nsetools Documentation

Release 1.0.4

Vivek Jha

Feb 10, 2020

Contents

Python library for extracting realtime data from National Stock Exchange (India)

CHAPTER 1

Table Of Contents

1.1 Introduction

nsetools is a library for collecting real time data from National Stock Exchange (India). It can be used in various types of projects which requires fetching live quotes for a given stock or index or building large data sets for further data analytics. You can also build cli applications which can provide you live market details at a blazing fast speeds, much faster than any browser. The accuracy of data is only as correct as provided on http://www.nseindia.com

Note: The data provided by APIs is only as correct as provided on http://www.nseindia.com

1.2 Github Project Page

https://github.com/vsjha18/nsetools

1.3 Main Features

1. Works out of box, without any required setup.
2. Fetches live stock code and index codes in blazing fast speed.
3. Provides list of all indices and stocks traded in National Stock Exchange.
4. **Additionally provides list of:**
    - Top losers.
    - Top gainers.
    - Most active.
5. Provides some useful APIs to validate a stock code and index code.

6. Optionally returns data in JSON format.
7. Hunderd Percent Unittest coverage.

1.4 Installation

Installing nsetools is very simple and it has no external dependencies. All its dependencies are part of standard python distribution. packages:

```
pip install nsetools
```

1.5 Update

To updated to the lasted version:

```
pip install nsetools --upgrade
```

1.6 Python 3 Support

Python 3 support has been added from 1.0.0 and onwards. Now this library works for both Python 2 and Python 3.

1.7 A Word On Exception Handling

Since this library would form a middleware of some other project. Hence it is not handling any exception. Apart from standard python exceptions, you should be catching the following two:

- URLError
- HTTPError

Warning: You need to have a working internet connection while using this library. It will raise URLErorr in case there is no internet connectivity. Hence please handle this scenario in your code.

Warning: If you are facing any issue with the APIs then it may be beacuse there had been some format change recently in the way NSE reports its live quotes. Please upgrade to the latest version in order to avoid this issue.

Warning: Now nsetools doesn't support python 2 officially, though it may still continue to work with python 2 for some more time.

1.8 API Walkthrough & Examples

In this section we will focus on the basic usage and cover all the APIs which nsetools offer. Though I would encourage you to take a look at the code and unittests in case you want to further customize it.

Note: All the data APIs support json return as well. Call any API with as_json=True to get the json return. For example following would provide a json return.

nse.get_quote('infy', as_json=True)

By default every API returns python data structures.

1.8.1 Instantiation

nsetools uses www.nseindia.com as a data source.

As mentioned earlier, nsetools comes pre-built with all the right url mappings and hence instantiating it requires no contructor arguments.

```
>>> from nsetools import Nse
>>> nse = Nse()
>>> print nse
Driver Class for National Stock Exchange (NSE)
```

Note: Please make sure that you are connected to internet while using this library. It will raise URLError in case of any network glitch.

1.8.2 Getting a Stock Quote

Before going though other fundamental APIs. We will first see how to get a quote. Assume that we want to fetch current price of *Infosys Technology*. The only thing we need is NSE Code for this company. The NSE stock code for *Infosys* is **INFY**.

```
>>> q = nse.get_quote('infy') # it's ok to use both upper or lower case for codes.
>>> from pprint import pprint # just for neatness of display
>>> pprint(q)
{'adhocMargin': None,
 'applicableMargin': 12.5,
 'averagePrice': 1999.82,
 'bcEndDate': None,
 'bcStartDate': None,
 'buyPrice1': 1999.45,
 'buyPrice2': 1999.4,
 'buyPrice3': 1999.35,
 'buyPrice4': 1999.15,
 'buyPrice5': 1999.1,
 'buyQuantity1': 50.0,
 'buyQuantity2': 209.0,
 'buyQuantity3': 22.0,
 'buyQuantity4': 1.0,
 'buyQuantity5': 24.0,
```

(continues on next page)

(continued from previous page)

```
 'change': 25.35,
 'closePrice': None,
 'cm_adj_high_dt': '01-DEC-14',
 'cm_adj_low_dt': '30-MAY-14',
 'cm_ffm': 190659.16,
 'companyName': 'Infosys Limited',
 'css_status_desc': 'Listed',
 'dayHigh': 2010.0,
 'dayLow': 1972.0,
 'deliveryQuantity': 258080.0,
 'deliveryToTradedQuantity': 51.54,
 'exDate': '02-DEC-14',
 'extremeLossMargin': 5.0,
 'faceValue': 5.0,
 'high52': 2201.1,
 'indexVar': None,
 'isinCode': 'INE009A01021',
 'lastPrice': 1999.75,
 'low52': 1440.0,
 'marketType': 'N',
 'ndEndDate': None,
 'ndStartDate': None,
 'open': 1972.0,
 'pChange': 1.28,
 'previousClose': 1974.4,
 'priceBand': 'No Band',
 'pricebandlower': 1777.0,
 'pricebandupper': 2171.8,
 'purpose': 'BONUS 1:1',
 'quantityTraded': 500691.0,
 'recordDate': '03-DEC-14',
 'secDate': '1JAN2015',
 'securityVar': 5.02,
 'sellPrice1': 2000.0,
 'sellPrice2': 2000.5,
 'sellPrice3': 2000.55,
 'sellPrice4': 2000.6,
 'sellPrice5': 2000.85,
 'sellQuantity1': 5.0,
 'sellQuantity2': 21.0,
 'sellQuantity3': 250.0,
 'sellQuantity4': 250.0,
 'sellQuantity5': 250.0,
 'series': 'EQ',
 'symbol': 'INFY',
 'totalBuyQuantity': 78715.0,
 'totalSellQuantity': 80295.0,
 'totalTradedValue': 22914.16,
 'totalTradedVolume': 1145811.0,
 'varMargin': 7.5}
>>>
```

Note: This is a stock quote with all possible details. Since it is a dictionary you can easily chop off fields of your interest.

Warning: Always use NSE codes of stocks. Yahoo Finance or Google codes are not supported.

1.8.3 And the Index Quote

You don't always need a stock quote. At times it is just enough to know the index status. A market in general is home to many indices, in other words there are more that on index which are traded in a market.

This is true with **NSE** as well. This is how we will get quote for *CNX NIFTY* and *BANK NIFTY*

```
>>> nse.get_index_list()
['NIFTY 50 Pre Open',
 'NIFTY 50',
 'NIFTY NEXT 50',
 'NIFTY100 LIQ 15',
 'NIFTY BANK',
 'INDIA VIX',
 'NIFTY 100',
 'NIFTY 500',
 'NIFTY MIDCAP 100',
 'NIFTY MIDCAP 50',
 'NIFTY INFRA',
```

Pick one of the index codes from the above list and use it as follows. For example let's find index quote for "Nifty Bank".

```
>>> nse.get_index_quote("nifty bank") # code can be provided in upper|lower case.
{'name': 'NIFTY BANK',
 'lastPrice': 27966.65,
 'change': 204.85,
 'pChange': 0.74,
 'imgFileName': 'NIFTY_BANK_open.png'}
```

1.8.4 List of Traded Stock Codes & Names

This is very trivial in general, if you are browsing manually. But there is a way to get it programatically as well.

```
>>> all_stock_codes = nse.get_stock_codes()
{'20MICRONS': '20 Microns Limited',
 '3IINFOTECH': '3i Infotech Limited',
 '3MINDIA': '3M India Limited',
 '8KMILES': '8K Miles Software Services Limited',
 'A2ZINFRA': 'A2Z INFRA ENGINEERING LIMITED',
 'AARTIDRUGS': 'Aarti Drugs Limited',
 'AARTIIND': 'Aarti Industries Limited',
.
.
.
.
 'ZODIACLOTH': 'Zodiac Clothing Company Limited',
 'ZODJRDMKJ': 'Zodiac JRD- MKJ Limited',
 'ZUARI': 'Zuari Agro Chemicals Limited',
 'ZUARIGLOB': 'ZUARI GLOBAL LIMITED',
 'ZYDUSWELL': 'Zydus Wellness Limited',
 'ZYLOG': 'Zylog Systems Limited'}
```

Note: Output has been truncated for better legibility. This is a dictionary with more that thousand entries.

Note: After the first time use, this api returns the cached value. To avoid caching use get_stock_codes(cached=False)

1.8.5 List of Index Codes

Similar to above, there is a way to get the list of *codes* of all the traded indices. Unlike in previous section, the return type is list.

```
>>> index_codes = nse.get_index_list()
>>> pprint(index_codes)
['CNX NIFTY Pre Open',
 'CNX NIFTY',
 'CNX NIFTY JUNIOR',
 'LIX 15',
 'BANK NIFTY',
 'INDIA VIX',
 'CNX 100',
 'CNX 500',
 'CNX MIDCAP',
 'NIFTY MIDCAP 50',
 'CNX INFRA',
 'CNX REALTY',
 'CNX ENERGY',
 'CNX FMCG',
 'CNX MNC',
 'CNX PHARMA',
 'CNX PSE',
 'CNX PSU BANK',
 'CNX SERVICE',
 'CNX IT',
 'CNX SMALLCAP',
 'CNX 200',
 'CNX AUTO',
 'CNX MEDIA',
 'CNX METAL',
 'CNX DIVIDEND OPPT',
 'CNX COMMODITIES',
 'CNX CONSUMPTION',
 'CPSE INDEX',
 'CNX FINANCE',
 'NI15',
 'NIFTY TR 2X LEV',
 'NIFTY PR 2X LEV',
 'NIFTY TR 1X INV',
 'NIFTY PR 1X INV']
>>>
```

1.8.6 Advances Declines

Advances Declines is a very important feature which, in a brief snapshot, tells you the story of a trading day for the given index.

Advances Declines It containes the number of rising stocks, falling stocks and unchanged stocks in a given trading day, per index.

The following API would return the list of dictionaries containing stats for every index.

```
>>> adv_dec = nse.get_advances_declines()
>>> pprint(adv_dec)
[{'advances': 43.0, 'declines': 7.0, 'indice': 'CNX NIFTY', 'unchanged': 0.0},
 {'advances': 35.0,
  'declines': 15.0,
  'indice': 'CNX NIFTY JUNIOR',
  'unchanged': 0.0},
 {'advances': 17.0, 'declines': 3.0, 'indice': 'CNX IT', 'unchanged': 0.0},
 {'advances': 10.0, 'declines': 2.0, 'indice': 'BANK NIFTY', 'unchanged': 0.0},
 {'advances': 41.0,
  'declines': 9.0,
  'indice': 'NIFTY MIDCAP 50',
  'unchanged': 0.0},
 {'advances': 19.0, 'declines': 4.0, 'indice': 'CNX INFRA', 'unchanged': 0.0},
 {'advances': 7.0, 'declines': 3.0, 'indice': 'CNX REALTY', 'unchanged': 0.0},
 {'advances': 7.0, 'declines': 3.0, 'indice': 'CNX ENERGY', 'unchanged': 0.0},
 {'advances': 11.0, 'declines': 4.0, 'indice': 'CNX FMCG', 'unchanged': 0.0},
 {'advances': 11.0, 'declines': 4.0, 'indice': 'CNX MNC', 'unchanged': 0.0},
 {'advances': 8.0, 'declines': 2.0, 'indice': 'CNX PHARMA', 'unchanged': 0.0},
 {'advances': 12.0, 'declines': 8.0, 'indice': 'CNX PSE', 'unchanged': 0.0},
 {'advances': 8.0,
  'declines': 4.0,
  'indice': 'CNX PSU BANK',
  'unchanged': 0.0},
 {'advances': 27.0,
  'declines': 3.0,
  'indice': 'CNX SERVICE',
  'unchanged': 0.0},
 {'advances': 23.0,
  'declines': 7.0,
  'indice': 'CNX COMMODITIES',
  'unchanged': 0.0},
 {'advances': 20.0,
  'declines': 10.0,
  'indice': 'CNX CONSUMPTION',
  'unchanged': 0.0},
 {'advances': 13.0,
  'declines': 2.0,
  'indice': 'CNX FINANCE',
  'unchanged': 0.0},
 {'advances': 9.0, 'declines': 6.0, 'indice': 'CNX AUTO', 'unchanged': 0.0},
 {'advances': 30.0,
  'declines': 20.0,
  'indice': 'CNX DIVIDEND OPPT',
  'unchanged': 0.0},
 {'advances': 4.0, 'declines': 7.0, 'indice': 'CNX MEDIA', 'unchanged': 0.0},
 {'advances': 10.0, 'declines': 5.0, 'indice': 'CNX METAL', 'unchanged': 0.0},
 {'advances': 14.0, 'declines': 1.0, 'indice': 'LIX 15', 'unchanged': 0.0},
 {'advances': 6.0, 'declines': 4.0, 'indice': 'CPSE INDEX', 'unchanged': 0.0},
 {'advances': 11.0, 'declines': 4.0, 'indice': 'NI15', 'unchanged': 0.0},
 {'advances': 38.0,
  'declines': 11.0,
  'indice': 'CNX NIFTY Pre Open',
```

(continues on next page)

(continued from previous page)

```
  'unchanged': 1.0}]
>>>
```

1.8.7 Top Losers & Gainers

The following two APIs provides list of top losing and gaining stocks for the last trading session.

```
>>> top_gainers = nse.get_top_gainers()
>>>
>>> pprint(top_gainers)
[{'highPrice': 1176.95,
  'lastCorpAnnouncement': 'Annual General Meeting / Dividend - Rs 14/- Per Share',
  'lastCorpAnnouncementDate': '04-Jul-2014',
  'lowPrice': 1125.35,
  'ltp': 1171.05,
  'netPrice': 4.19,
  'openPrice': 1127.3,
  'previousPrice': 1124.0,
  'series': 'EQ',
  'symbol': 'HDFC',
  'tradedQuantity': 2019816.0,
  'turnoverInLakhs': 23428.25},
.
.
.
.
{'highPrice': 1539.0,
  'lastCorpAnnouncement': 'Annual General Meeting / Dividend - Rs 14.25/- Per Share',
  'lastCorpAnnouncementDate': '13-Aug-2014',
  'lowPrice': 1501.5,
  'ltp': 1537.0,
  'netPrice': 2.27,
  'openPrice': 1501.5,
  'previousPrice': 1502.95,
  'series': 'EQ',
  'symbol': 'LT',
  'tradedQuantity': 1291056.0,
  'turnoverInLakhs': 19709.13}]
>>>
>>> top_losers = nse.get_top_losers()
>>> pprint(top_losers)
[{'highPrice': 655.9,
  'lastCorpAnnouncement': 'Annual General Meeting/Dividend - Rs.17/- Per Share',
  'lastCorpAnnouncementDate': '05-Sep-2014',
  'lowPrice': 642.45,
  'ltp': 646.75,
  'netPrice': None,
  'openPrice': 650.9,
  'previousPrice': 654.2,
  'series': 'EQ',
  'symbol': 'BPCL',
  'tradedQuantity': 1023715.0,
  'turnoverInLakhs': 6638.08},
.
.
```

(continues on next page)

(continued from previous page)

```
.
 {'highPrice': 766.0,
  'lastCorpAnnouncement': 'Interim Dividend Rs.6/- Per Share (Purpose Revised)',
  'lastCorpAnnouncementDate': '31-Oct-2014',
  'lowPrice': 752.65,
  'ltp': 757.05,
  'netPrice': None,
  'openPrice': 757.0,
  'previousPrice': 758.45,
  'series': 'EQ',
  'symbol': 'HINDUNILVR',
  'tradedQuantity': 1207322.0,
  'turnoverInLakhs': 9174.56}]
>>>
```

1.8.8 Checking If the Code is Valid

It is expected that this library will not be used directly and the resulting data will be consumed and processed by some higher level application. Hence it is important to provide some APIs which can check the validity of the stock code or index code before fetching live quote of the same. It is recommend that in your application you always use the following two APIs before fetching a live quote.

```
>>> nse.is_valid_code('infy') # this should return True
True
>>> nse.is_valid_code('innnfy') # should return False
False
```

Similarly for index codes

```
>>> nse.is_valid_index('cnx nifty') # should return True
True
>>> nse.is_valid_index('cnxnifty') # should return False
False
```

Note: In case you perform a get_quote or get_index_quote on a code which is invalid, then the APIs return None. It doesn't raise exception as one might expect.

1.8.9 Getting Lot Sizes

In order to get the F&O lot sizes of all the stocks and indices. You can do the following.

```
>>> nse.get_fno_lot_sizes()
{'BANKNIFTY': 40,
 'NIFTYCPSE': 250,
 'FTSE100': 100,
 'NIFTYIT': 50,
 'NIFTY': 75,
 'NIFTYPSE': 200,
 'NIFTYMID50': 110,
 .
 .
```

(continues on next page)

(continued from previous page)

```
.
}
```

1.9 Release History & Change Log

0.1.0 : First release.

0.2.0 : First stable release.

1.0.0 : Support for both Python2 and Python3

1.0.1 : Few bug fixes to address format changes.

1.0.3 : Fixed encoding bug. End of offical python 2 support.

1.0.5 : Handled url format change in nseindia website

1.0.6 : Fixed the bug related to “negative change” in stock price.

1.0.7 : Added support for getting F&O lot sizes.

1.0.8 : Few build issues.

1.0.10: Handling change in backend data format.

1.0.11: Updated URLs after NSE website changes.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search