

---

# AVAPI

Aug 29, 2019



---

## Contents:

---

<b>1</b>	<b>Quickstart</b>	<b>3</b>
1.1	Installation . . . . .	3
1.2	Examples . . . . .	4
<b>2</b>	<b>Reference</b>	<b>7</b>
2.1	Getting the data . . . . .	7
2.2	Converting dictionary to pandas.DataFrame . . . . .	8
2.3	Unexpected results . . . . .	8
<b>3</b>	<b>Indices and tables</b>	<b>9</b>
	<b>Index</b>	<b>11</b>



There are other python wrappers for the alpha vantage API, such as [alphavantage](#), [alpha-vantage-downloader](#), [alpha\\_vantage](#) and [alphavantage-wrapper](#). They all have in common that they have different functions for each of the data types that Alpha Vantage offers. This may be less practical when downloading multiple data types at multiple resolutions in one session. Avapi has a single function that can download any data type from Alpha Vantage: `avapi.get_data()` takes `**kwargs` as input and outputs a dictionary, unless `datatype="csv"` is given, in which case a csv file is saved. If the data type is time series, `avapi.to_df()` converts it to a Pandas data frame. The intention is a simple and flexible way to get data from Alpha Vantage into python.



# CHAPTER 1

---

## Quickstart

---

Alpha Vantage docs

### Is a python wrapper for the Alpha Vantage API really necessary?

Not really. If you are content with downloading csv files and you don't want to or can't include a wrapper in your project or workflow, getting data from Alpha Vantage is fairly straight forward. Referencing the [docs](#), you could do something like:

```
import requests

url = 'https://www.alphavantage.co/query?'
url += 'function=TIME_SERIES_INTRADAY'
url += '&symbol=MSFT'
url += '&interval=5min'
url += '&apikey=demo'
url += '&datatype=csv'

save_to = 'MSFT_5min_intraday.csv'

r = requests.get(url)
content = r.content

with open(save_to, 'wb') as file:
    file.write(content)
```

## 1.1 Installation

```
pip install avapi
```

### 1.1.1 Before use

Follow instructions on the [Alpha vantage](#) website to get a free API-key. The [docs](#) provide all necessary info for à la carte downloading of historical data and indicators. Please checkout Alpha Vantage [support](#) and read through the [Frequently Asked Questions](#).

## 1.2 Examples

The “demo” api-key only works for these specific examples and a limited number of times. If you modify the code you also need to provide your own api-key.

### 1.2.1 Single download

```
import avapi as aa
import pandas as pd

data = aa.get_data(function='VWAP', symbol='MSFT',
                  interval='15min', apikey='demo')

df = aa.to_df(data)
df.head()
```

### 1.2.2 Multiple files download

There are limits for free accounts: up to 5 API requests per minute and 500 requests per day. In loops, if you don't incorporate `time.sleep()`, you might get errors.

```
import avapi as aa
import pandas as pd
from time import sleep

data_calls = [

    {
        'function': 'TIME_SERIES_DAILY',
        'symbol': 'MSFT',
        'apikey': 'demo',
        'datatype': 'csv',
    },
    {
        'function': 'TIME_SERIES_INTRADAY',
        'symbol': 'MSFT',
        'interval': '5min',
        'apikey': 'demo',
        'datatype': 'csv',
    }
]

for i in range(len(data_calls)):
    save_to = str(i) + '.csv'
```

(continues on next page)



(continued from previous page)

```
data = aa.get_data(save_to=save_to, **data_calls[i])
sleep(15)
```

### 1.2.3 Unexpected results

If you get unexpected results, you may check the latest, original response from the Alpha Vantage server in the following way:

```
response = aa.response()
print(response)
```



AVAPI is a simple python wrapper for the Alpha Vantage API. AVAPI has one module: `avapi.data`. It handles getting and manipulating data from Alpha Vantage.

## 2.1 Getting the data

If `csv=True`, a csv file will be saved in current working directory. Otherwise, `avapi.data.get_data()` will return a dictionary of Alpha Vantage data.

`avapi.get_data(save_to=None, **kwargs)`

Downloads a json file from Alpha Vantage.

### Parameters

- **save\_to** (str or None) – Default None. Where to save csv file if datatype="csv" is provided.
- **\*\*kwargs** – See below

### Keyword Arguments

- **function** (str) – Any of the Alpha Vantage function types. Such as "TIME\_SERIES\_INTRADAY". See their [documentation](#)
- **symbol** (str) – Company ticker symbol, such as "GOOGL".
- **interval** (str) – "1min", "5min", "15min", "30min" or "60min". Also "daily", "weekly", "monthly"

- **outputsize (str)** – "compact" (default) or "full"
- **datatype (str)** – "json" (default) or "csv"
- **apikey (str)** – You need to get a free API key from [Alpha Vantage](#)

The above list is not exhaustive. Please see [Alpha Vantage docs](#) for complete listing and what function requires which keyword arguments.

**Returns** If datatype is not set to "csv", a dictionary is returned

**Return type** dict [str, float]

## 2.2 Converting dictionary to pandas.DataFrame

When `avapi.get_data(csv=False)` downloads data, it converts the Alpha Vantage server response from json to python dict. If the data type is a time series, `avapi.to_df()` can then convert it to a Pandas data frame.

`avapi.to_df(dic)`

Converts data dictionary, downloaded from Alpha Vantage, to pandas dataframes.

**Parameters** **dic** (dict [str, float]) – Python dictionary of Alpha Vantage time series data

**Returns** Returns the converted dictionary as a Pandas data frame

**Return type** pandas.DataFrame()

## 2.3 Unexpected results

When you get unexpected results, you might want to check out the latest raw response from the Alpha Vantage server.

`avapi.response()`

Opens and reads last response from Alpha Vantage server.

**Returns** Content of response.

**Return type** str

## CHAPTER 3

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



## G

`get_data()` (*in module avapi*), 7

## R

`response()` (*in module avapi*), 8

## T

`to_df()` (*in module avapi*), 8