

Актуальные
зарплаты в IT

110k

100k

150k

120k

180k



empenoso

22 июн 2020 в 05:26

Скрипт выборки российских облигаций по параметрам



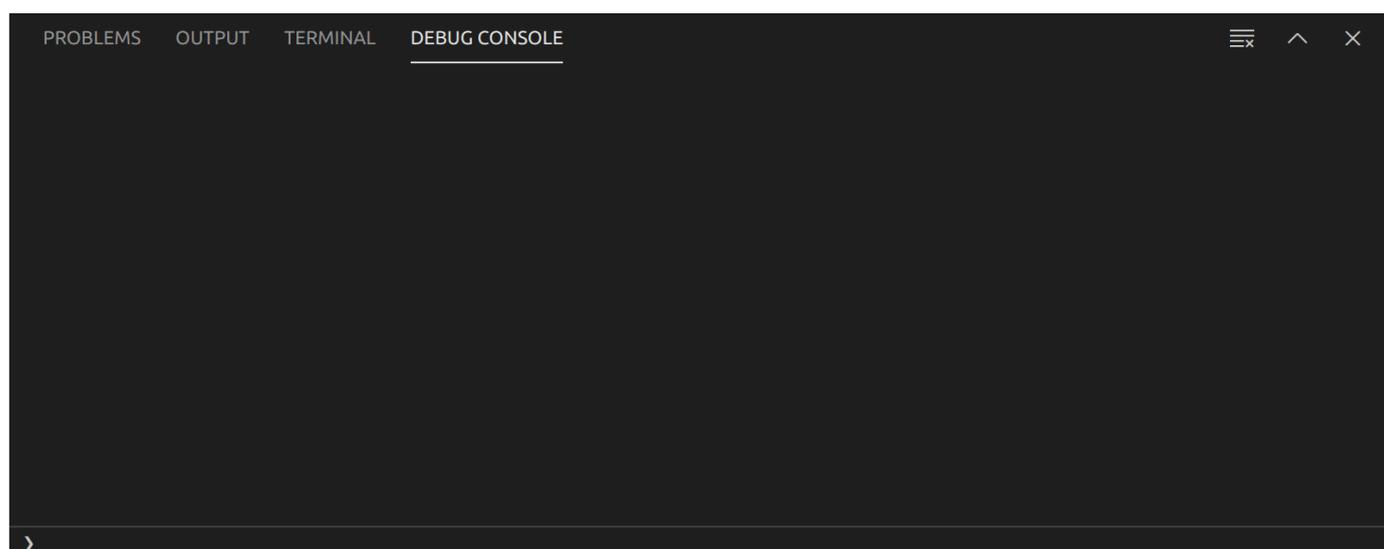
6 мин



32K

API*, JavaScript*, Node.JS*, 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:

- [sMart-lab.ru](#) — блоги трейдеров и инвесторов.
- [Cbonds](#). Рынок внутренних и международных облигаций.
- Облигации в России — [Rusbonds](#).
- [Bonds.Finam.RU](#) — информация о российском рынке облигаций.
- Сервис поиска от [Московской Биржи](#).
- [Анализ облигаций. Beta](#). 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:

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

А на выходе я бы хотел получать не больше 2-х десятков вариантов, которые точно попадали бы под моим 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 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.



где: p_i – цена i – ой сделки; n_i – объем i – ой сделки (в штуках ценных бумаг)

Пример:

Совершено 3 сделки: 3 акции по 1 руб., 3 акции по 3 руб. и 6 акций по 4 руб.

Тогда: $WAPRICE = \frac{1 \text{ руб.} \cdot 3 + 3 \text{ руб.} \cdot 3 + 4 \text{ руб.} \cdot 6}{3 + 3 + 6} = 3 \text{ руб.}$

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.

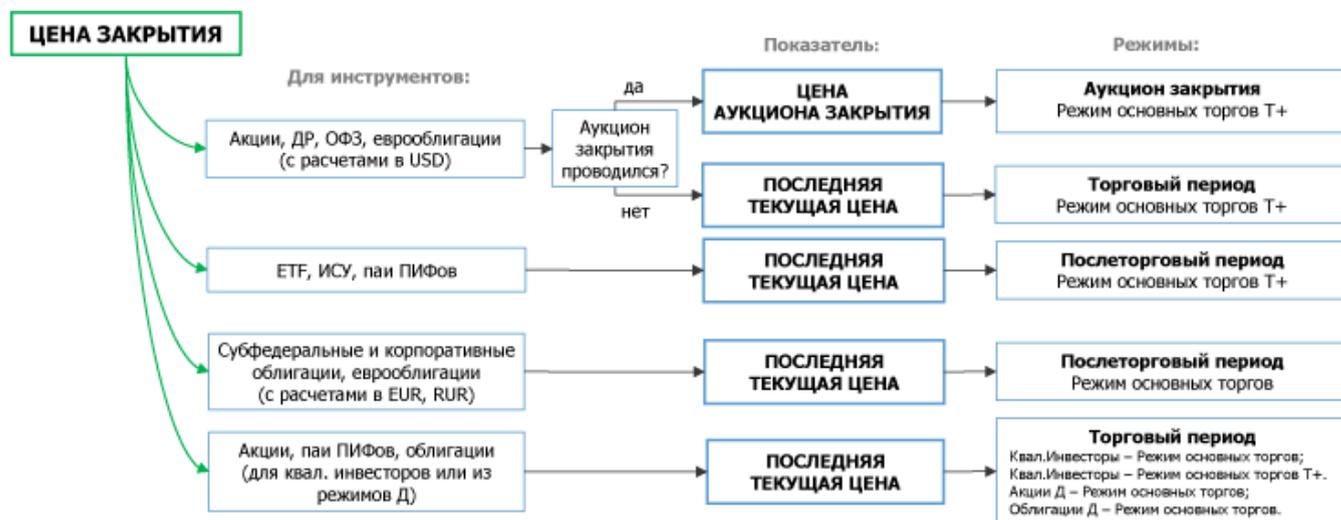


Схема определения цены закрытия (`LEGALCLOSEPRICE`)

Чтобы уменьшить количество обращений к [API Московской биржи](https://habr.com/ru/post/506720/) 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.

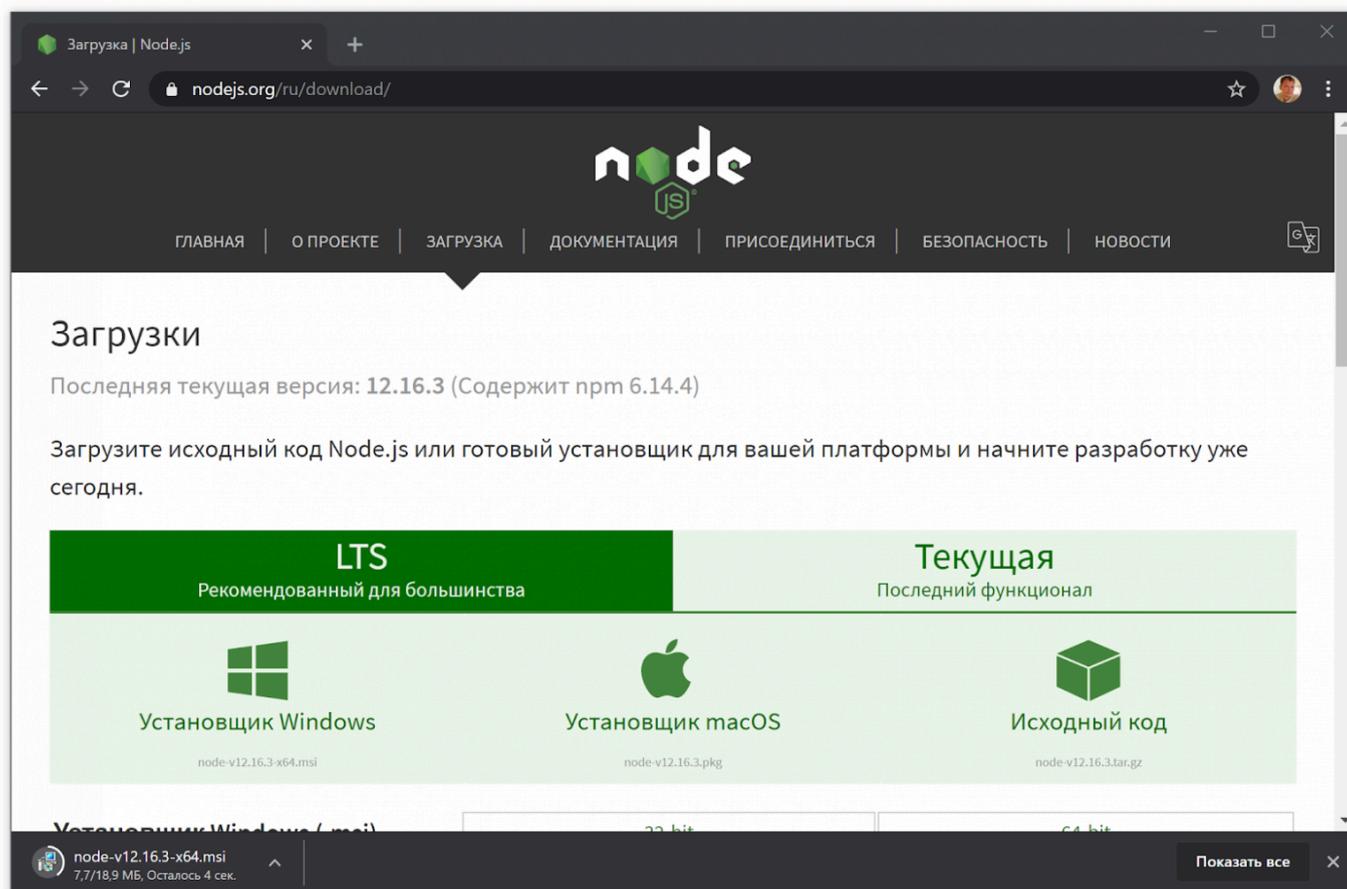
Самое главное что понадобится для работы скрипта — [Node.js](#). Это — среда выполнения JavaScript. Если раньше JavaScript можно было запустить только в браузере, но однажды разработчики расширили его, и теперь можно запускать JS на своем компьютере в качестве отдельного приложения.

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

Поиск облигаций под Windows

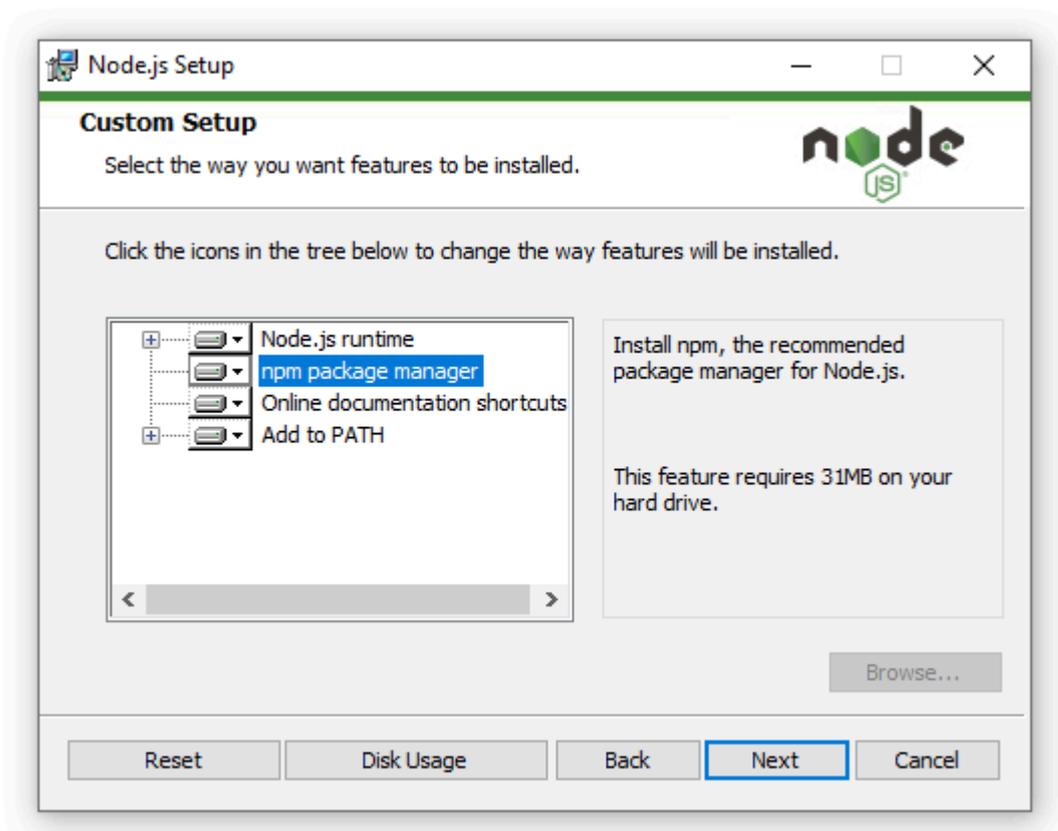
Это будет самый подробный раздел, потому что большинство пользователей, которым это интересно, скорее всего, работают под Windows.

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.

 1 environment
 1 contributor
 Apache-2.0

Create new file
Upload files
Find file
Clone or download ▾

Clone with HTTPS ?

Use SSH

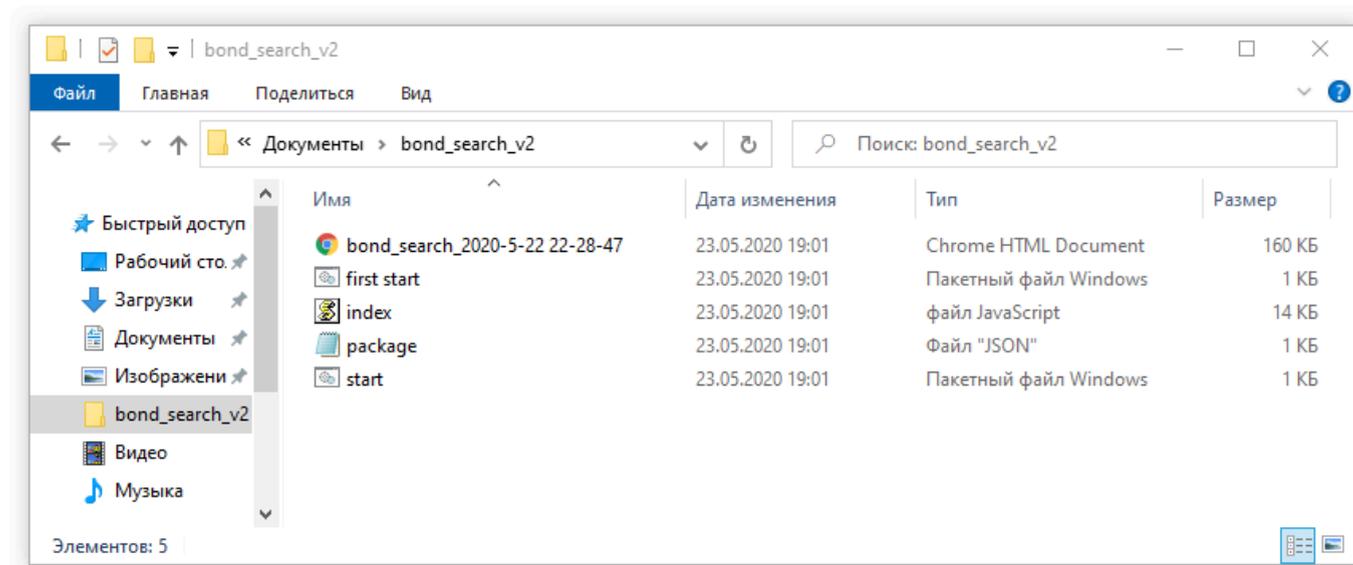
Use Git or checkout with SVN using the web URL.

https://github.com/empenoso/SilverFir
📄

Download ZIP

Ссылка на скачивание с GitHub

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



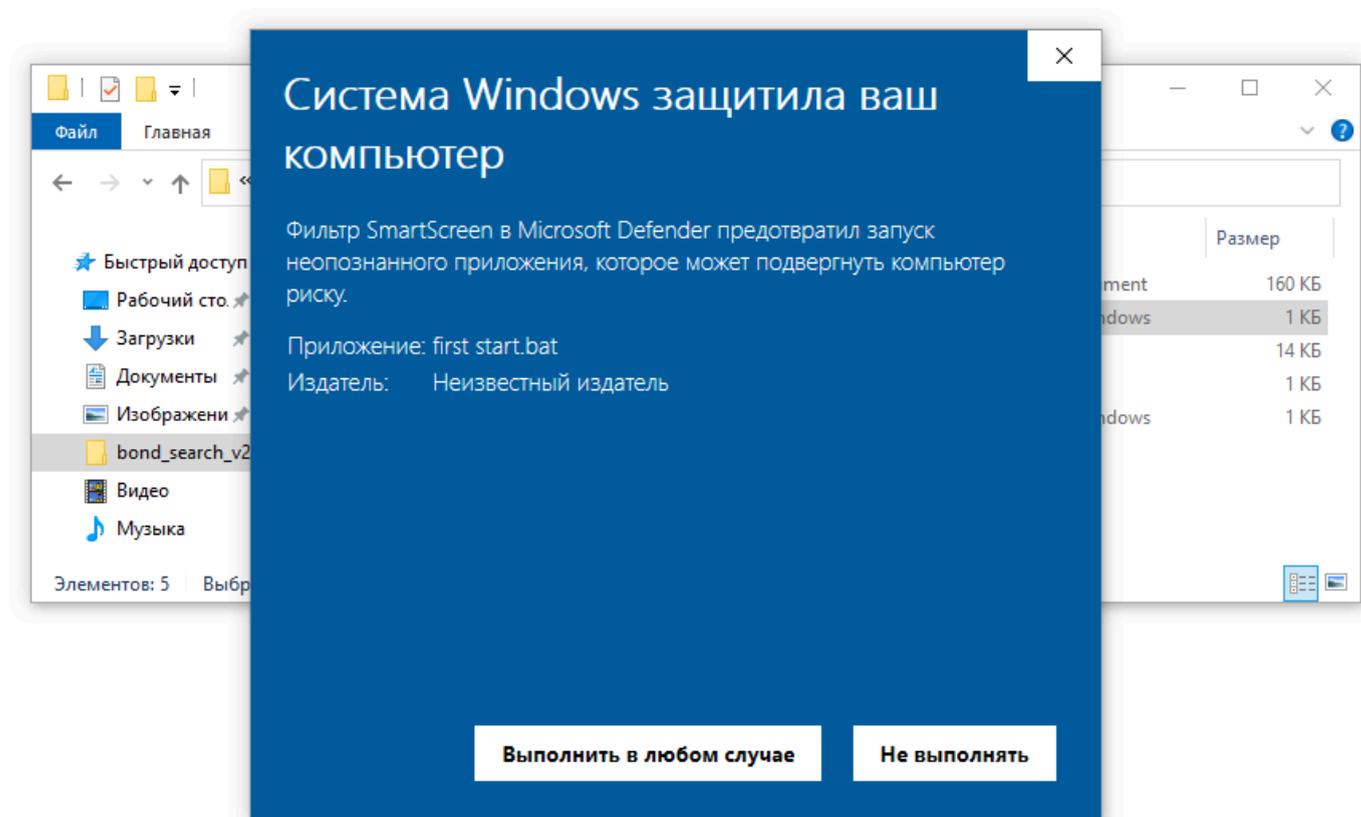
Каталог с необходимыми для запуска проекта файлами

И запускаем файл `first start.bat`, который содержит указание показать установленную текущую версию Node.js и установить необходимую для запуска проекта

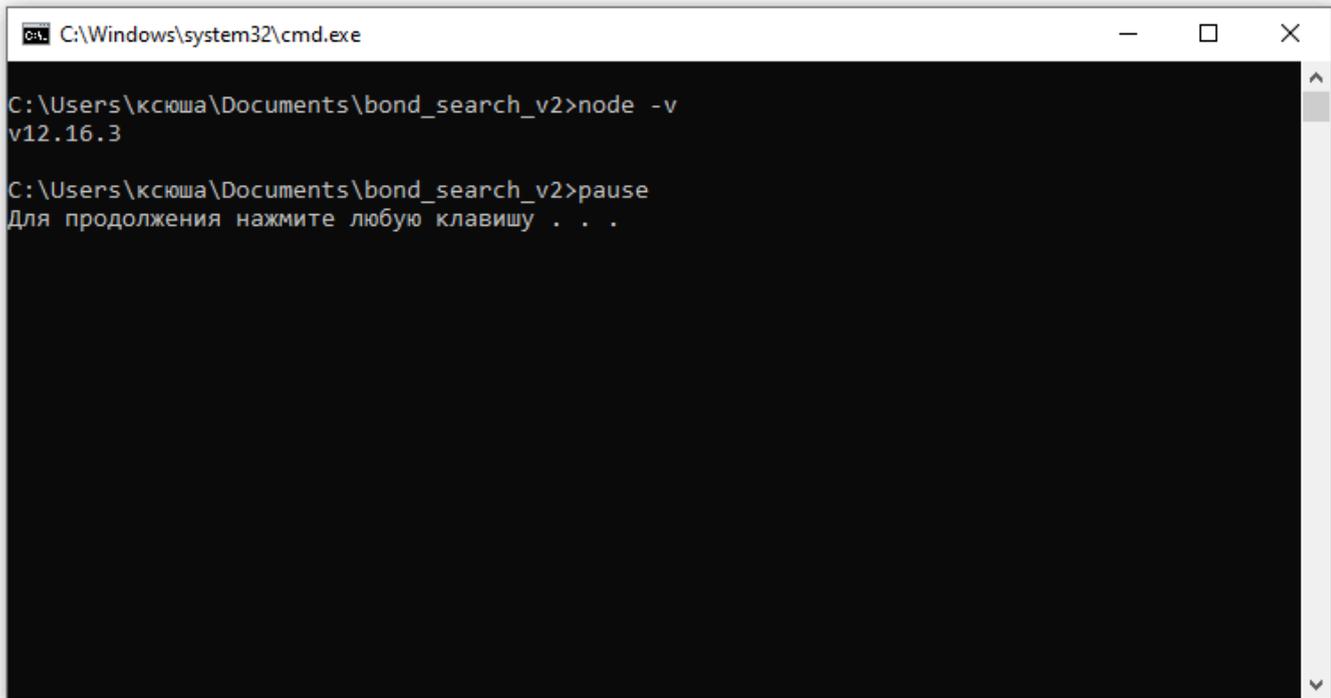
зависимость `node-fetch`:

```
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



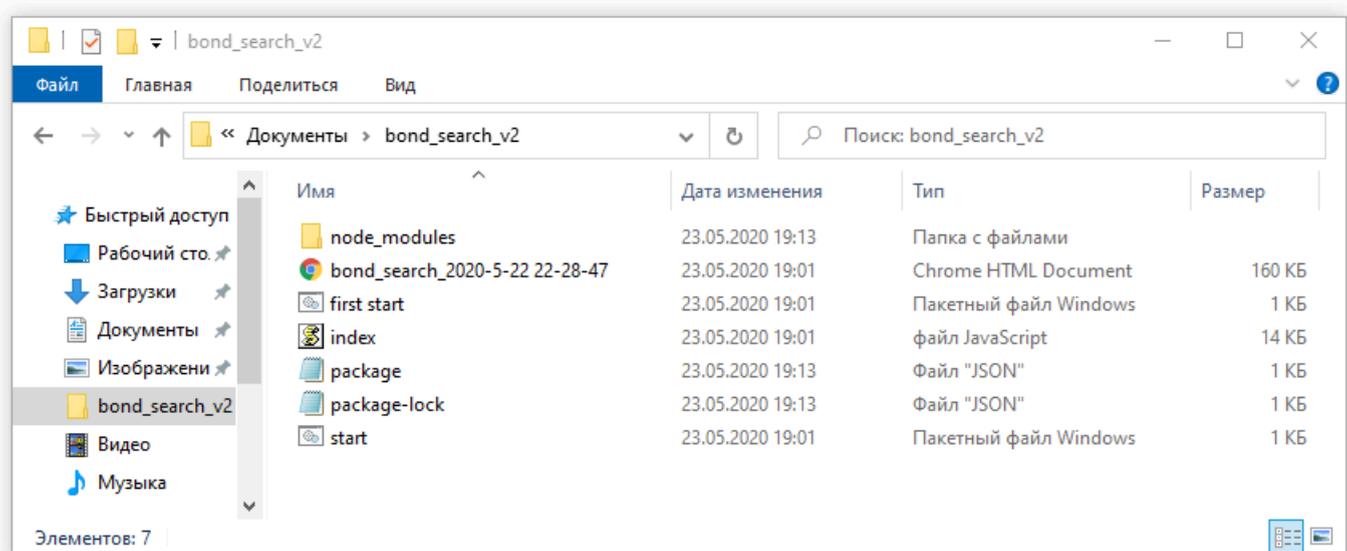
```
C:\Windows\system32\cmd.exe

C:\Users\ксюша\Documents\bond_search_v2>node -v
v12.16.3

C:\Users\ксюша\Documents\bond_search_v2>pause
Для продолжения нажмите любую клавишу . . .
```

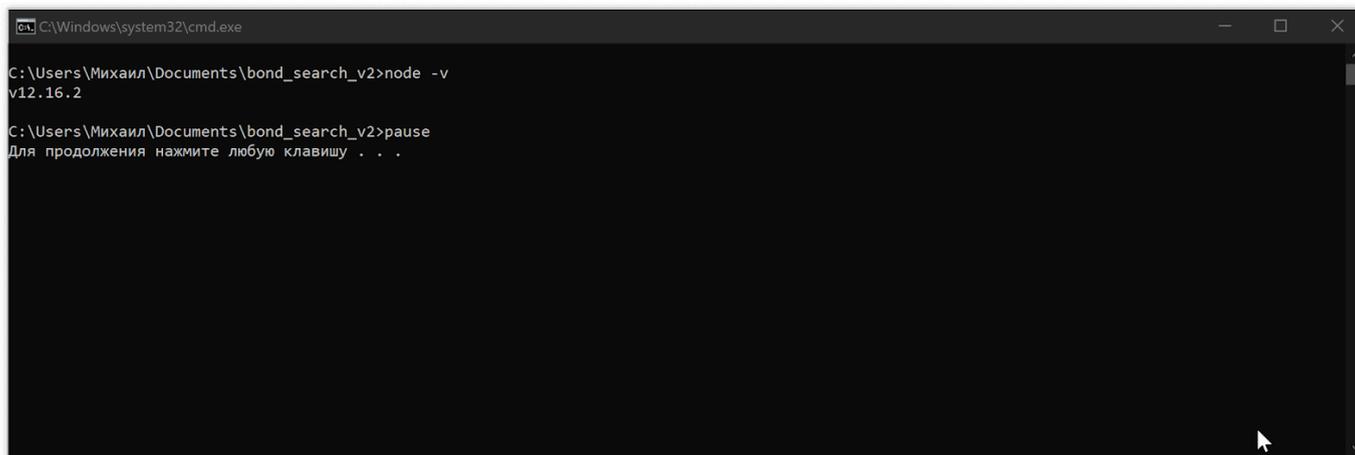
Во время выполнения bat файла

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`:



```
C:\Windows\system32\cmd.exe
C:\Users\Михаил\Documents\bond_search_v2>node -v
v12.16.2
C:\Users\Михаил\Documents\bond_search_v2>pause
Для продолжения нажмите любую клавишу . . .
```

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
```

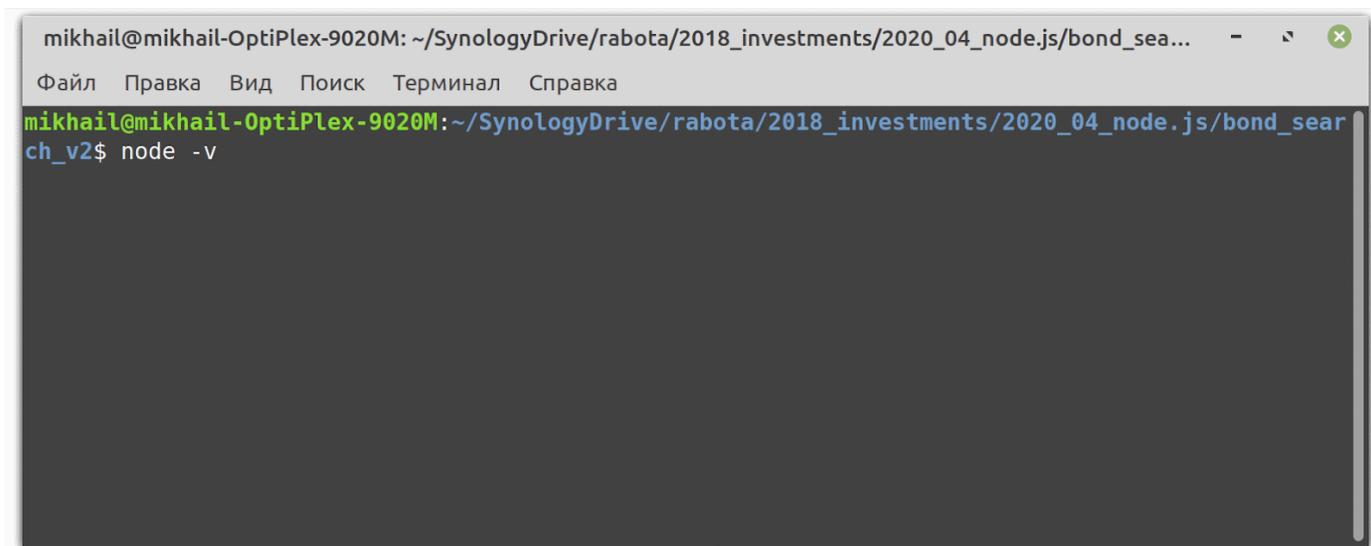
Установите зависимости (в данном случае это только `node-fetch`):

```
$ 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.



```
mikhail@mikhail-OptiPlex-9020M: ~/SynologyDrive/rabota/2018_investments/2020_04_node.js/bond_sea...
Файл  Правка  Вид  Поиск  Терминал  Справка
mikhail@mikhail-OptiPlex-9020M:~/SynologyDrive/rabota/2018_investments/2020_04_node.js/bond_sea
ch_v2$ node -v
```

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:

file:///home/mikhail/SynologyDrive/rabota/2018_investments/2020_04 ☆

	Полное наименование	Код ценной бумаги	Цена, %	Объем сделок за n дней, шт.	Доходность	Дюрация, месяцев	Есть льгота?
1	АО ЛК Европлан БО-02	RU000A0JVQT7	100.55	1,030,020	8.33	14.33	✗
2	ТрансФин-М ПАО 001P-04	RU000A0ZYEB1	100	195,026	10.73	5.09	✓
3	РН БАНК АО об. 01	RU000A0JWMJ5	99.65	100,098	6.88	13.2	✗
4	Евразийский банк развития 08	RU000A0JS900	100.91	100,003	5.57	4.59	✗
5	СК Легион ООО БО-01	RU000A0ZZNV7	100.2	72,639	13.04	4.53	✓
6	Банк ВТБ (ПАО) Б-1-62	RU000A101JT3	100.34	62,601	5.7	10.03	✓
7	Акрон (ПАО) об. сер. 05	RU000A0JRHG1	100.8	57,289	5.74	11.83	✗
8	ЛЕГЕНДА БО 001P-01	RU000A0ZZCV0	100.05	50,106	14.57	7.96	✓
9	Банк ВТБ (ПАО) Б-1-49	RU000A101C22	100.18	39,327	5.7	7.9	✓
10	Онлайн Микрофинанс БО-02	RU000A101JF2	100	36,532	12.34	9.36	✓
11	Банк ВТБ (ПАО) Б-1-41	RU000A1012L2	100.34	34,363	5.86	5.96	✓
12	Банк ВТБ (ПАО) Б-1-50	RU000A101FV7	100.1	30,579	5.64	8.93	✓
13	Банк ВТБ (ПАО) Б-1-63	RU000A101M53	100.15	30,319	5.75	11.13	✓
14	Банк ВТБ (ПАО) Б-1-38	RU000A100Y01	100.5	25,083	5.67	4.9	✓
15	Удмуртская респ. 2013 обл.	RU000A0JU740	100.71	23,974	5.5	4.4	✗
16	РСГ-Финанс ООО обл. БО-04	RU000A0ZYLB6	100.34	18,939	10.3	6.66	✓
17	Совкомбанк ПАО обл. БО-05	RU000A0ZYJR6	100.91	18,601	6.83	11.93	✓
18	ЧЗПСН-Профнастил ПАО БО-П01	RU000A100DF7	100.39	18,509	14.12	13.93	✓
19	КарМани БО-1-2	RU000A0ZZ1F6	100.88	18,002	13.85	4.96	✓
20	Кредит Европа Банк АО 001P-02	RU000A0ZZXP8	100.3	15,111	6.68	6.46	✓

Выборка сгенерирована 2020-5-22 22:28:47 по условиям:

- 5% < Доходность < 15%
- 95% < Цена < 101%
- 4 мес. < Дюрация < 15 мес.
- Объем сделок за n дней > 15000 шт.
- Поиск в T0, T+, T+ (USD) - Основной режим - безадрес.

Составил [Михаил Шардин](#).

► [Техническая информация](#)

На компьютере

19:34 G G G H+ 55 %

content://org.telegram.messenger.provider/media/Telegram/Telegram% 7

	Полное наименование	Код ценной бумаги	Цена, %	Объем сделок за n дней, шт.	Доходность	Дюрация, месяцев	Есть льгота?
1	АО ЛК Европлан БО-02	RU000A0JVQT7	100.55	1,030,020	8.33	14.33	✗
2	ТрансФин-М ПАО 001P-04	RU000A0ZYEB1	100	195,026	10.73	5.09	✓
3	РН БАНК АО об. 01	RU000A0JWMJ5	99.65	100,098	6.88	13.2	✗
4	Евразийский банк развития 08	RU000A0JS900	100.91	100,003	5.57	4.59	✗
5	СК Легион ООО БО-01	RU000A0ZZNV7	100.2	72,639	13.04	4.53	✓
6	Банк ВТБ (ПАО) Б-1-62	RU000A101JT3	100.34	62,601	5.7	10.03	✓
7	Акрон (ПАО) об. сер. 05	RU000A0JRHG1	100.8	57,289	5.74	11.83	✗
8	ЛЕГЕНДА БО 001P-01	RU000A0ZZCV0	100.05	50,106	14.57	7.96	✓
9	Банк ВТБ (ПАО) Б-1-49	RU000A101C22	100.18	39,327	5.7	7.9	✓
10	Онлайн Микрофинанс БО-02	RU000A101JF2	100	36,532	12.34	9.36	✓
11	Банк ВТБ (ПАО) Б-1-41	RU000A1012L2	100.34	34,363	5.86	5.96	✓
12	Банк ВТБ (ПАО) Б-1-50	RU000A101FV7	100.1	30,579	5.64	8.93	✓
13	Банк ВТБ (ПАО) Б-1-63	RU000A101M53	100.15	30,319	5.75	11.13	✓
14	Банк ВТБ (ПАО) Б-1-38	RU000A100Y01	100.5	25,083	5.67	4.9	✓
15	Удмуртская респ. 2013 обл.	RU000A0JU740	100.71	23,974	5.5	4.4	✗
16	РСГ-Финанс ООО обл. БО-04	RU000A0ZYLB6	100.34	18,939	10.3	6.66	✓
17	Совкомбанк ПАО обл. БО-05	RU000A0ZYJR6	100.91	18,601	6.83	11.93	✓

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.

```
bond_search_v2 > JS index.js > MOEXsearchBonds
37
38  async function MOEXsearchBonds() { //поиск облигаций по параметрам
39    const YieldMore = 7 //Доходность больше этой цифры
40    const YieldLess = 14 //Доходность меньше этой цифры
41    const PriceMore = 95 //Цена больше этой цифры
42    const PriceLess = 101 //Цена меньше этой цифры
43    const DurationMore = 1 //Дюрация больше этой цифры
44    const DurationLess = 6 //Дюрация меньше этой цифры
45    const VolumeMore = 5000 //Объем сделок за n дней, шт. больше этой цифры
```

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 делиметр 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

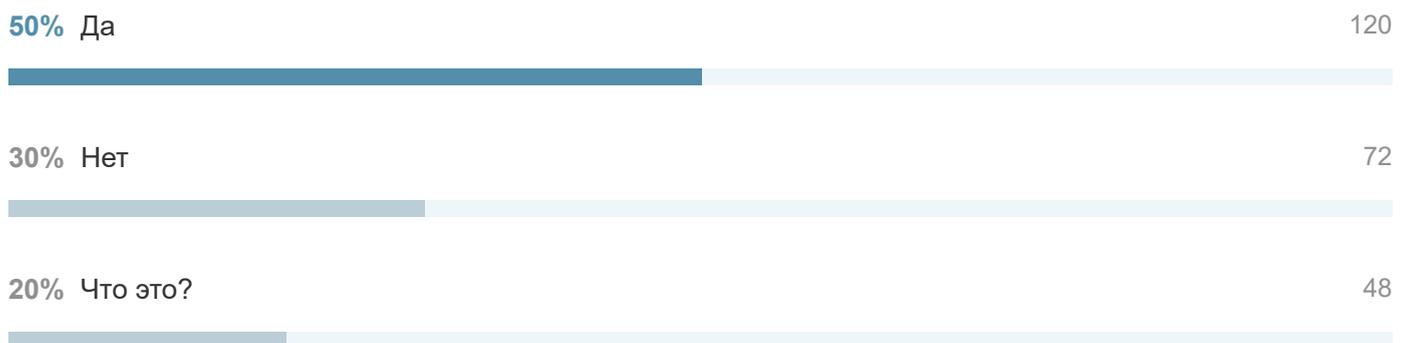
[Моя онлайн-визитка](#)

[Telegram «Умный Дом Инвестора»](#)

22 июня 2020 г.

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

Используете облигации?



Проголосовали 240 пользователей. Воздержались 26 пользователей.

Теги: [парсинг](#), [котировка](#), [биржа](#), [инвестиции](#), [статистика](#), [ценные бумаги](#), [облигации](#), [биржевая торговля](#), [JavaScript](#), [Node.JS](#)

Хабы: [API](#), [JavaScript](#), [Node.JS](#), [Визуализация данных](#), [Финансы в IT](#)

Редакторский дайджест



Присылаем лучшие статьи раз в месяц

**183**

Карма

87.1

Рейтинг

Михаил Шардин @empenoso

Автоматизация / Данные / Финансы / Умные дома

[Подписаться](#)[Сайт](#) [Сайт](#) [Github](#)

Комментарии 114

Публикации

ЛУЧШИЕ ЗА СУТКИ

ПОХОЖИЕ



rssdev10

23 часа назад

Почему въехав по «визе талантов» в США я с радостью вернулся в Россию



Средний



32 мин



35K

Мнение

+176

113

518



melnik909

19 часов назад

Вы не знаете CSS. Мои вопросы о CSS с ответами. Часть 2



Средний



7 мин



2.1K

Обзор

+39

32

1



DAN_SEA

17 часов назад

Генерация случайных чисел



Средний



10 мин



2.4K

Обзор

+32

25

34

**OrkBiotechnologist**

23 часа назад

VPS за 139 рублей — дом для вашего резюме на основе Hugo

**Простой**

7 мин



8.4K

Тutorial

**+28**

48



12

**PatientZero**

2 часа назад

Пишем стек TCP/IP с нуля: Ethernet, ARP, IPv4 и ICMPv4

**Простой**

13 мин



927

Tutorial

Перевод

**+20**

34



1

**tertiumnon**

17 часов назад

Минимум книг, которые нужно прочитать начинающему или продолжающему свою кривую обучения программисту

**Простой**

3 мин



9K

Обзор

**+19**

195



21

**lbankter**

1 час назад

Бэктдор Auto-color: разбор угрозы, технический анализ и способы защиты

**Средний**

4 мин



303

Обзор

**+15**

5



2

**FlatSpike**

19 часов назад

Создаём многомодульную библиотеку на Android: как же собрать fat-aar?

Средний 19 мин 534

[Кейс](#)

+13

16

0

**alexander-shustanov**

19 часов назад

В поисках идеального Database-клиента для IDE: Amplicode выбирает DBeaver

Простой 6 мин 2.6K

+13

13

7

**ptsecurity**

21 час назад

Безопасность без боли: плагины, которые упрощают жизнь разработчикам

7 мин 942

[Кейс](#)

+12

13

3

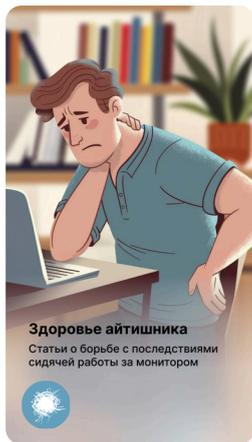
Бегущий по лезвию ИИ — 2025: сезон футурологии на Хабре с крутыми призами

[Турбо](#)[Показать еще](#)

ИСТОРИИ



Как расти на работе?



Здоровье айтишника



Угадайте будущее в новом сезоне



С праздником весны!



Всегда котов!

КУРСЫ

 **Специалист по кибербезопасности с нуля до Junior**

По факту набора · GB (GeekBrains)

 **Курс «Тестировщик с нуля до Junior»**

По факту набора · GB (GeekBrains)

 **Системный аналитик**

По желанию · Яндекс Практикум

 **Инженер по тестированию**

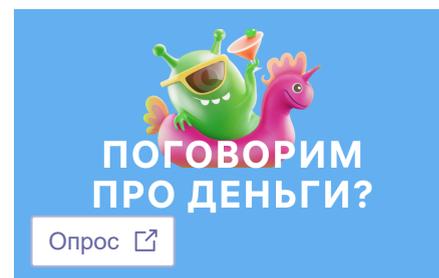
По желанию · Яндекс Практикум

 **Тестировщик ПО с нуля**

По факту набора · Skyprom

[Больше курсов на Хабр Карьере](#)

МИНУТОЧКУ ВНИМАНИЯ



[Девушка с розовыми волосами и Слизень на планете Рбах](#)

[Казино и аутсорс: как работают айтишники в РБ](#)

[Как айтишники подходят к финансовому планированию?](#)

РАБОТА

[JavaScript разработчик](#)

134 вакансии

[React разработчик](#)

48 вакансий

[Node.js разработчик](#)

56 вакансий

[Все вакансии](#)

БЛИЖАЙШИЕ СОБЫТИЯ

Выиграйте грант на профессию

и обучитесь новой специальности бесплатно

Хекслет

Advertisement for a grant competition. The main text reads 'Выиграйте грант на профессию' (Win a grant for a profession) and 'и обучитесь новой специальности бесплатно' (and learn a new specialty for free). The logo 'Хекслет' (Hekslet) is in the top right. The background is black with blue squares and a white code symbol '</>'. A right arrow icon is in the bottom right corner.

17 февраля – 24 марта

Конкурс «Снежный код» от Хекслета. Три гранта на бесплатное 10-месячное обучение

Онлайн

Разработка

Больше событий в календаре

Хабр



🌐 Настройка языка

Техническая поддержка

© 2006–2025, Habr