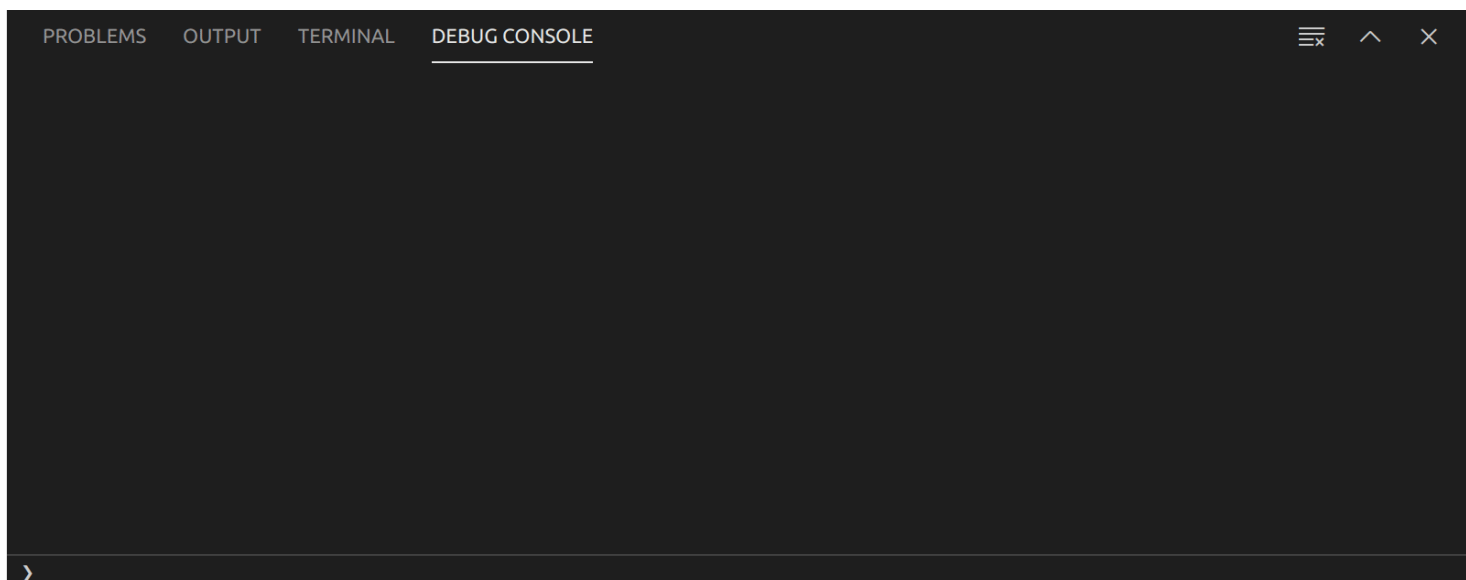


**empenoso** 22 июня 2020 в 07:26

Скрипт выборки российских облигаций по параметрам

JavaScript *, Node.js *, API *, Визуализация данных , Финансы в IT

Уже несколько лет я пользуюсь облигациями в качестве замены депозита, потому что процент дохода, который можно получить со вклада **стабильно падает**. В отличие от ситуации с депозитом, в облигациях всегда **можно найти** большую доходность. И в этой ситуации меня не устраивало только количество времени на механическую работу по поиску подходящих вариантов бумаг.

*Работа скрипта по поиску облигаций на Московской бирже*

Так как сервисов по поиску российских облигаций много, но ни один из них не имеет достаточной гибкости и простоты и поэтому на работу с ними тратится достаточно много времени. Исходя из этого и решил разработать собственный скрипт для поиска облигаций.

Сделал это на Node.js с выводом полученных результатов в локальный html файл с интерактивной таблицей от [Google Charts](#) (а в случае, если JavaScript отключен в браузере, что например происходит при открытии этого html файла из мессенджера на iPhone, то отображается статическая версия таблицы, также сгенерированная скриптом).

Существующие сервисы и мои параметры для поиска

Существующих сервисов довольно много:



+45



209



114 +114



- [sMart-lab.ru](https://smart-lab.ru) — блоги трейдеров и инвесторов.
- [Cbonds](https://cbonds.ru). Рынок внутренних и международных облигаций.
- Облигации в России — [Rusbonds](https://rusbonds.ru).
- [Bonds.Finam.RU](https://bonds.finam.ru) — информация о российском рынке облигаций.
- Сервис поиска от [Московской Биржи](https://moscow-exchange.ru).
- [Анализ облигаций. Beta](https://beta.ukdohod.ru). УК ДОХОДЪ.

В разное время я пользовался всеми из них, причем некоторые из них были платные. Что мне не нравилось в этих сервисах — так это обилие параметров, в которых легко погрязнуть и которые не ведут к желаемому результату.

Мой желаемый результат — актуальная выборка из всех российских облигаций по следующим параметрам:

1. Заданный диапазон текущей доходности.
2. Заданный диапазон текущих цен.
3. Заданный диапазон дюрации.
4. Объем сделок за последние n дней больше порогового.
5. Ответ на вопрос — есть ли налоговая льгота для корпоративных облигаций, выпущенных после 1 января 2017 года?

Конкретные цифры диапазонов могут быть любыми, например:

- $5\% < \text{Доходность} < 11\%$
- $98\% < \text{Цена} < 101\%$
- $4 \text{ мес.} < \text{Дюрация} < 15 \text{ мес.}$
- Объем сделок за n дней $> 15\,000$ шт.

А на выходе я бы хотел получать не больше 2-х десятков вариантов, которые точно попадали бы под мои критерии. Если вариантов находится больше, то лучше ужесточить свои критерии для получения меньшей по размеру выборки, которая бы точно соответствовала моим ожиданиям.

Облигации на Московской бирже доступны внутри основных режимов торгов:

- T0: Основной режим — безадрес. (до 22.05.2020: 1443 бумаг, в июне — 131 шт.).

- T+: Основной режим — безадрес. (до 22.05.2020: 295 бумаг, в июне — 1638 шт.).
- T+: Основной режим (USD) — безадрес. (до 22.05.2020: 125 бумаг, в июне — 128 шт.).

Облигаций много, именно потому автоматизированное решение поиска мне кажется правильным шагом.

Мой скрипт поиска облигаций на Московской бирже

Я понимаю, что человек, которому необходим поиск облигации может и не разбираться в программировании, а тому, кто легко разберется в коде этого скрипта облигации могут быть неинтересны. И разбирающихся в программировании на Хабре явно больше, чем тех, кто разбирается в облигациях.

Я хотел найти некий баланс — чтобы минимально подкованный человек мог воспользоваться результатами работы этого скрипта.

Ещё одно очень важное отступление — в скрипте всё напрямую зависит от работы [API Московской биржи](#), которое имеет свои особенности.

Если говорить про поиск облигаций, то сразу после открытия торгов значения доходности по средневзвешенной цене (`YIELDATWAPRICE`) обнуляются.



Схема определения средневзвешенной цены (`WAPRICE`)

Значение `YIELDATWAPRICE` на мой взгляд выглядело лучше для целей поиска, но пришлось использовать `YIELD`, иначе сразу после открытия биржи работа скрипта была невозможна.

Ещё я использую цену предыдущего закрытия (`PREVLEGALCLOSEPRICE`), из-за того что по некоторым облигациям торгов может не быть несколько дней.



Схема определения цены закрытия (`LEGALCLOSEPRICE`)

Чтобы уменьшить количество обращений к [API Московской биржи](#) я использую значение дюрации (`DURATION`), а не беру готовое значение количества дней до погашения (`DAYSTOREDEMPTION`), ведь я пользуюсь собственным скриптом только в личных целях.

Распишу подробно все шаги которые нужны для работы моего скрипта.

Самое главное что понадобится для работы скрипта — [Node.js](#). Это — среда выполнения JavaScript. Если раньше JavaScript можно было запустить только в браузере, но сейчас
<https://habr.com/ru/post/506720/>

Скрипт, если раньше скрипт можно было запустить только в браузере, но однажды разработчики расширили его, и теперь можно запускать JS на своем компьютере в качестве отдельного приложения.

Исходный код моего скрипта [размещен на GitHub](#), и любой может свободно просматривать, проверять и может быть даже посоветует правки.

Поиск облигаций под Windows

Это будет самый подробный раздел, потому что большинство пользователей, которым это интересно, скорее всего, работают под Windows.

Для Windows доступен установщик Node.js [в разделе загрузить официального сайта](#).



Раздел загрузки сайта проекта Node.js

Далее скачиваем установщик для Windows и запускаем его.



Выбор компонентов для установки Node.js

Кроме компонентов, находящихся на этом экране, больше ничего устанавливать не надо.

[Скачиваем код скрипта с гитхаба](#).



Ссылка на скачивание с GitHub

После этого переходим каталог « `/SilverFir-Investment-Report-master/Node.js Release/bond_search_v2/` », где находятся скачанные файлы:



Каталог с необходимыми для запуска проекта файлами

И запускаем файл `first start.bat`, который содержит указание показать установленную текущую версию Node.js и установить необходимую для запуска проекта зависимость [node-fetch](#):

```
node -v
pause
npm install node-fetch
```

Несмотря на такое короткое содержание Защитник Windows проявляет бдительность, но если нажать подробнее, то можно увидеть кнопку **Выполнить в любом случае**:



Первоначальная настройка запуска проекта



Во время выполнения bat файла

После нажатия любой клавиши зависимость будет установлена в эту же папку:



Каталог вместе с добавленными файлами

После этого всё готово для запуска скрипта поиска облигаций. Для этого запускаем файл `start.bat`:



Выполнение скрипты поиска облигаций. После запуска файла start.bat

Менее чем за минуту будет создан HTML файлов с текущей датой и временем в имени — он и содержит в себе найденные результаты.

Поиск облигаций под macOS

Для macOS доступен установщик Node.js [в разделе загрузить официального сайта](#).

Сам процесс похож на установку под Windows и Linux.

Поиск облигаций под Linux

Если на вашем компьютере установлен Linux, скорее всего вы и сами знаете как лучше сделать. [Код скрипта доступен на гитхабе](#). Перейдите в каталог « `/SilverFir-Investment-Report-master/Node.js Release/bond_search_v2/` ».

Проверьте что Node.js установлена:

```
$ node -v
```

Проверьте что пакетный менеджер npm для Node.js установлен:

```
$ npm -v
```

Установите зависимости (в данном случае это только [node-fetch](#)):

```
$ npm install
```

Запустите файл скрипта:

```
$ npm start
```

Примерно за минуту html файл под именем файл `bond_search_${new Date().toLocaleString().replace(/\:/g, '-')} .html` будет создан.



Выполнение работы скрипта под Linux

Выборка облигаций

Я не сразу пришел именно к такой форме отчета, потому что я хотел чтобы этот файл отображался на любом устройстве и был удобен для просмотра. Больше всего проблем доставили айфоны — JS на них отключен и при пересылке этого отчета через любой мессенджер вместо интерактивной таблицы открывалось просто пустое место. Так что я дописал генератор обычных html таблиц.

Получились следующие виды:



На компьютере



На Android



На iPhone

Редактирование параметров выборки

Самое важное — настроить именно те параметры, которые важны именно вам, а не те, которые указал я для примера. Сделать это можно в файле `index.js`, со строки 42.



Задаваемые параметры поиска

Указываете нужные вам цифры, запускаете скрипт заново и примерно за минуту выборка готова.

Docker support

Пользователь [@supaflyster](#) сделал [форк с Docker версией](#):

- Скрипт запускается в докере (не нужно ставить nodejs и модули, обновление версии nodejs в Dockerfile) — проверял только на маке (linux тоже должен работать).
- HTML сохраняется в ./out/ (пришлось изменить, так как нужно примонтировать локальную папку в контейнер, куда будет сохраняться HTML).
- Изменил формирование имени файла, toLocaleString().replace(/:/g, '-') — зависит от локали, и если делитель не ":" то replace не отработает (в англ пытается создать bond_search_2020/5/22/11-00-00.html — что есть путь а не имя).
- docker-compose запускает контейнер (и соберет образ если он еще не собран) и передает параметры как environment variables, node читает их и если их нет то использует дефолтные из index.js — чтобы можно было запускать скрипт локально без докера.

Итог

Надеюсь что скрипт поможет экономить время и находить подходящие результаты, которые устраивают именно вас. Выборка «не является индивидуальной инвестиционной рекомендацией и может не соответствовать вашему инвестиционному профилю» — эту фразу я скопипастил, но она абсолютно верна, ведь сам скрипт это просто инструмент — решение о дальнейшей покупке конкретных найденных бумаг должен принимать уже человек после знакомства с эмитентом.

Скрипт работает только за счет API Московской биржи, которое предоставляет широкие возможности. Также хочу отметить, что я никак не связан с Московской биржей и использую ИСС Мосбиржи только в личных интересах.

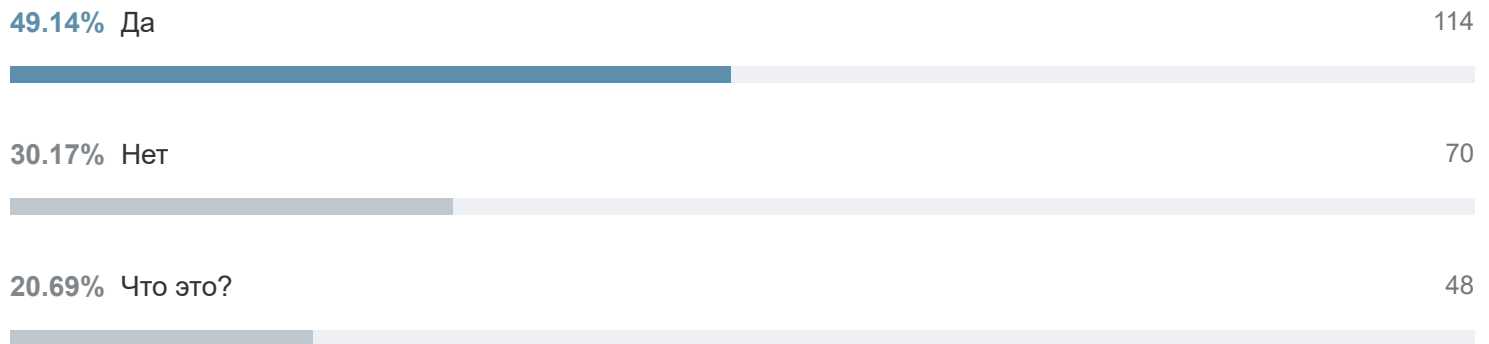
Обновил скрипт в начале 2021 года: [новая статья на Хабре](#).

Автор: [Михаил Шардин](#),

22 июня 2020 г.

Только зарегистрированные пользователи могут участвовать в опросе. [Войдите](#), пожалуйста.

Используете облигации?



Проголосовали 232 пользователя. Воздержались 26 пользователей.

Теги: парсинг, котировка, биржа, инвестиции, статистика, ценные бумаги, облигации, биржевая торговля, JavaScript, Node.JS

Хабы: JavaScript, Node.JS, API, Визуализация данных, Финансы в IT

Редакторский дайджест

Присылаем лучшие статьи раз в месяц

**107.5**

Карма

13.1

Рейтинг

Михаил Шардин @empenoso

Разработчик

[Сайт](#) Комментарии 114

ПОХОЖИЕ ПУБЛИКАЦИИ

24 августа 2020 в 06:25

Поиски фундаментальных данных для акций через API Financial Modeling Prep

◆ +4 👁 5.6K 📖 42 💬 6

8 июня 2020 в 07:39

Все финансовые рынки мира в API Яху Финанс

◆ +15 👁 39K 📖 129 💬 40 +40

4 февраля 2020 в 06:45

Что делает Free API Московской биржи в Google Таблицах

◆ +14 👁 62K 📖 165 💬 76 +76

МИНУТОЧКУ ВНИМАНИЯ



Ход мамонтом: что мы узнали о Yandex Scale 2021



Неделя аналитиков на Хабр Карьере!

ЗАКАЗЫ

Написать на JS пересчет итог. стоимости в зав-сти от кол-ва в корзине

1500 руб./за проект · 2 отклика · 7 просмотров

Внести правки в приложение

285000 руб./за проект · 16 просмотров

Разработка агрегатора недвижимости

240320 руб./за проект · 3 отклика · 24 просмотра

Доработать работа на Binance + написать бэк-тест-систему

590000 руб./за проект · 34 просмотра

Написать код для бекенда сайта NodeJs + Express + SocketIO + MySql

5000 руб./за проект · 3 отклика · 22 просмотра

[Больше заказов на Хабр Фрилансе](#)

ЛУЧШИЕ ПУБЛИКАЦИИ ЗА СУТКИ

вчера в 17:10

Что же не так с ДЭГ в Москве?

◆ +248

👁 34K

🔖 57

💬 175 +175

сегодня в 04:55

Данные выборов получили, теперь деобфусцируем и очищаем

◆ +99

👁 12K

🔖 27

💬 21 +21

вчера в 19:32

Получаем данные результатов выборов с сайта Центризбиркома РФ

◆ +73

👁 13K

🔖 30

💬 19 +19

сегодня в 02:26

Моя клубничная чудо-коробка

◆ +69

👁 6.1K

🔖 83

💬 38 +38

вчера в 21:49

Батарейки «Тест на правду»

◆ +51

👁 11K

🔖 6

💬 14 +14

Хабр



🌐 [Настройка языка](#)

[О сайте](#)

[Техническая поддержка](#)

[Полная версия](#)

[Вернуться на старую версию](#)

© 2006–2021 «Habr»