bselib

Release 0.0.1

May 12, 2021

Contents

Python library for extracting real-time data from Bombay Stock Exchange (India)

CHAPTER 1

Usage

Warning: Purpose of this library is educational. I'll highly recommend to use data service if you are looking for production level use. This library scrapes data from BSE, MarketMojo and Screener, putting too much load on their servers might get your ip address blacklisted.

1.1 Installing with pip

```
pip install bselib
```

1.2 Instantiation

```
from bselib.bse import BSE
b = BSE()
print(b)
# Output:
Driver Class for Bombay Stock Exchange
```

1.3 Getting/Verifying a script code

This method takes string/integer as an argument and returns list of stocks results. You can use pprint() for better view of the json/dict data

```
stocks = b.script('reliance')
pprint(stocks)
# Output:
pprint(stocks)
```

(continues on next page)

(continued from previous page)

```
{500111: 'Reliance Capital Ltd.',
500325: 'Reliance Industries Ltd.',
500390: 'Reliance Infrastructure Ltd.',
503162: 'Reliance Chemotex Industries Ltd.',
523445: 'Reliance Industrial Infrastructure Ltd.',
532703: 'Reliance Capital Ventures (Delisted)',
532704: 'Reliance Energy Ventures (Delisted)',
532712: 'Reliance Communications Ltd.',
533143: 'Reliance Broadcast Network Ltd. (Delisted)',
540709: 'Reliance Home Finance Ltd.'}

stocks = b.script('500325')
# Output:
pprint(stocks)
{500325: 'Reliance Industries Ltd. - 500325'}
```

1.4 Getting a stock quote

This method returns current price, detail high-low prices and other info you can pick the fields you require from the dictionary

```
 data = b.quote('500325')
#Output:
pprint(data)
{'change': '+8.40',
'daysHigh': '1537.00',
'daysLow': '1521.00',
'faceValue': '10.00',
'fiftytwo_WeekHigh': 1603.24,
'fiftytwo_WeekLow': 867.82,
'freeFloat': {'in': 'Cr', 'value': '5,13,675.84'},
'group': 'A',
'index': 'S&P BSE SENSEX',
'lastOpen': '1521.00',
'ltd': 'LTD- 02 Jun 20 | 12:19 PM',
'mktCap': {'in': 'Cr', 'value': '9,69,199.69'},
'monthHighLow': '1600.32 / 1393.65',
'pChange': '+0.55',
'previousClose': '1520.45',
'scriptCode': '500325',
'securityId': 'RELIANCE',
'stockName': 'RELIANCE INDUSTRIES LTD.',
'stockPrice': '1528.85',
'totalTradedQty': {'in': 'Lac', 'value': '1.64'},
'totalTradedValue': {'in': 'Cr', 'value': '25.12'},
'twoWeekAvgQty': {'in': 'Lac', 'value': '6.78'},
'wtdAvgPrice': '1529.46'}
```

1.5 Getting top gainers

```
top_performers = b.get_gainers()
# Output:
pprint(top_performers)
{'gainers': [{'LTP': '196.30',
              'change': '32.70',
              'pChange': '19.99',
              'scriptCode': '532638',
              'securityID': 'SHOPERSTOP'},
             {'LTP': '6.80',
              'change': '1.13',
              'pChange': '19.93',
              'scriptCode': '500106',
              'securityID': 'IFCI'},
             {'LTP': '5.96',
              'change': '0.99',
              'pChange': '19.92',
              'scriptCode': '521064',
              'securityID': 'TRIDENT'},
             {'LTP': '35.50',
              'change': '5.30',
              'pChange': '17.55',
              'scriptCode': '500101',
              'securityID': 'ARVIND'},
             {'LTP': '7.44',
              'change': '0.88',
              'pChange': '13.41',
              'scriptCode': '532822',
              'securityID': 'IDEA'}]}
```

1.6 Getting top losers

```
worst_performers = b.get_losers()
pprint(worst_performers)
# Output:
{'losers': [{'LTP': '216.25',
             'change': '-46.90',
             'pChange': '-17.82',
             'scriptCode': '540767',
             'securityID': 'NAM-INDIA'},
            {'LTP': '132.90',
             'change': '-19.00',
             'pChange': '-12.51',
             'scriptCode': '511243',
             'securityID': 'CHOLAFIN'},
             {'LTP': '314.90',
             'change': '-39.45',
             'pChange': '-11.13',
             'scriptCode': '533273',
             'securityID': 'OBEROIRLTY'},
            {'LTP': '12.66',
             'change': '-1.39',
             'pChange': '-9.89',
             'scriptCode': '532505',
             'securityID': 'UCOBANK'},
            {'LTP': '140.45',
```

(continues on next page)

(continued from previous page)

```
        'change': '-15.30',
        'pChange': '-9.82',
        'scriptCode': '532720',
        'securityID': 'M&MFIN'}]}
```

1.7 Getting Financial Statements

This function returns financial statement for latest quarter or financial year. All the data returned here is in Crores.

```
fin = b.statement(500325,stats="balancesheet")
{'CWIP': '187,022',
'borrowings': '239,843',
'fixed_assets': '403,885',
'investments': '82,862',
'other_assets': '137,504',
'other_liabilities': '277,939',
'reserves': '287,569',
'share_capital': '5,922',
'total_assets': '811,273',
'total_liabilities': '811,273'}
```

Parameters are yoy_results, quarter_results, balancesheet and cashflow

1.8 Getting Historic Financial Statements

This function returns a dictionary that can be turned to pandas dataframe. So it returns quarterly results, balance-sheets, profit-loss statements and cash-flow statements of 10+ years. All the data returned here is in Crores.

```
fin = b.historical_stats(500325,stats="cashflow")
```

Parameters are yoy_results, quarter_results, balancesheet and cashflow

1.9 Statement Analysis

Detail analysis of balance sheets, profit-loss statements, cash-flow statements, quarter results comparing to last years respective statements. Output is quite big to be displayed but it will be similar to Performance Analysis.

```
stats = b.stmt_analysis(500325,stats="yoy_results")
```

Parameters are yoy_results, quarter_results, balancesheet and cashflow.

Dir: +1 is good, -1 is bad and 0 is neutral

1.10 Getting Performance Analysis

Daily basis performance analysis of stock and analysis of company's overall performance

```
pa = b.analysis(500325)
```

1.11 Getting Financial Ratios

PE, EPS, CEPS, PB, ROE, OPM, NPM, RONW, and info like Face-value, Revenue and PAT

```
ratios = b.ratios(500325)
{'profit_ratio': {'CEPS': 64.09,
                  'EPS': 48.75,
                  'NPM': 3.35,
                  'OPM': 8.94,
                  'PE': 31.53,
                  'RONW': 40.7},
'value_ratio': {'CEPS': '64.09',
                'EPS': '48.75',
                'FaceVal': '10.00',
                'Group': 'A',
                'Grp_Index': 'A / S&P BSE SENSEX',
                'ISIN': 'INE002A01018',
                'Index': 'S&P BSE SENSEX',
                'Industry': 'Integrated Oil & Gas',
                'NPM': '-',
                'OPM': '-',
                'PB': '2.27',
                'PE': '31.19',
                'ROE': '7.28',
                'SecurityCode': '500325',
                'SecurityId': 'RELIANCE'}}
```

1.12 Getting peers comparisons

Peer comparison with info (52 wk high-low with dates,Revenue,PAT,Equity,Shareholdings) and ratios (OPM,NPM,RONW,EPS,CEPS,PE)

```
peers = b.peers(500325)
```

1.13 Getting corporate News

News related to corporate.

```
news = b.news(500325)
```

1.14 Getting Corporate's information

This function returns what corporate's business is, CEO, MD and website's link.

```
info = b.comp_profile(500325)
```

1.15 Getting Corporate's actions

Corporate actions include board meetings, declaring things like bonus, dividends, splits and rights.

```
data = b.corporate_actions(500325)
```

1.16 Getting Shareholding information and analysis

This function returns holdings information and annalysis. .. code-block:: Python

data = b.holdings(500325)

1.17 Getting Bulk deal information

```
data = b.bulk_deals(500325)
#Output:
[{"DealDate":"27/03/2020","Type":"B","Qty":"76735388","Rate":"1056.00","TO":"8103.26"}
↪,
{"DealDate":"27/03/2020","Type":"S","Qty":"86552244","Rate":"1056.61","TO":"9145.20"},
{"DealDate":"25/03/2020","Type":"B","Qty":"116081170","Rate":"949.50","TO":"11021.91"}
↪,
{"DealDate":"25/03/2020","Type":"S","Qty":"116081170","Rate":"949.50","TO":"11021.91"}
```

1.18 Getting Annual_report links

This function returns links of last 5 years of corporates annual_reports

```
data = b.annual_reports(500325)
```

1.19 Getting Credit_report links

This function returns links of last 5 years of corporates credit_reports

```
data = b.credit_reports(500325)
```