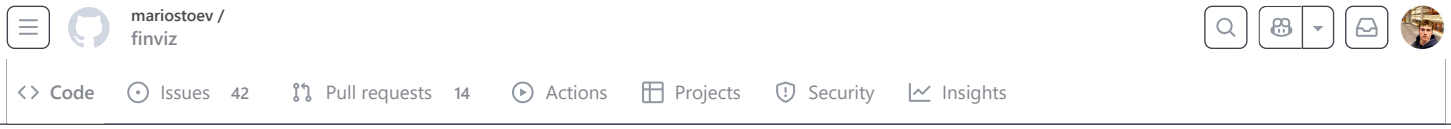


Unofficial API for finviz.com

MIT license

1.1k stars 276 forks 44 watching Branches Activity
Tags

Public repository

master 1 Branch 20 Tags Go to file Go to file Add file Code

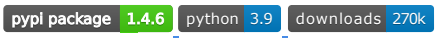
igalci Fix typo in toml file and headers mis-alignment (#184) bc02ca9 · 2 years ago

finviz	Fix typo in toml file and headers mis-alignment (...	2 years ago
.gitignore	ignore filter.json	2 years ago
LICENSE	Initial commit	8 years ago
README.rst	Update README.rst	3 years ago
example.py	Increased scraping speed, added progress bar (#...	4 years ago
pyproject.toml	Fix typo in toml file and headers mis-alignment (...	2 years ago
requirements.txt	add pyproject.toml	2 years ago
setup.py	add pyproject.toml	2 years ago

README MIT license

finviz-api

Unofficial Python API for FinViz



Downloading & Installation

```
$ pip install -U git+https://github.com/mariostoev/finviz
```

What is Finviz?

FinViz aims to make market information accessible and provides a lot of data in visual snapshots, allowing traders and investors to quickly find the stock, future or forex pair they are looking for. The site provides advanced screeners, market maps, analysis, comparative tools, and charts.

Important Information

Any quotes data displayed on finviz.com is delayed by 15 minutes for NASDAQ, and 20 minutes for NYSE and AMEX. This API should **NOT** be used for live trading, it's main purpose is financial analysis, research, and data scraping.

Using Screener

Before using the Screener class, you have to manually go to the website's screener and enter your desired settings. The URL will automatically change every time you add a new setting. After you're done the URL will look something like this:

`?v=111&s=ta_newhigh&f=cap_largeover,exch_nasd,fa_fpe_o10&o=-ticker&t=ZM` are the extra parameters provided to the screener. Those parameters are a list of key/value pairs separated with the & symbol. Some keys have a clear intent - `f=cap_largeover,exch_nasd,fa_fpe_o10` are filters, `o=-ticker` is order and `t=ZM` are tickers - yet, some are ambiguous like `v=111`, which stands for the type of table.

To make matters easier inside the code you won't refer to tables by their number tag, but instead you will use their full name (ex. `table=Performance`).

```
from finviz.screener import Screener

filters = ['exch_nasd', 'idx_sp500']  # Shows companies in NASDAQ which are in the S&P500
stock_list = Screener(filters=filters, table='Performance', order='price')  # Get the performance table and sort it by price as

# Export the screener results to .csv
stock_list.to_csv("stock.csv")

# Create a SQLite database
stock_list.to_sqlite("stock.sqlite3")

for stock in stock_list[9:19]:  # Loop through 10th - 20th stocks
    print(stock['Ticker'], stock['Price']) # Print symbol and price

# Add more filters
stock_list.add(filters=['fa_div_high'])  # Show stocks with high dividend yield
# or just stock_list(filters=['fa_div_high'])

# Print the table into the console
print(stock_list)
```

```
No.  | Ticker | Company                                    | Sector           | Industry                                  | Country       | Market Cap | P/E    | Price  | Change  | Volume
---  | ------ | ------------------------------------------ | ---------------- | ----------------------------------------- | ------------- | ---------- | ------ | ------ | ------- | ----------
1    | FTR    | Frontier Communications Corporation        | Technology       | Telecom Services - Domestic               | USA           | 410.21M    | -      | 3.51   | -7.28%  | 1,628,661
2    | NAVI   | Navient Corporation                        | Financial        | Credit Services                           | USA           | 2.92B      | -      | 10.88  | -4.39%  | 805,564
3    | NWSA   | News Corporation                           | Services         | Broadcasting - TV                         | USA           | 8.00B      | -      | 13.13  | -1.28%  | 883,095
4    | NWS    | News Corporation                           | Services         | Broadcasting - TV                         | USA           | -          | -      | 13.50  | -1.39%  | 100,996
5    | MAT    | Mattel, Inc.                               | Consumer Goods   | Toys & Games                              | USA           | 4.82B      | -      | 13.63  | -2.33%  | 1,683,980
6    | HBAN   | Huntington Bancshares Incorporated         | Financial        | Regional - Midwest Banks                  | USA           | 15.28B     | 12.70  | 13.69  | -5.46%  | 6,849,830
7    | PBCT   | People's United Financial, Inc.            | Financial        | Savings & Loans                           | USA           | 6.22B      | 13.90  | 16.22  | -4.51%  | 4,022,882
8    | FOSL   | Fossil Group, Inc.                         | Consumer Goods   | Textile - Apparel Footwear & Accessories  | USA           | 1.01B      | -      | 20.48  | -1.04%  | 692,568
9    | SYMC   | Symantec Corporation                       | Technology       | Security Software & Services              | USA           | 14.59B     | 28.52  | 22.25  | -0.49%  | 3,118,057
10   | GT     | The Goodyear Tire & Rubber Company         | Consumer Goods   | Rubber & Plastics                         | USA           | 5.61B      | 7.18   | 23.05  | -3.07%  | 1,474,602
```

Using Portfolio

```
from finviz.portfolio import Portfolio

portfolio = Portfolio('<your-email-address>', '<your-password>', '<portfolio-name>')
# Print the portfolio into the console
print(portfolio)
```

Note that, portfolio name is optional - it would assume your default portfolio (if you have one) if you exclude it. The Portfolio class can also create new portfolio from an existing `.csv` file. The `.csv` file must be in the following format:

Ticker	Transaction	Date (Opt.)	Shares	Price (Opt.)
AAPL	1	05-25-2017	34	141.28
NVDA	2		250	243.32
WMT	1	01.19.2019	45	

Note that, if any *optional* fields are left empty, the API will assign them today's data.

```
portfolio.create_portfolio('<portfolio-name>', '<path-to-csv-file>')
```

Individual stocks

```
>>> import finviz
>>> finviz.get_stock('AAPL')
{'Index': 'DJIA S&P500', 'P/E': '12.91', 'EPS (ttm)': '12.15',...
>>> finviz.get_insider('AAPL')
[{'Insider Trading': 'KONDO CHRIS', 'Relationship': 'Principal Accounting Officer', 'Date': 'Nov 19', 'Transaction':
>>> finviz.get_news('AAPL')
[('Chinas Economy Slows to the Weakest Pace Since 2009', 'https://finance.yahoo.com/news/china-economy-slows-weakest-pace-
>>>
>>> finviz.get_analyst_price_targets('AAPL')
[{'date': '2019-10-24', 'category': 'Reiterated', 'analyst': 'UBS', 'rating': 'Buy', 'price_from': 235, 'price_to': 275}, ...
```

Downloading charts

```
# Monthly, Candles, Large, No Technical Analysis
stock_list.get_charts(period='m', chart_type='c', size='l', ta='0')

# period='d' > daily
# period='w' > weekly
# period='m' > monthly

# chart_type='c' > candle
# chart_type='l' > lines

# size='m' > small
# size='l' > large

# ta='1' > display technical analysis
# ta='0' > ignore technical analysis
```

Environment Variables

Set `DISABLE_TQDM=1` in your environment to disable the progress bar.

Documentation

You can read the rest of the documentation inside the docstrings.

Contributing

You can contribute to the project by reporting bugs, suggesting enhancements, or directly by extending and writing features (see the ongoing projects).

You can also buy me a coffee!

Disclaimer

Using the library to acquire data from FinViz is against their Terms of Service and robots.txt. Use it responsibly and at your own risk. This library is built purely for educational purposes.

Releases 20

1.4.6 Latest
on Apr 8, 2023

+ 19 releases

Packages

No packages published

Contributors 25

+ 11 contributors

Languages

- Python 100.0%