Hacking Google Finance in Real-Time for Algorithmic Traders. (2) Pre-Market Trading.

may 7, 2015 by [pawel](http://www.quantatrisk.com/author/admin/) [6 comments](http://www.quantatrisk.com/2015/05/07/hacking-google-finance-in-pre-market-trading-python/#disqus_thread)

***Featured in:****Data Science Weekly Newsletter,*[*Issue 76*](http://www.datascienceweekly.org/newsletters/data-science-weekly-newsletter-issue-76) (May 7, 2015)

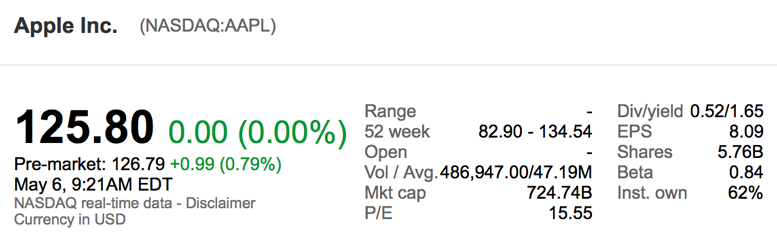
It has been over a year since I posted [Hacking Google Finance in Real-Time for Algorithmic Traders](http://www.quantatrisk.com/2014/01/14/hacking-google-finance-in-real-time-for-algorithmic-traders/) article. Surprisingly, it became the number one URL of QaR that Google has been displaying as a result to various queries and the number two most frequently read post. Thank You! It’s my pleasure to provide quality content covering interesting topics that I find potentially useful.

You can be surprised how fast Python solutions went forward facilitating life of quants and algo traders. For instance, yesterday, haphazardly, I found a code that seems to work equally well as compared to my first version, and, in fact, is more flexible in data content that could be retrieved. The idea stays the same as previously, however, our goal this time is to**monitor changes of stock prices**provided by [Google Finance](https://www.google.com/finance) in real-time **before** the market opens.

**Constructing Pre-Market Price-Series**

The **pre-market trading** session typically occurs between 8:00am and 9:30am EDT each trading day though for some stocks we often observe frequent movements much earlier, e.g. at 6:00am. Many investors and traders watch the pre-market trading activity to judge the strength and direction of the market in anticipation for the regular trading session. Pre-market trading activity generally has limited volume and liquidity, and therefore, large bid-ask spreads are common. Many retail brokers offer pre-market trading, but may limit the types of orders that can be used during the pre-market period11.

In Google Finance the stock price in pre-market is usually displayed right beneath the tricker, for example:

[](http://www.quantatrisk.com/wp-content/uploads/2015/05/AAPLpm1.png)

The price of the stock (here: AAPL) varies depending on interest, good/bad news, etc.

In Python we can fetch those changes (I adopt a code found on the Web) in the following way:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | **import** urllib2 *# works fine with Python 2.7.9 (not 3.4.+)*  **import** json  **import** time    **def** fetchPreMarket(symbol, exchange):  link = "http://finance.google.com/finance/info?client=ig&q="  url = link+"%s:%s" % (exchange, symbol)  u = urllib2.urlopen(url)  content = u.read()  data = json.loads(content[3:])  info = data[0]  t = str(info["elt"]) *# time stamp*  l = float(info["l"]) *# close price (previous trading day)*  p = float(info["el"]) *# stock price in pre-market (after-hours)*  **return** (t,l,p)      p0 = 0  **while** True:  t, l, p = fetchPreMarket("AAPL","NASDAQ")  **if**(p!=p0):  p0 = p  **print**("%s**\t**%.2f**\t**%.2f**\t**%+.2f**\t**%+.2f%%" % (t, l, p, p-l,  (p/l-1)\*100.))  time.sleep(60) |

In this code we target Google to get every 60 seconds an update of the pre-market price (line #14). What we retrieve is a JSON file of the form:

|  |
| --- |
| // [  {  "id": "22144"  ,"t" : "AAPL"  ,"e" : "NASDAQ"  ,"l" : "125.80"  ,"l\_fix" : "125.80"  ,"l\_cur" : "125.80"  ,"s": "1"  ,"ltt":"4:02PM EDT"  ,"lt" : "May 5, 4:02PM EDT"  ,"lt\_dts" : "2015-05-05T16:02:28Z"  ,"c" : "-2.90"  ,"c\_fix" : "-2.90"  ,"cp" : "-2.25"  ,"cp\_fix" : "-2.25"  ,"ccol" : "chr"  ,"pcls\_fix" : "128.7"  ,"el": "126.10"  ,"el\_fix": "126.10"  ,"el\_cur": "126.10"  ,"elt" : "May 6, 6:35AM EDT"  ,"ec" : "+0.30"  ,"ec\_fix" : "0.30"  ,"ecp" : "0.24"  ,"ecp\_fix" : "0.24"  ,"eccol" : "chg"  ,"div" : "0.52"  ,"yld" : "1.65"  ,"eo" : ""  ,"delay": ""  ,"op" : "128.15"  ,"hi" : "128.45"  ,"lo" : "125.78"  ,"vo" : "21,812.00"  ,"avvo" : "46.81M"  ,"hi52" : "134.54"  ,"lo52" : "82.90"  ,"mc" : "741.44B"  ,"pe" : "15.55"  ,"fwpe" : ""  ,"beta" : "0.84"  ,"eps" : "8.09"  ,"shares" : "5.76B"  ,"inst\_own" : "62%"  ,"name" : "Apple Inc."  ,"type" : "Company"  }  ] |

You can download it individually if we execute in the browser a query:

|  |
| --- |
| http://www.google.com/finance/info?infotype=infoquoteall&q=NASDAQ:AAPL |

Some of those information you can easily decipher. For our task we need to get only: *el*(an asset price in pre-market or after-hours trading; a.k.a. extended hours trading);*elt* (corresponding time stamp); and *l* (most recent last price). This is what our Python code does for us in lines #12-14. Nice and smoothly.

When executed before 9.30am EDT (here for NASDAQ:AAPL), we may **construct** the pre-market price-series every time the price has been changed:

|  |
| --- |
| May 6, 6:35AM EDT 125.80 126.18 +0.38 +0.30%  May 6, 6:42AM EDT 125.80 126.21 +0.41 +0.33%  May 6, 6:45AM EDT 125.80 126.16 +0.36 +0.29%  May 6, 6:46AM EDT 125.80 126.18 +0.38 +0.30%  May 6, 6:49AM EDT 125.80 126.10 +0.30 +0.24%  May 6, 6:51AM EDT 125.80 126.20 +0.40 +0.32%  May 6, 6:57AM EDT 125.80 126.13 +0.33 +0.26%  May 6, 7:00AM EDT 125.80 126.20 +0.40 +0.32%  May 6, 7:01AM EDT 125.80 126.13 +0.33 +0.26%  May 6, 7:07AM EDT 125.80 126.18 +0.38 +0.30%  May 6, 7:09AM EDT 125.80 126.20 +0.40 +0.32%  May 6, 7:10AM EDT 125.80 126.19 +0.39 +0.31%  May 6, 7:10AM EDT 125.80 126.22 +0.42 +0.33%  May 6, 7:12AM EDT 125.80 126.20 +0.40 +0.32%  May 6, 7:22AM EDT 125.80 126.27 +0.47 +0.37%  May 6, 7:28AM EDT 125.80 126.24 +0.44 +0.35%  ...  May 6, 9:02AM EDT 125.80 126.69 +0.89 +0.71%  May 6, 9:03AM EDT 125.80 126.71 +0.91 +0.72%  May 6, 9:04AM EDT 125.80 126.73 +0.93 +0.74%  May 6, 9:08AM EDT 125.80 126.67 +0.87 +0.69%  May 6, 9:09AM EDT 125.80 126.69 +0.89 +0.71%  May 6, 9:10AM EDT 125.80 126.68 +0.88 +0.70%  May 6, 9:13AM EDT 125.80 126.67 +0.87 +0.69%  May 6, 9:14AM EDT 125.80 126.72 +0.92 +0.73%  May 6, 9:16AM EDT 125.80 126.74 +0.94 +0.75%  May 6, 9:17AM EDT 125.80 126.72 +0.92 +0.73%  May 6, 9:18AM EDT 125.80 126.70 +0.90 +0.72%  May 6, 9:19AM EDT 125.80 126.73 +0.93 +0.74%  May 6, 9:20AM EDT 125.80 126.75 +0.95 +0.76%  May 6, 9:21AM EDT 125.80 126.74 +0.94 +0.75%  May 6, 9:21AM EDT 125.80 126.79 +0.99 +0.79% (\*)  May 6, 9:23AM EDT 125.80 126.78 +0.98 +0.78%  May 6, 9:24AM EDT 125.80 126.71 +0.91 +0.72%  May 6, 9:25AM EDT 125.80 126.73 +0.93 +0.74%  May 6, 9:26AM EDT 125.80 126.75 +0.95 +0.76%  May 6, 9:27AM EDT 125.80 126.70 +0.90 +0.72%  May 6, 9:28AM EDT 125.80 126.75 +0.95 +0.76%  May 6, 9:29AM EDT 125.80 126.79 +0.99 +0.79% |

Since the prices in pre-market tend to vary slowly, 60 second time interval is sufficient to keep our eye on the stock. You can compare a live result retrieved using our Python code at 9:21am (\*) with the above screenshot I took at the same time.

A simple joy of Python in action. Enjoy!

**HOMEWORK**  
     1. The code fails after 9.30am EST (NYC time). Modify it to catch this exception.  
     2. Modify the code (or write a new function) that works after 9.30am EDT.  
     3. It is possible to get N>1N>1 queries for NN stocks by calling, for example:  
            NASDAQ:AAPL,NYSE:JNJ,… in line #7 of the code. Modify the program  
            to fetch pre-market time-series, xi(t)xi(t) (i=1,…,N)(i=1,…,N), for NN-asset portfolio.  
            Given that, compute a fractional root-mean-square volatility, σxi(t)/⟨xi(t)⟩σxi(t)/⟨xi(t)⟩,  
            i.e. standard deviation divided by the mean, between 6am and 9.30am EDT  
            for each asset and check can you use it as an indicator for stock price movement  
            after 9.30am? Tip: the higher frms the more trading is expected in first 15 min  
            of a new session at Wall Street.  
     4. Modify the code to monitor **after-hours trading** till 4.30pm.

**RELATED POSTS**  
    [Hacking Google Finance in Real-Time for Algorithmic Traders](http://www.quantatrisk.com/2014/01/14/hacking-google-finance-in-real-time-for-algorithmic-traders/)

**FURTHER READING**  
    Chenoweth, M., 2011, [Downloading Google Intraday historical data with Python](http://trading.cheno.net/downloading-google-intraday-historical-data-with-python/)  
    NetworkError.org, 2013, [Google’s Undocumented Finance API](http://www.networkerror.org/component/content/article/1-technical-wootness/44-googles-undocumented-finance-api.html)

**REFERENCES**  
    11Pre-Market, *Investopedia*, <http://www.investopedia.com/terms/p/premarket.asp>

filed under: [python for quants](http://www.quantatrisk.com/category/python-for-quants/) tagged with: [python](http://www.quantatrisk.com/tag/python/), [time-series](http://www.quantatrisk.com/tag/time-series/)