

Use Finviz API to screen stocks with python



Tomer Katzav · Follow

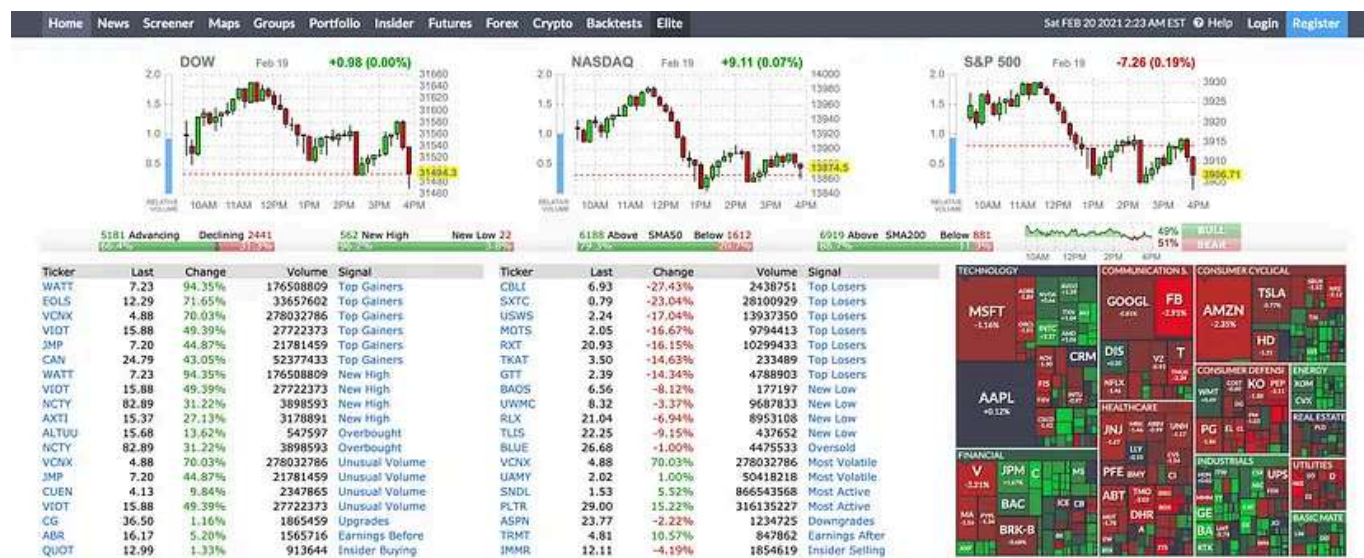
3 min read · Mar 7, 2021



85



1



Do you always feel like you're rushing to find the next best stock? Well with the Finviz API screener you can let the stocks come to you.

I'm going to show you how to connect to Finviz API and with a defined set of parameters, then you get a list of your preferable stocks and save it as CSV or in your Database.

Finviz makes the URL simple to implement in the code (just add the ticker symbol after <http://finviz.com/quote.ashx?t=>) although it comes with several limitations I won't describe here that makes it a harder task than it is and that's the reason I used [Mario Stoev unofficial API](#) as it was rather easy and intuitive to use and didn't involve many installs like in other API's out there.

So this is what you need to do to connect to the API

First, you need to install Finviz API

```
pip install finviz
```

Open in app ↗

Sign up

Sign in



Search

Write



```
from finviz.screener import Screener

from typing import List, Dict
from sklearn import logger

from stocks_finviz.get_finviz_filtered_list import
find_filters_symbols # This one you'll need to create of your own
```

You'll need to add your parameters as you would do on the Finviz website. This can be quite tricky as several parameters have the same name and their names convention change when you send them through the API.

In order to have all options available with this API, I took all the filters from [here](#) and added them as JSON to a file called `finviz_filters.py`. Without this file you cannot resolve the filter name Finviz API expects to get.

Say you want to find stocks that answer the following parameters:

1. Industry=Stocks only (ex-Funds),
2. Country=USA
3. Market Cap=-Small (under \$2bln)

All you need to do is to add them as an input parameter like the following:

```
input_filters=Industry,Stocks only (ex-Funds),Country,USA, Market  
Cap,-Small (under $2bln)
```

And with the following methods:

```
def convert_to_dict(lst):  
    lst = lst.split(',')  
    res_dct = {lst[i]: lst[i + 1] for i in range(0, len(lst), 2)}  
    return res_dct  
  
def get_filters(filters):  
    # type: (List[Dict]) -> List[str]  
    """  
    get name filters for screener  
    :param filters: list of dict  
    :return: str filters values  
    """  
  
    if isinstance(filters, str):  
        filters = [filters]  
  
    if len(filters) == 0:  
        logger.info('No filters')  
        return  
  
    filtered_results = find_filters_symbols(filters)  
  
    if filtered_results:  
        logger.info(f'The following filters are found:
```

```

{filtered_results}')
    return filtered_results
else:
    logger.warning('No filtered results from finviz screener')
    return

```

You'll end up with something like this:

```

filters = convert_to_dict(input_filters)
filters_finviz = get_filters(filters)
print(filters_finviz)

['ind_stocksonly', 'geo_usa', 'fa_eps5years_pos',
'ta_volatility_wo10', 'sh_relvol_o1', 'sh_outstanding_u20',
'sh_float_u20']

```

Which is what the Finviz screener expects to get.

From here the road is easy, you need to call the API, add the relevant filters you want and save it to a CSV or SQL DB

```

export_path = YOUR_PATH
file_name = YOUR_FILE_NAME

try:
    stock_list = Screener(filters=filters_finviz,
table='Performance', order='price')
except Exception as e:
    print(f'No stocks after screening. {e}')
    stock_list = None

# Either use finviz built in to_csv
stock_list.to_csv("stock.csv")

# Or use pandas to_excel
stock_df = pd.DataFrame(stock_list.data)

stock_df.to_excel(export_path+file_name, sheetname='filtered_stocks',
index=False) # You might need to use xlswriter

```

```
# Or create a SQLite database
stock_list.to_sqlite("stock.sqlite3")
```

If you would like the entire code for this article, you can visit this [GitHub](#). I hope this article will prove useful to your endeavors in the future. Thank you so much for reading!

Disclaimer: The material in this article is purely educational and should not be taken as professional investment advice. Invest at your own discretion.

Stocks

Screeners

Python

Stock Market



Written by Tomer Katzav

39 Followers

Correlating since 1991

Follow



More from Tomer Katzav

Split Time-Series Dataset

Sometimes the easiest part is the hardest part. Usually splitting your dataset to train, validation and test isn't a complicated task, on...



4 min read · Jul 15, 2021



11



See all from Tomer Katzav

Recommended from Medium



yfinance: 10 Ways to Get Stock Data with Python

The yfinance library offers Python users a seamless way to retrieve stock data from...

2 min read · Oct 31, 2023



How to Boost Your Returns Using the Supertrend Strategy with a...

This article provides a step-by-step guide on how to code the Supertrend indicator in...

14 min read · 6 days ago



116



55



2



Lists



Coding & Development

11 stories · 549 saves



Predictive Modeling w/ Python

20 stories · 1069 saves



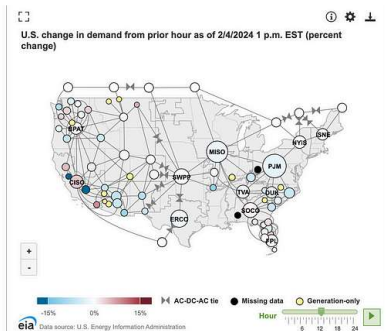
Practical Guides to Machine Learning

10 stories · 1282 saves



ChatGPT

21 stories · 554 saves



datons



Jesús López in datons

EIA API (US Energy Data) with Python

Understand the structure of the EIA API and learn how to use it with Python to automate...

4 min read · Feb 16, 2024



7



1.1K



21

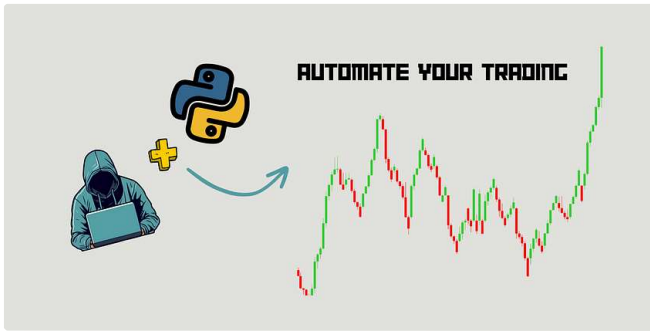


Sohan Dillikar in ILLUMINATION'S MIRROR

Here's How I Make \$50/Day By Only Placing One Trade

Here is my super simple one-trade-a-day strategy that you can use to consistently gro...

★ · 5 min read · Mar 8, 2024



 Konstantin Borimechkov

Running A Trading Bot With 15,000\$: Step-By-Step Guide On...

Automate your trades with Python & Binance!
Learn & Earn by leveraging the power of...

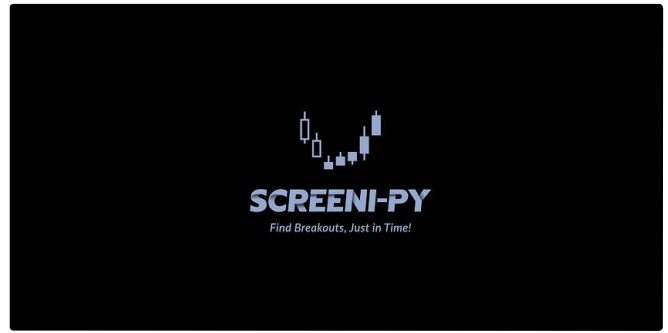
★ · 6 min read · Feb 10, 2024




436



5



 Pranjal Joshi

A Free Python-based Stock Breakout Screener with AI feature...

Screenipy is a Free, Ready-to-Use Stock Breakout Screener for NSE India powered by...

3 min read · Oct 12, 2023



81



See more recommendations