

[Все потоки](#) [Разработка](#) [Администрирование](#) [Дизайн](#) [Менеджмент](#) [Маркетинг](#) [Наука](#)

empenoso 8 июня 2020 в 07:39

Все финансовые рынки мира в API Яху Финанс

Python *, Алгоритмы *, Node.JS *, API *, Финансы в IT

В этой статье я расскажу об API агрегатора финансовых данных [Yahoo! Finance](#). В рассказе есть один нюанс — официальное API Яху Финанс было [закрыто три года назад](#), однако практически сразу же появилась его недокументированная работоспособная версия, которая жива до сих пор. Хочу в исследовательских целях рассказать об использовании этой работоспособной версии подробнее.

Тем более, что список рынков, данные с которых можно получать через Яху Финанс [огромен](#). На текущий момент в нем 79 стран, включая и Россию.

The screenshot shows the Yahoo Finance website interface. At the top, there's a navigation bar with links like Home, Mail, News, Finance, Sports, Entertainment, Search, Mobile, and More... Below this is a search bar and a 'Sign in' button. The main content area features a 'Leadership Development' advertisement, a section for COVID-19 confirmed cases, and a grid of market futures data including S&P, Dow, Nasdaq, Russell 2000, Crude Oil, and Gold. There are also news snippets about stock futures, Amazon being sued, and the Senate passing a bill. On the right, there's a 'My Portfolio & Markets' section with a table of recently viewed funds.

Symbol	Last Price	Change	% Change
VTI	158.35	+2.43	+1.56%
CSCO	46.94	+0.05	+0.11%
SPY	312.09	+4.01	+1.30%

Apple Inc. (AAPL) на сайте и в API Яху Финанс

Взгляд на данные с позиции долгосрочного частного инвестора

Ниже рассмотрю получение только тех параметров, которые, на мой взгляд, могут быть

1. Название бумаги
2. Цена бумаги
3. Доход с начала года
4. Дивидендная доходность
5. Дата предыдущего дивиденда
6. Значение предыдущего дивиденда
7. Годовая плата для фондов
8. Категория бумаг

API Яху Финанс предоставляет ответы в формате JSON. На скриншотах с использованием API я использую расширение для браузера Google Chrome: [JSON-handle](#).

Название бумаги / Name

Имея только тикер всегда можно получить огромное число параметров, первым в списке, на мой взгляд стоит наименование инструмента. Сначала [найдем его на сайте](#) на примере SPDR S&P 500 ETF Trust (SPY):



Имя ETF на сайте Яху Финанс

После этого найдем имя уже через API — в формате JSON оно выводится в двух вариантах: `shortName` и `longName`. Ссылка на данные, включающая в себя имя выглядит следующим образом:

```
https://query1.finance.yahoo.com/v10/finance/quoteSummary/SPY?modules=price
```



Имя ETF через API Яху Финанс

Свойство с именем `longName` содержит строковое значение `SPDR S&P 500 ETF Trust`. Путь для получения этого ключа: `JSON.quoteSummary.result[0].price.longName`

Цена бумаги / Price

Следующий важный параметр — конечно цена. Найдем её для акций [Berkshire Hathaway Inc. \(BRKA\)](#):



Цена акций на сайте Яху Финанс

Дальше получим цену через API. Раздел где содержится цена, аналогичен получению имени:

```
https://query1.finance.yahoo.com/v10/finance/quoteSummary/BRKA?modules=price
```



Цена акций через API Яху Финанс

Ключ с именем `raw` для `regularMarketPrice` можно получить как с форматированием, так и без него. Путь для получения этого свойства:

```
JSON.quoteSummary.result[0].price.regularMarketPrice.raw
```

Доход с начала года / YTD Daily Total Return

Этот параметр можно посмотреть только для фондов. На примере [Vanguard Total Stock Market Index Fund ETF Shares \(VTI\)](#):



Доход с начала года на сайте Яху Финанс

Посмотрим этот параметр через API. Раздел где содержится доход с начала года:

```
https://query1.finance.yahoo.com/v10/finance/quoteSummary/VTI?modules=defaultKeyStatistics
```



Доход с начала года через API Яху Финанс

Свойство с именем `fmt` для `ytdReturn` можно получить как с форматированием, так и без. Путь для получения этого свойства:

```
JSON.quoteSummary.result[0].defaultKeyStatistics.ytdReturn.fmt
```

Дивидендная доходность / Dividend Yield

Важный параметр, финансовые консультанты даже складывают доходность бумаги с дивидендной доходностью и показывают получившуюся цифру как потенциал роста бумаги. Найдем её для [VanEck Vectors Russia ETF \(RSX\)](#):



Дивидендная доходность ETF на сайте Яху Финанс

Получим этот параметр через API. Раздел где содержится дивидендная доходность:

<https://query1.finance.yahoo.com/v10/finance/quoteSummary/RX?modules=defaultKeyStatistics>



Дивидендная доходность ETF через API Яху Финанс

Свойство с именем `fmt` для `yield` можно получить как с форматированием, так и без. Путь для получения этого свойства:

```
JSON.quoteSummary.result[0].defaultKeyStatistics.yield.fmt
```

Дата предыдущего дивиденда / Dividend Date

Найдем дату последней выплаты дивиденда. Для этого придется обращаться к истории и можно будет получить все дивидендные выплаты. На сайте эти данные в разделе [Historical Data](#), возьмем например акции Microsoft Corporation (MSFT):



Дата предыдущего дивиденда акции на сайте Яху Финанс

Получить их через API задача уже немного сложнее, потому что ссылка будет иметь вид:

<https://query1.finance.yahoo.com/v8/finance/chart/MSFT?symbol=MSFT&period1=1559457037&period2=1591079437&interval=1mo&includePrePost=true&events=div%7Csplitted>

Где:

- `period1` начальная дата в виде [Unix Timestamp](#).
- `period2` — конечная дата в виде [Unix Timestamp](#).
- `interval=1mo` — укрупненные свечи, меня интересуют только дивиденды.
- `events=div%7Csplitted` — добавляет информацию о дивидендах и сплитах в вывод.



Дата предыдущего дивиденда акции через API Яху Финанс

Для получения даты возможны два варианта:

1. Считывать ключи `JSON.chart.result[0].timestamp` и перебирать по этим ключам даты дивидендов.
2. Более предпочтительный — получать массив значений перечисляемых свойств объекта

```
JSON.chart.result[0].events.dividends .
```

Значение предыдущего дивиденда / Next Dividend

Полностью аналогично предыдущему разделу. Только ищем не дату, а значение. Найдем значение прошлого дивиденда [для ETF iShares MSCI Mexico Capped ETF \(EWW\)](#):



Значение предыдущего дивиденда ETF на сайте Яху Финанс

В API ссылка будет выглядеть:

```
https://query1.finance.yahoo.com/v8/finance/chart/EWW?
symbol=MSFT&period1=1559457037&period2=1591079437&interval=1mo&includePrePost=true&
events=div%7Csplitt
```

Расшифровка запроса аналогична получению даты выше.



Значение предыдущего дивиденда ETF через API Яху Финанс

Годовая плата / Expense Ratio

Годовая плата, которую все фонды или ETF взимают со своих акционеров. На сайте можно посмотреть [в разделе Summary](#):



Годовая плата, которую все фонды или ETF взимают со своих акционеров на сайте Яху Финанс

В API ссылка будет выглядеть:

```
https://query1.finance.yahoo.com/v10/finance/quoteSummary/HYD?modules=fundProfile
```



Годовая плата, которую все фонды или ETF взимают со своих акционеров через API Яху Финанс

Свойство с именем `fmt` для `annualReportExpenseRatio` можно получить как с форматированием, так и без. Путь для получения этого свойства:

```
JSON.quoteSummary.result[0].fundProfile.feesExpensesInvestment.annualReportExpense
Ratio.fmt
```

Категория акций / Sector и Industry

На сайте можно посмотреть в разделе [Profile](#). Для примера возьмем [Cisco Systems, Inc. \(CSCO\)](#):



Категория акции на сайте Яху Финанс

Эти данные можно посмотреть только для акций и возможных вариантов секторов не так уж и много:

1. Basic Materials
2. Consumer Cyclical
3. Financial Services
4. Real Estate
5. Consumer Defensive
6. Healthcare
7. Utilities
8. Communication Services
9. Energy
10. Industrials
11. Technology

Подкатегории для секторов — Industry.

В API ссылка будет выглядеть:

```
https://query1.finance.yahoo.com/v10/finance/quoteSummary/CSCO?modules=assetProfile
```



Категория акции через API Яху Финанс

Свойство с ключом `sector` можно получить по следующему пути:

```
JSON.quoteSummary.result[0].assetProfile.sector
```

Общие правила работы с API Яху Финанс

Хост

`query1.finance.yahoo.com` для HTTP / 1.0

`query2.finance.yahoo.com` для HTTP / 1.1

Основные данные

`/v10/finance/quoteSummary/GOOGL?modules=` (Полный список модулей ниже)

(замените `GOOGL` на любой символ)

Входные данные для `?modules=` запроса:

1. `assetProfile`
2. `incomeStatementHistory`
3. `incomeStatementHistoryQuarterly`
4. `balanceSheetHistory`
5. `balanceSheetHistoryQuarterly`
6. `cashflowStatementHistory`
7. `cashflowStatementHistoryQuarterly`
8. `defaultKeyStatistics`
9. `financialData`
10. `calendarEvents`
11. `secFilings`
12. `recommendationTrend`
13. `upgradeDowngradeHistory`
14. `institutionOwnership`
15. `fundOwnership`
16. `majorDirectHolders`
17. `majorHoldersBreakdown`
18. `insiderTransactions`
19. `insiderHolders`
20. `netSharePurchaseActivity`
21. `earnings`
22. `earningsHistory`

22. earningsHistory

23. earningsTrend

24. industryTrend

25. indexTrend

26. sectorTrend

и возможно что-то ещё...

Пример URL:

```
https://query1.finance.yahoo.com/v10/finance/quoteSummary/GOOGL?modules=assetProfile%2Cear
```

Запрос для: `assetProfile` и `earningsHistory` .

История цен, сплитов и дивидендов

```
/v8/finance/chart/GOOGL?symbol=GOOGL&period1=0&period2=9999999999&interval=3mo
```

Интервалы:

`&interval=3mo` это 3 месяца.

`&interval=1d` это 1 день.

`&interval=5m` это 5 минут, возвращает 80 дней.

`&interval=1m` это 1 минута, возвращает 4-5 дней.

`period1=` unix timestamp представление даты, с которой вы хотите начать. Значения ниже начальной торговой даты будут округлены до начальной торговой даты.

`period2=` unix timestamp представление даты, на которой вы хотите закончить. Значения, превышающие последнюю торговую дату, будут округлены до последней доступной отметки времени.

Добавить данные до и после рынка: `&includePrePost=true`

Добавить дивиденды и сплиты: `&events=div%2Csplit`

Пример полного запроса:

```
https://query1.finance.yahoo.com/v8/finance/chart/GOOGL?symbol=AAPL&period1=0&period2=9999
```

Приведенный выше запрос вернет все данные о цене тикера `GOOGL` с интервалом в 1 день, включая данные до и после рынка, а также дивиденды и сплиты.

Всю информацию об API можно получить изучая код страницы Яху Финанс, но первоначально информация была взята с [Stack Overflow](#) и [GitHub](#).

Пример кода на Node.js



```
const fetch = require('node-fetch');
async function USAStockGetName(ID) { //получаем имя бумаги
  const url = `https://query1.finance.yahoo.com/v10/finance/quoteSummary/${ID}?modules=price`;
  // console.log("USAStockGetName. url для %s: %s", ID, url);
  try {
    const response = await fetch(url)
    const json = await response.json()
    const value = json.quoteSummary.result[0].price.longName
    console.log("USAStockGetName. Название для %s: %s", ID, value)
    if (value == 0) return 'нет'
    return value
  } catch (e) {
    console.log('Ошибка в USAStockGetName')
  }
}
module.exports.USAStockGetName = USAStockGetName
```

Пример кода на Python

→ Можно посмотреть [на GitHub](#)

Итог

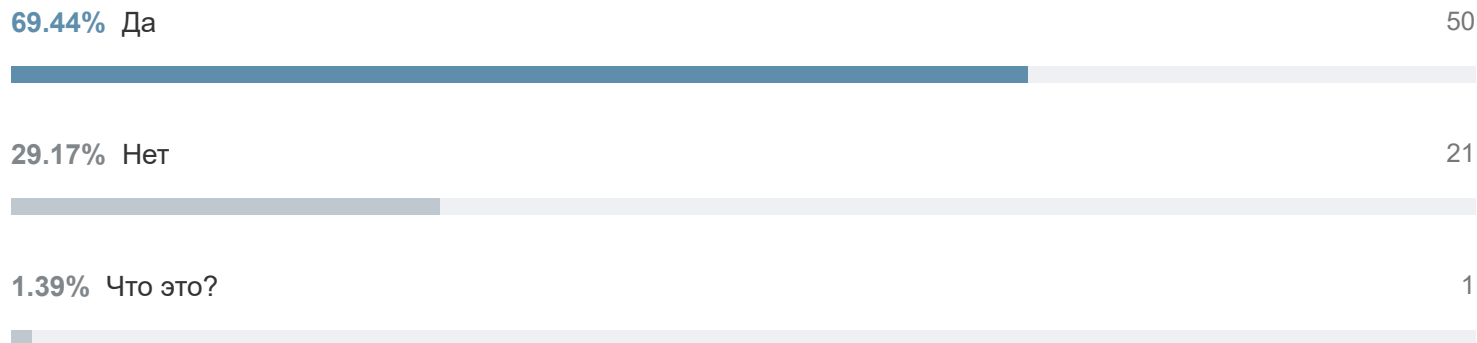
Я написал эту статью, желая в исследовательских целях разобраться в работающем API Яху Финанс, содержащем подробные данные о десятках тысяч ценных бумаг по всему миру, включая Россию.

Автор: [Михаил Шардин](#),

8 июня 2020 г.

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

Используете финансовые API?



Проголосовали 72 пользователя. Воздержались 16 пользователей.

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

Хабы: Python, Алгоритмы, Node.JS, API, Финансы в IT

Редакторский дайджест

Присылаем лучшие статьи раз в месяц

**107.5**

Карма

13.1

Рейтинг

Михаил Шардин @empenoso

Разработчик

[Сайт](#)

ПОХОЖИЕ ПУБЛИКАЦИИ

24 августа 2020 в 06:25

Поиски фундаментальных данных для акций через API Financial Modeling Prep

+4

5.7K

42

6

22 июня 2020 в 07:26

Скрипт выборки российских облигаций по параметрам

+45

23K

209

114

4 февраля 2020 в 06:45

Что делает Free API Московской биржи в Google Таблицах

+14

62K

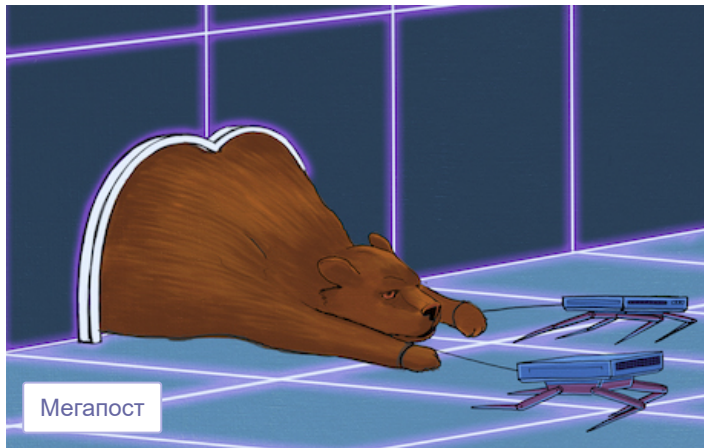
165

76 +76

МИНУТОЧКУ ВНИМАНИЯ



В шортах уже зябко, а промокод согревает



Как мы учили файерволы строить каналы и рулить сетями

ВОПРОСЫ И ОТВЕТЫ

Как достать список всех ютуб каналов?

YouTube · Простой · 0 ответов

Как получить src атрибут тега img при парсинге с помощью BS4?

Python · Простой · 1 ответ

Как спарсить url картинок через гугл таблицы?

Парсинг · Простой · 1 ответ

Как спарсить ссылку на продавца Kufar?

Python · Простой · 1 ответ

Как перейти по ссылке через картинку, не имея ссылки в div?


Парсинг · Простой · 0 ответов

[Больше вопросов на Хабр Q&A](#)

ЛУЧШИЕ ПУБЛИКАЦИИ ЗА СУТКИ


вчера в 18:00

Разгадываем ребус вторичных часов «Воронеж»

 **+53**  3.8K  20  4 **+4**

вчера в 14:00

Позолоченный век интернета: как монополии уничтожают свободный рынок, который их создал

 **+43**  6.5K  44  17 **+17**

вчера в 11:31

Потребительский экстремизм в GameDev

 **+39**  8K  46  46 **+46**

вчера в 13:50

Опыт миграции из Gitea в GitLab. Сложно, но успешно

 **+38**  2.5K  4  22 **+22**

вчера в 19:38

Микросервисы. Не всё то золото, что хайп

 **+30**  6K  43  26 **+26**



 [Настройка языка](#)

[О сайте](#)

[Техническая поддержка](#)

[Полная версия](#)

[Вернуться на старую версию](#)

© 2006–2021 «Habr»