными. Переименовывать файл и менять название вкладок нельзя - в скрипте жестко заданы названия файла и вкладки. Убедитесь, что Excel-файл « bonds.xlsx » закрыт перед запуском.





Теперь, найдите « 2_bonds_search coupons.py » в проводнике, дважды кликните по нему. Это запустит скрипт, и он обновит файл Excel. После окончания работы можно посмотреть логи работы - скрипт ждёт нажатия кнопки.

Все эти данные, полученные через API Мосбиржи доступны только вам и для ознакомления.



Скрипт автоматически заполняет вкладку с денежным потоком, а все сводные данные в человеческом понятном виде находится на вкладке аналитика. Надеюсь этот скрипт облегчит Вашу работу!

Итоги: почему стоит учитывать облигации и как это сделать эффективно?

- Без учёта облигаций инвестор упускает важную часть доходности.
- Системный учёт купонных выплат даёт чёткое понимание будущих денежных потоков.
- Автоматизация позволяет не тратить время на рутину.



Получи грант за код

Авторы статьи: Михаил Шардин и Екатерина Кутняк

Код и техническая реализация: Михаил Шардин

P.S. Я планирую написать и опубликовать ещё три продолжения.

P.P.S. Небольшой анонс для публикации следующей недели: для меня, как для частного инвестора многие варианты облигаций с большими процентами окажутся мусорными. И у меня, как частного инвестора, много других проблем и забот - и мне не всегда есть время разбираться чем занят тот или иной эмитент. Нет времени знакомиться с эмитентом или проводить какую-то аналитику.

Публикация следующей недели будет посвящена скрипту, который по коду ценной бумаги с Московской биржи внутри файла bonds.xlsx автоматически собирает все последние новости об этой компании.

Теги: moex, облигации, сезон open source

Хабы: Python, Проектирование API, Open source, Финансы в IT



Присылаем лучшие статьи раз в месяц

Электропочта

X

Оставляя свою почту, я принимаю Политику конфиденциальности и даю согласие на получение рассылок



212

63.3

Карма Рейтинг

Михаил Шардин @empenoso



Получи грант за код

Сайт Сайт GitHub

Комментарии 3

Публикации

ЛУЧШИЕ ЗА СУТКИ ПОХОЖИЕ

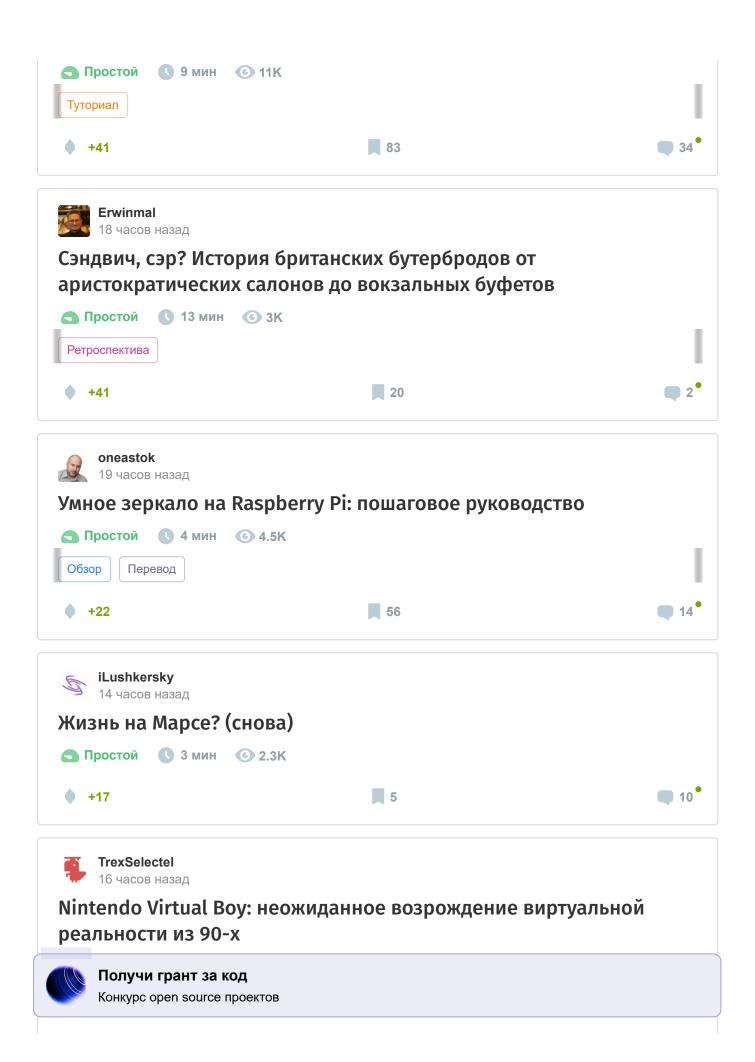


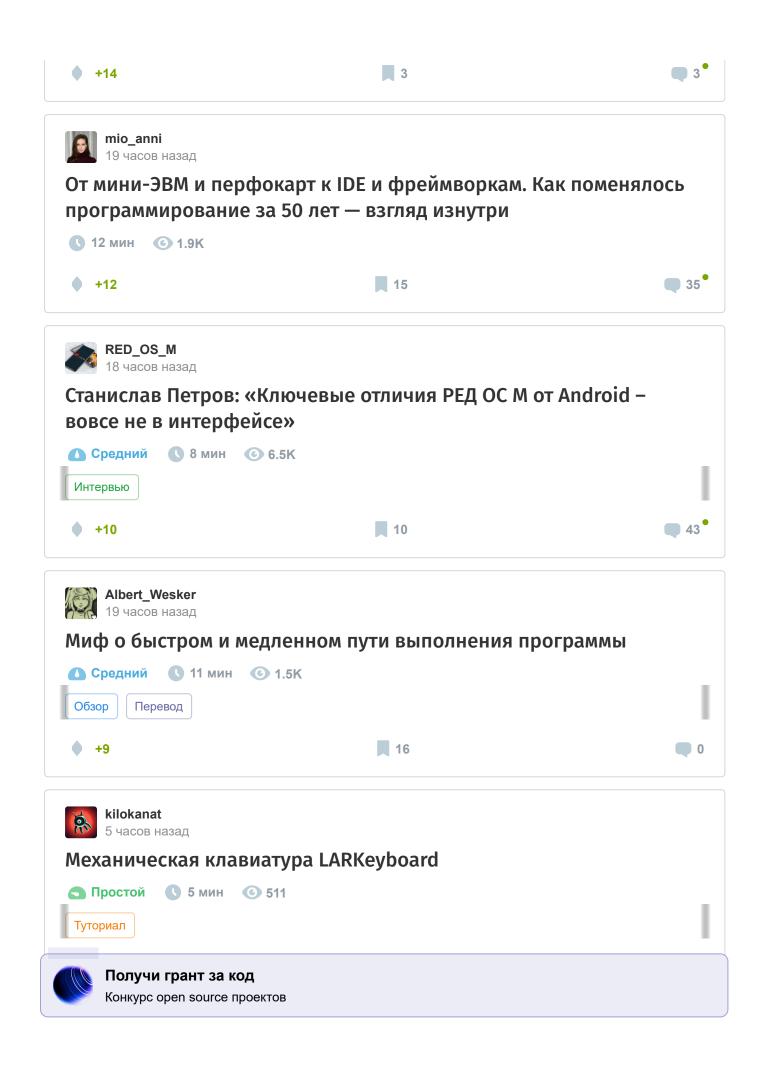
Tirarex

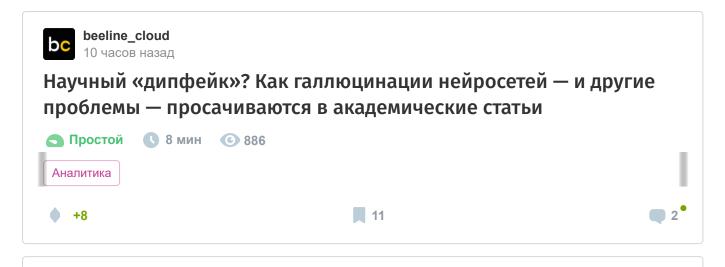
14 часов назад



Получи грант за код







Цифровые двойники и Лаборатория ИИ— мир ИТ-прома, каким вы его не знали

Турбо

Показать еще

ИСТОРИИ



Чай, тортик и код: с Днём программиста!



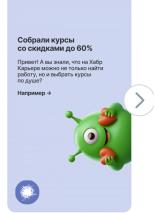
Made in Al



Чего хотят лиды в бигтехе?



Как расти в ИТ: советы, гайды и опыт сеньоров



Курсы со скидками до 60%

КУРСЫ



Получи грант за код

Профессия Веб-разработчик

По мере набора группы

🐧 Профессия: Фронтенд-разработчик

По мере набора группы

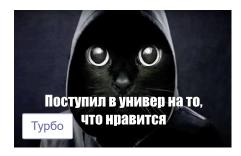
🔀 Инженер по тестированию

По мере набора группы

💲 Аналитик данных

По мере набора группы Больше курсов на Хабр Карьере

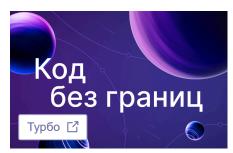
минуточку внимания



Где обучают вести кибертурниры и разрабатывать игры

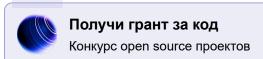


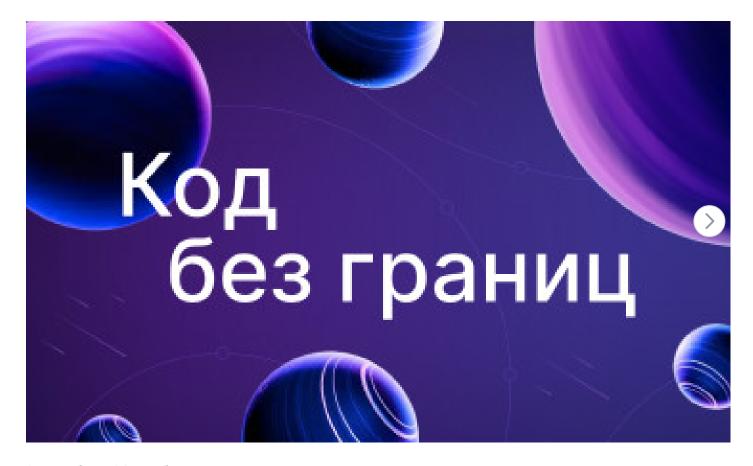
Прямая трансляция IT Elements: ключевые доклады



Отправь свой open source проект на конкурс и выиграй грант

БЛИЖАЙШИЕ СОБЫТИЯ





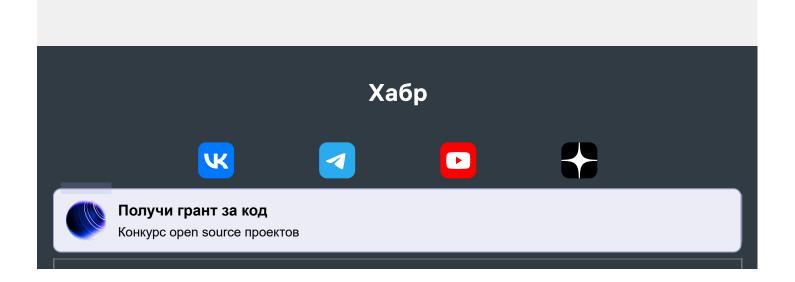
3 сентября – 31 октября

Программа грантов для развития open source проектов «Код без грани

Онлайн

Разработка

Больше событий в календаре



Техническая поддержка

© 2006–2025, Habr

