



Ran Aroussi

Ran Aroussi [aroussi.com]
@aroussi

Reliably download historical market data from with Python

Ever since Yahoo decommissioned their historical data API, Python developers looked for a reliable workaround. As a result, my library, **yfinance**, gained momentum and gets 300k+ 3M+ installs per month, according to PyPi!

17 APRIL 2019 RAN AROUSSI #PYTHON #YFINANCE

Ever since Yahoo decommissioned their historical data API, Python developers looked for a reliable workaround. As a result, my library, **yfinance** [<https://github.com/ranaroussi/yfinance>], gained momentum and gets 300k+ 3M+ installs per month, according to PyPi!

Legal note:

Yahoo!, Y!Finance, and Yahoo! finance are registered trademarks of Yahoo, Inc.

yfinance is not affiliated, endorsed, or vetted by Yahoo, Inc. It's an open-source tool that uses Yahoo's publicly available APIs, and is intended for research and educational purposes.

You should refer to Yahoo!'s terms of use (here

[<https://policies.yahoo.com/us/en/yahoo/terms/product-atos/apiforydn/index.htm>], **here**

[<https://legal.yahoo.com/us/en/yahoo/terms/otos/index.html>], **and here**

[<https://policies.yahoo.com/us/en/yahoo/terms/index.htm>]) **for details on your rights to use the actual data downloaded. Remember - the Yahoo! finance API is intended for personal use only.**

fix-yahoo-finance aimed to offer a temporary fix [<https://aroussi.com/post/fix-yahoo-finance>] to the problem by getting data from Yahoo! Finance and returning it in the same format as **pandas_datareader's** `get_data_yahoo()`, thus keeping the code changes in existing software to minimum.

The problem was, that this hack was a bit unreliable, causing data to not being downloaded and required developers to force session re-initialization and re-fetching of cookies, by calling `yf.get_yahoo_crumb(force=True)`.

yfinance is a complete re-write of the library, offering a reliable method of downloading historical market data from Yahoo! Finance's API, up to 1 minute granularity, in a more Pythonic way.

Introducing the `Ticker()` module:

The `Ticker()` module allows you get market and meta data for a security, using a Pythonic way:

```

1  import yfinance as yf
2
3  msft = yf.Ticker("MSFT")
4  print(msft)
5  """
6  returns
7  <yfinance.Ticker object at 0x1a1715e898>
8  """
9
10 # get stock info
11 msft.info
12
13 """
14 returns:
15 {
16   'quoteType': 'EQUITY',
17   'quoteSourceName': 'Nasdaq Real Time Price',
18   'currency': 'USD',
19   'shortName': 'Microsoft Corporation',
20   'exchangeTimezoneName': 'America/New_York',
21   ...
22   'symbol': 'MSFT'
23 }
24 """
25
26 # get historical market data
27 msft.history(period="max")
28 """
29 returns:
30
31      Open    High    Low    Close    Volume  Dividends  Splits
32  Date
33  1986-03-13    0.06    0.07    0.06    0.07  1031788800      0.0    0.0
34  1986-03-14    0.07    0.07    0.07    0.07   308160000      0.0    0.0
35  ...
36  2019-04-15  120.94  121.58  120.57  121.05   15792600      0.0    0.0
37  2019-04-16  121.64  121.65  120.10  120.77   14059700      0.0    0.0
38  """
39
40 # show actions (dividends, splits)
41 msft.actions
42 """
43 returns:
44
45      Dividends  Splits
46  Date
47  1987-09-21      0.00    2.0
48  1990-04-16      0.00    2.0
49  ...
50  2018-11-14      0.46    0.0
51  2019-02-20      0.46    0.0
52  """
53
54 # show dividends
55 msft.dividends
56 """
57 returns:
58
59      Date
60  2003-02-19    0.08
61  2003-10-15    0.16
62  ...
63  2018-11-14    0.46
64  2019-02-20    0.46

```

```

62     """
63
64     # show splits
65     msft.splits
66     """
67     returns:
68     Date
69     1987-09-21    2.0
70     1990-04-16    2.0
71     ...
72     1999-03-29    2.0
73     2003-02-18    2.0
74     """

```

Available paramaters for the `history()` method are:

- **period:** data period to download (Either Use period parameter or use start and end) Valid periods are: 1d, 5d, 1mo, 3mo, 6mo, 1y, 2y, 5y, 10y, ytd, max
- **interval:** data interval (intraday data cannot extend last 60 days) Valid intervals are: 1m, 2m, 5m, 15m, 30m, 60m, 90m, 1h, 1d, 5d, 1wk, 1mo, 3mo
- **start:** If not using period - Download start date string (YYYY-MM-DD) or datetime.
- **end:** If not using period - Download end date string (YYYY-MM-DD) or datetime.
- **prepost:** Include Pre and Post market data in results? (Default is `False`)
- **auto_adjust:** Adjust all OHLC automatically? (Default is `True`)
- **actions:** Download stock dividends and stock splits events? (Default is `True`)

Mass download of market data:

You can also download data for multiple tickers at once, like before.

```

1  import yfinance as yf
2  data = yf.download("SPY AAPL", start="2017-01-01", end="2017-04-30")

```

To access the closing price data for **SPY**, you should use: `data['Close']['SPY']`.

If, however, you want to group data by Symbol, use:

```

1  import yfinance as yf
2  data = yf.download("SPY AAPL", start="2017-01-01", end="2017-04-30",
3                      group_by="ticker")

```

To access the closing price data for **SPY**, you should use: `data['SPY']['Close']`.

The `download()` method accepts an additional parameter - `threads` for faster completion when downloading a lot of symbols at once.

* **NOTE:** To keep compatibility with older versions, **auto_adjust** defaults to **False** when using mass-download.

Using pandas_datareader:

If your legacy code is using **pandas_datareader** and you want to keep the code changes to minimum, you can simply call the override method and keep your code as it was:

```
1  from pandas_datareader import data as pdr
2
3  import yfinance as yf
4  yf.pdr_override() # <== that's all it takes :-)
5
6  # download dataframe using pandas_datareader
7  data = pdr.get_data_yahoo("SPY", start="2017-01-01", end="2017-04-30")
```

To install/upgrade **yfinance** using pip, run:

```
$ pip install yfinance --upgrade --no-cache-dir
```

The **Github repository** [<https://github.com/ranaroussi/yfinance>] has more information and issue tracking.

Enjoy!

Copyrights © Ran Aroussi [aroussi.com]. All rights reserved.

Updated on 17 April 2019. For the latest version and comments, please see:
<https://aroussi.com/post/python-yahoo-finance>