Alpha Vantage API Documentation

Our [stock APIs ©](https://www.alphavantage.co/) are grouped into eight categories: (1) Core Time Series Stock Data APIs, (2) US Options Data APIs, (3) Alpha Intelligence™, (4) Fundamental Data, (5) Physical and Digital/Crypto Currencies (e.g., Bitcoin), (6) Commodities, (7) Economic Indicators, and (8) Technical Indicators - also outlined [here](https://www.alphavantage.co/best_stock_market_api_review/). Examples in this documentation are for demo purposes. [Claim your free API key](https://www.alphavantage.co/support/#api-key) today to explore our full API offerings!

[Time Series Stock Data APIs](https://www.alphavantage.co/documentation/#time-series-data)

This suite of APIs provide global equity data in 4 different temporal resolutions: (1) daily, (2) weekly, (3) monthly, and (4) intraday, with 20+ years of historical depth. A lightweight ticker quote endpoint and several utility functions such as ticker search and market open/closure status are also included for your convenience.

[TIME\_SERIES\_INTRADAY Trending](https://www.alphavantage.co/documentation/#intraday)

This API returns current and 20+ years of historical intraday OHLCV time series of the equity specified, covering extended trading hours where applicable (e.g., 4:00am to 8:00pm Eastern Time for the US market). You can query both raw (as-traded) and split/dividend-adjusted intraday data from this endpoint. The OHLCV data is sometimes called "candles" in finance literature.

**API Parameters**

**❚ Required: function**

The time series of your choice. In this case, function=TIME\_SERIES\_INTRADAY

**❚ Required: symbol**

The name of the equity of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min

❚ Optional: adjusted

By default, adjusted=true and the output time series is adjusted by historical split and dividend events. Set adjusted=false to query raw (as-traded) intraday values.

❚ Optional: extended\_hours

By default, extended\_hours=true and the output time series will include both the regular trading hours and the extended trading hours (4:00am to 8:00pm Eastern Time for the US market). Set extended\_hours=false to query regular trading hours (9:30am to 4:00pm US Eastern Time) only.

❚ Optional: month

By default, this parameter is not set and the API will return intraday data for the most recent days of trading. You can use the month parameter (in YYYY-MM format) to query a specific month in history. For example, month=2009-01. Any month in the last 20+ years since 2000-01 (January 2000) is supported.

❚ Optional: outputsize

By default, outputsize=compact. Strings compact and full are accepted with the following specifications: compact returns only the latest 100 data points in the intraday time series; full returns trailing 30 days of the most recent intraday data if the month parameter (see above) is not specified, or the full intraday data for a specific month in history if the month parameter is specified. The "compact" option is recommended if you would like to reduce the data size of each API call.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the intraday time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

*The API will return the most recent 100 intraday OHLCV bars by default when the outputsize parameter is not set*

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_INTRADAY&**symbol**=IBM&**interval**=5min&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_INTRADAY&symbol=IBM&interval=5min&apikey=demo)

*Query the most recent full 30 days of intraday data by setting outputsize=full*

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_INTRADAY&**symbol**=IBM&**interval**=5min&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_INTRADAY&symbol=IBM&interval=5min&outputsize=full&apikey=demo)

*Query intraday data for a given month in history (e.g., 2009-01). Any month in the last 20+ years (since 2000-01) is supported*

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_INTRADAY&**symbol**=IBM&**interval**=5min&**month**=2009-01&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_INTRADAY&symbol=IBM&interval=5min&month=2009-01&outputsize=full&apikey=demo)

Downloadable CSV file:

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_INTRADAY&**symbol**=IBM&**interval**=5min&**apikey**=demo&**datatype**=csv](https://www.alphavantage.co/query?function=TIME_SERIES_INTRADAY&symbol=IBM&interval=5min&apikey=demo&datatype=csv)

💡 Tip: the intraday data (including 20+ years of historical data) is updated at the end of each trading day for all users by default. If you would like to access realtime or 15-minute delayed intraday data, please subscribe to a [premium membership plan](https://www.alphavantage.co/premium/) for your personal use. For commercial use, please [contact sales](mailto:support@alphavantage.co).

\* Realtime and 15-minute delayed US market data is regulated by the stock exchanges, FINRA, and the SEC. [Learn more](https://www.alphavantage.co/realtime_data_policy/) about the key market data policies you need to know as a data consumer.

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=TIME\_SERIES\_INTRADAY&symbol=IBM&interval=5min&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[TIME\_SERIES\_DAILY](https://www.alphavantage.co/documentation/#daily)

This API returns raw (as-traded) daily time series (date, daily open, daily high, daily low, daily close, daily volume) of the global equity specified, covering 20+ years of historical data. The OHLCV data is sometimes called "candles" in finance literature. If you are also interested in split/dividend-adjusted data, please use the [Daily Adjusted API](https://www.alphavantage.co/documentation/#dailyadj), which covers adjusted close values and historical split and dividend events.

**API Parameters**

**❚ Required: function**

The time series of your choice. In this case, function=TIME\_SERIES\_DAILY

**❚ Required: symbol**

The name of the equity of your choice. For example: symbol=IBM

❚ Optional: outputsize

By default, outputsize=compact. Strings compact and full are accepted with the following specifications: compact returns only the latest 100 data points; full returns the full-length time series of 20+ years of historical data. The "compact" option is recommended if you would like to reduce the data size of each API call.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

*Sample ticker traded in the United States*

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY&**symbol**=IBM&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol=IBM&apikey=demo)

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY&**symbol**=IBM&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol=IBM&outputsize=full&apikey=demo)

*Sample ticker traded in UK - London Stock Exchange*

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY&**symbol**=TSCO.LON&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol=TSCO.LON&outputsize=full&apikey=demo)

*Sample ticker traded in Canada - Toronto Stock Exchange*

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY&**symbol**=SHOP.TRT&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol=SHOP.TRT&outputsize=full&apikey=demo)

*Sample ticker traded in Canada - Toronto Venture Exchange*

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY&**symbol**=GPV.TRV&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol=GPV.TRV&outputsize=full&apikey=demo)

*Sample ticker traded in Germany - XETRA*

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY&**symbol**=MBG.DEX&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol=MBG.DEX&outputsize=full&apikey=demo)

*Sample ticker traded in India - BSE*

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY&**symbol**=RELIANCE.BSE&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol=RELIANCE.BSE&outputsize=full&apikey=demo)

*Sample ticker traded in China - Shanghai Stock Exchange*

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY&**symbol**=600104.SHH&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol=600104.SHH&outputsize=full&apikey=demo)

*Sample ticker traded in China - Shenzhen Stock Exchange*

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY&**symbol**=000002.SHZ&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol=000002.SHZ&outputsize=full&apikey=demo)

The above is just a small sample of the 100,000+ symbols we support. Please refer to our [Search Endpoint](https://www.alphavantage.co/documentation/#symbolsearch) to look up any supported global stock, ETF, or mutual fund symbols of your interest.

Downloadable CSV file:

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY&**symbol**=IBM&**apikey**=demo&**datatype**=csv](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol=IBM&apikey=demo&datatype=csv)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=TIME\_SERIES\_DAILY&symbol=IBM&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[TIME\_SERIES\_DAILY\_ADJUSTED Trending Premium](https://www.alphavantage.co/documentation/#dailyadj)

This API returns raw (as-traded) daily open/high/low/close/volume values, adjusted close values, and historical split/dividend events of the global equity specified, covering 20+ years of historical data. The OHLCV data is sometimes called "candles" in finance literature.

**API Parameters**

**❚ Required: function**

The time series of your choice. In this case, function=TIME\_SERIES\_DAILY\_ADJUSTED

**❚ Required: symbol**

The name of the equity of your choice. For example: symbol=IBM

❚ Optional: outputsize

By default, outputsize=compact. Strings compact and full are accepted with the following specifications: compact returns only the latest 100 data points; full returns the full-length time series of 20+ years of historical data. The "compact" option is recommended if you would like to reduce the data size of each API call.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

*Sample ticker traded in the United States*

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY\_ADJUSTED&**symbol**=IBM&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY_ADJUSTED&symbol=IBM&apikey=demo)

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY\_ADJUSTED&**symbol**=IBM&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY_ADJUSTED&symbol=IBM&outputsize=full&apikey=demo)

*Sample ticker traded in UK - London Stock Exchange*

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY\_ADJUSTED&**symbol**=TSCO.LON&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY_ADJUSTED&symbol=TSCO.LON&outputsize=full&apikey=demo)

*Sample ticker traded in Canada - Toronto Stock Exchange*

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY\_ADJUSTED&**symbol**=SHOP.TRT&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY_ADJUSTED&symbol=SHOP.TRT&outputsize=full&apikey=demo)

*Sample ticker traded in Canada - Toronto Venture Exchange*

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY\_ADJUSTED&**symbol**=GPV.TRV&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY_ADJUSTED&symbol=GPV.TRV&outputsize=full&apikey=demo)

*Sample ticker traded in Germany - XETRA*

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY\_ADJUSTED&**symbol**=MBG.DEX&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY_ADJUSTED&symbol=MBG.DEX&outputsize=full&apikey=demo)

*Sample ticker traded in India - BSE*

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY\_ADJUSTED&**symbol**=RELIANCE.BSE&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY_ADJUSTED&symbol=RELIANCE.BSE&outputsize=full&apikey=demo)

*Sample ticker traded in China - Shanghai Stock Exchange*

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY\_ADJUSTED&**symbol**=600104.SHH&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY_ADJUSTED&symbol=600104.SHH&outputsize=full&apikey=demo)

*Sample ticker traded in China - Shenzhen Stock Exchange*

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY\_ADJUSTED&**symbol**=000002.SHZ&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY_ADJUSTED&symbol=000002.SHZ&outputsize=full&apikey=demo)

The above is just a small sample of the 100,000+ symbols we support. Please refer to our [Search Endpoint](https://www.alphavantage.co/documentation/#symbolsearch) to look up any supported global stock, ETF, or mutual fund symbols of your interest.

Downloadable CSV file:

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_DAILY\_ADJUSTED&**symbol**=IBM&**apikey**=demo&**datatype**=csv](https://www.alphavantage.co/query?function=TIME_SERIES_DAILY_ADJUSTED&symbol=IBM&apikey=demo&datatype=csv)

💡 Tip: this is a premium API function. Subscribe to a [premium membership plan](https://www.alphavantage.co/premium/) to instantly unlock all premium APIs.

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=TIME\_SERIES\_DAILY\_ADJUSTED&symbol=IBM&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[TIME\_SERIES\_WEEKLY](https://www.alphavantage.co/documentation/#weekly)

This API returns weekly time series (last trading day of each week, weekly open, weekly high, weekly low, weekly close, weekly volume) of the global equity specified, covering 20+ years of historical data.

**API Parameters**

**❚ Required: function**

The time series of your choice. In this case, function=TIME\_SERIES\_WEEKLY

**❚ Required: symbol**

The name of the equity of your choice. For example: symbol=IBM

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the weekly time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_WEEKLY&**symbol**=IBM&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_WEEKLY&symbol=IBM&apikey=demo)

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_WEEKLY&**symbol**=TSCO.LON&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_WEEKLY&symbol=TSCO.LON&apikey=demo)

Downloadable CSV file:

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_WEEKLY&**symbol**=IBM&**apikey**=demo&**datatype**=csv](https://www.alphavantage.co/query?function=TIME_SERIES_WEEKLY&symbol=IBM&apikey=demo&datatype=csv)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=TIME\_SERIES\_WEEKLY&symbol=IBM&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[TIME\_SERIES\_WEEKLY\_ADJUSTED](https://www.alphavantage.co/documentation/#weeklyadj)

This API returns weekly adjusted time series (last trading day of each week, weekly open, weekly high, weekly low, weekly close, weekly adjusted close, weekly volume, weekly dividend) of the global equity specified, covering 20+ years of historical data.

**API Parameters**

**❚ Required: function**

The time series of your choice. In this case, function=TIME\_SERIES\_WEEKLY\_ADJUSTED

**❚ Required: symbol**

The name of the equity of your choice. For example: symbol=IBM

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the weekly time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_WEEKLY\_ADJUSTED&**symbol**=IBM&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_WEEKLY_ADJUSTED&symbol=IBM&apikey=demo)

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_WEEKLY\_ADJUSTED&**symbol**=TSCO.LON&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_WEEKLY_ADJUSTED&symbol=TSCO.LON&apikey=demo)

Downloadable CSV file:

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_WEEKLY\_ADJUSTED&**symbol**=IBM&**apikey**=demo&**datatype**=csv](https://www.alphavantage.co/query?function=TIME_SERIES_WEEKLY_ADJUSTED&symbol=IBM&apikey=demo&datatype=csv)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=TIME\_SERIES\_WEEKLY\_ADJUSTED&symbol=IBM&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[TIME\_SERIES\_MONTHLY](https://www.alphavantage.co/documentation/#monthly)

This API returns monthly time series (last trading day of each month, monthly open, monthly high, monthly low, monthly close, monthly volume) of the global equity specified, covering 20+ years of historical data.

**API Parameters**

**❚ Required: function**

The time series of your choice. In this case, function=TIME\_SERIES\_MONTHLY

**❚ Required: symbol**

The name of the equity of your choice. For example: symbol=IBM

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the monthly time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_MONTHLY&**symbol**=IBM&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_MONTHLY&symbol=IBM&apikey=demo)

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_MONTHLY&**symbol**=TSCO.LON&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_MONTHLY&symbol=TSCO.LON&apikey=demo)

Downloadable CSV file:

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_MONTHLY&**symbol**=IBM&**apikey**=demo&**datatype**=csv](https://www.alphavantage.co/query?function=TIME_SERIES_MONTHLY&symbol=IBM&apikey=demo&datatype=csv)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=TIME\_SERIES\_MONTHLY&symbol=IBM&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[TIME\_SERIES\_MONTHLY\_ADJUSTED](https://www.alphavantage.co/documentation/#monthlyadj)

This API returns monthly adjusted time series (last trading day of each month, monthly open, monthly high, monthly low, monthly close, monthly adjusted close, monthly volume, monthly dividend) of the equity specified, covering 20+ years of historical data.

**API Parameters**

**❚ Required: function**

The time series of your choice. In this case, function=TIME\_SERIES\_MONTHLY\_ADJUSTED

**❚ Required: symbol**

The name of the equity of your choice. For example: symbol=IBM

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the monthly time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_MONTHLY\_ADJUSTED&**symbol**=IBM&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_MONTHLY_ADJUSTED&symbol=IBM&apikey=demo)

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_MONTHLY\_ADJUSTED&**symbol**=TSCO.LON&**apikey**=demo](https://www.alphavantage.co/query?function=TIME_SERIES_MONTHLY_ADJUSTED&symbol=TSCO.LON&apikey=demo)

Downloadable CSV file:

[https://www.alphavantage.co/query?**function**=TIME\_SERIES\_MONTHLY\_ADJUSTED&**symbol**=IBM&**apikey**=demo&**datatype**=csv](https://www.alphavantage.co/query?function=TIME_SERIES_MONTHLY_ADJUSTED&symbol=IBM&apikey=demo&datatype=csv)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=TIME\_SERIES\_MONTHLY\_ADJUSTED&symbol=IBM&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Quote Endpoint Trending](https://www.alphavantage.co/documentation/#latestprice)

A lightweight alternative to the time series APIs, this service returns the latest price and volume information for a ticker of your choice.

**API Parameters**

**❚ Required: function**

The API function of your choice.

**❚ Required: symbol**

The symbol of the global ticker of your choice. For example: symbol=IBM.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the quote data in JSON format; csv returns the quote data as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=GLOBAL\_QUOTE&**symbol**=IBM&**apikey**=demo](https://www.alphavantage.co/query?function=GLOBAL_QUOTE&symbol=IBM&apikey=demo)

[https://www.alphavantage.co/query?**function**=GLOBAL\_QUOTE&**symbol**=300135.SHZ&**apikey**=demo](https://www.alphavantage.co/query?function=GLOBAL_QUOTE&symbol=300135.SHZ&apikey=demo)

Downloadable CSV file:

[https://www.alphavantage.co/query?**function**=GLOBAL\_QUOTE&**symbol**=IBM&**apikey**=demo&**datatype**=csv](https://www.alphavantage.co/query?function=GLOBAL_QUOTE&symbol=IBM&apikey=demo&datatype=csv)

💡 Tip: by default, the quote endpoint is updated at the end of each trading day for all users. If you would like to access realtime or 15-minute delayed stock quote data for the US market, please subscribe to a [premium membership plan](https://www.alphavantage.co/premium/) for your personal use. For commercial use, please [contact sales](mailto:support@alphavantage.co).

\* Realtime and 15-minute delayed US market data is regulated by the stock exchanges, FINRA, and the SEC. [Learn more](https://www.alphavantage.co/realtime_data_policy/) about the key market data policies you need to know as a data consumer.

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=GLOBAL\_QUOTE&symbol=IBM&apikey=demo'

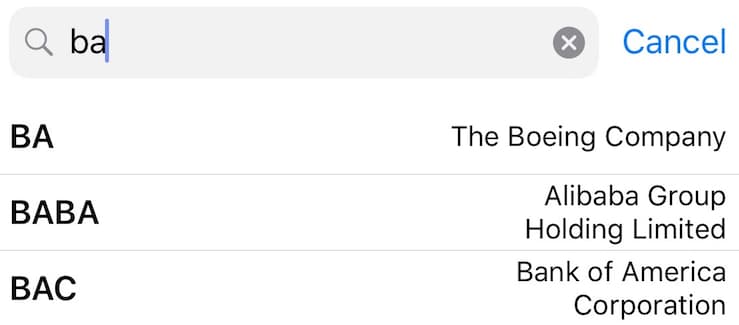
r = requests.get(url)

data = r.json()

print(data)

[Search Endpoint Utility](https://www.alphavantage.co/documentation/#symbolsearch)

Looking for some specific symbols or companies? Trying to build an auto-complete search box similar to the one below?



We've got you covered! The Search Endpoint returns the best-matching symbols and market information based on keywords of your choice. The search results also contain match scores that provide you with the full flexibility to develop your own search and filtering logic.

**API Parameters**

**❚ Required: function**

The API function of your choice. In this case, function=SYMBOL\_SEARCH

**❚ Required: keywords**

A text string of your choice. For example: keywords=microsoft.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the search results in JSON format; csv returns the search results as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=SYMBOL\_SEARCH&**keywords**=tesco&**apikey**=demo](https://www.alphavantage.co/query?function=SYMBOL_SEARCH&keywords=tesco&apikey=demo)

[https://www.alphavantage.co/query?**function**=SYMBOL\_SEARCH&**keywords**=tencent&**apikey**=demo](https://www.alphavantage.co/query?function=SYMBOL_SEARCH&keywords=tencent&apikey=demo)

[https://www.alphavantage.co/query?**function**=SYMBOL\_SEARCH&**keywords**=BA&**apikey**=demo](https://www.alphavantage.co/query?function=SYMBOL_SEARCH&keywords=BA&apikey=demo)

[https://www.alphavantage.co/query?**function**=SYMBOL\_SEARCH&**keywords**=SAIC&**apikey**=demo](https://www.alphavantage.co/query?function=SYMBOL_SEARCH&keywords=SAIC&apikey=demo)

Downloadable CSV file:

[https://www.alphavantage.co/query?**function**=SYMBOL\_SEARCH&**keywords**=BA&**apikey**=demo&**datatype**=csv](https://www.alphavantage.co/query?function=SYMBOL_SEARCH&keywords=BA&apikey=demo&datatype=csv)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=SYMBOL\_SEARCH&keywords=tesco&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Global Market Open & Close Status Utility](https://www.alphavantage.co/documentation/#market-status)

This endpoint returns the current market status (open vs. closed) of major trading venues for equities, forex, and cryptocurrencies around the world.

**API Parameters**

**❚ Required: function**

The API function of your choice. In this case, function=MARKET\_STATUS

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=MARKET\_STATUS&**apikey**=demo](https://www.alphavantage.co/query?function=MARKET_STATUS&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=MARKET\_STATUS&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Options Data APIs](https://www.alphavantage.co/documentation/#options)

This suite of APIs provide realtime and historical US options data, spanning 15+ years of history with full market/volume coverage.

[Realtime Options Trending Premium](https://www.alphavantage.co/documentation/#realtime-options)

This API returns realtime US options data with full market coverage. Option chains are sorted by expiration dates in chronological order. Within the same expiration date, contracts are sorted by strike prices from low to high.

**API Parameters**

**❚ Required: function**

The time series of your choice. In this case, function=REALTIME\_OPTIONS

**❚ Required: symbol**

The name of the equity of your choice. For example: symbol=IBM

❚ Optional: contract

The US options contract ID you would like to specify. By default, the contract parameter is not set and the entire option chain for a given symbol will be returned.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the options data in JSON format; csv returns the data as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=REALTIME\_OPTIONS&**symbol**=IBM&**apikey**=demo](https://www.alphavantage.co/query?function=REALTIME_OPTIONS&symbol=IBM&apikey=demo)

💡 Tip: this is a premium API function. Subscribe to either the 600 requests per minute or the 1200 requests per minute [premium membership plan](https://www.alphavantage.co/premium/) to unlock realtime options data.

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=REALTIME\_OPTIONS&symbol=IBM&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Historical Options Trending](https://www.alphavantage.co/documentation/#historical-options)

This API returns the full historical options chain for a specific symbol on a specific date, covering 15+ years of history. Implied volatility (IV) and common Greeks (e.g., delta, gamma, theta, vega, rho) are also returned. Option chains are sorted by expiration dates in chronological order. Within the same expiration date, contracts are sorted by strike prices from low to high.

**API Parameters**

**❚ Required: function**

The time series of your choice. In this case, function=HISTORICAL\_OPTIONS

**❚ Required: symbol**

The name of the equity of your choice. For example: symbol=IBM

❚ Optional: date

By default, the date parameter is not set and the API will return data for the previous trading session. Any date later than 2008-01-01 is accepted. For example, date=2017-11-15.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the options data in JSON format; csv returns the data as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

*When the date paramter is not set, data from the previous trading session is returned*

[https://www.alphavantage.co/query?**function**=HISTORICAL\_OPTIONS&**symbol**=IBM&**apikey**=demo](https://www.alphavantage.co/query?function=HISTORICAL_OPTIONS&symbol=IBM&apikey=demo)

*Specify a date to retrieve options data for any trading day in the past 15+ years (since 2018-01-01)*

[https://www.alphavantage.co/query?**function**=HISTORICAL\_OPTIONS&**symbol**=IBM&**date**=2017-11-15&**apikey**=demo](https://www.alphavantage.co/query?function=HISTORICAL_OPTIONS&symbol=IBM&date=2017-11-15&apikey=demo)

Downloadable CSV file:

[https://www.alphavantage.co/query?**function**=HISTORICAL\_OPTIONS&**symbol**=IBM&**date**=2017-11-15&**apikey**=demo&**datatype**=csv](https://www.alphavantage.co/query?function=HISTORICAL_OPTIONS&symbol=IBM&date=2017-11-15&apikey=demo&datatype=csv)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=HISTORICAL\_OPTIONS&symbol=IBM&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Alpha Intelligence™](https://www.alphavantage.co/documentation/#intelligence)

The APIs in this section contain advanced market intelligence built with our decades of expertise in AI, machine learning, and quantitative finance. We hope these highly differentiated alternative datasets can help turbocharge your trading strategy, market research, and financial software application to the next level.

[Market News & Sentiment Trending](https://www.alphavantage.co/documentation/#news-sentiment)

Looking for market news data to train your LLM models or to augment your trading strategy? You have just found it. This API returns live and historical market news & sentiment data from a large & growing selection of premier news outlets around the world, covering stocks, cryptocurrencies, forex, and a wide range of topics such as fiscal policy, mergers & acquisitions, IPOs, etc. This API, combined with our core stock API, fundamental data, and technical indicator APIs, can provide you with a 360-degree view of the financial market and the broader economy.

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=NEWS\_SENTIMENT

**❚ Optional: tickers**

The stock/crypto/forex symbols of your choice. For example: tickers=IBM will filter for articles that mention the IBM ticker; tickers=COIN,CRYPTO:BTC,FOREX:USD will filter for articles that simultaneously mention Coinbase (COIN), Bitcoin (CRYPTO:BTC), and US Dollar (FOREX:USD) in their content.

**❚ Optional: topics**

The news topics of your choice. For example: topics=technology will filter for articles that write about the technology sector; topics=technology,ipo will filter for articles that simultaneously cover technology and IPO in their content. Below is the full list of supported topics:

* Blockchain: blockchain
* Earnings: earnings
* IPO: ipo
* Mergers & Acquisitions: mergers\_and\_acquisitions
* Financial Markets: financial\_markets
* Economy - Fiscal Policy (e.g., tax reform, government spending): economy\_fiscal
* Economy - Monetary Policy (e.g., interest rates, inflation): economy\_monetary
* Economy - Macro/Overall: economy\_macro
* Energy & Transportation: energy\_transportation
* Finance: finance
* Life Sciences: life\_sciences
* Manufacturing: manufacturing
* Real Estate & Construction: real\_estate
* Retail & Wholesale: retail\_wholesale
* Technology: technology

**❚ Optional: time\_from and time\_to**

The time range of the news articles you are targeting, in YYYYMMDDTHHMM format. For example: time\_from=20220410T0130. If time\_from is specified but time\_to is missing, the API will return articles published between the time\_from value and the current time.

**❚ Optional: sort**

By default, sort=LATEST and the API will return the latest articles first. You can also set sort=EARLIEST or sort=RELEVANCE based on your use case.

**❚ Optional: limit**

By default, limit=50 and the API will return up to 50 matching results. You can also set limit=1000 to output up to 1000 results.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

*Querying news articles that mention the AAPL ticker.*

[https://www.alphavantage.co/query?**function**=NEWS\_SENTIMENT&**tickers**=AAPL&**apikey**=demo](https://www.alphavantage.co/query?function=NEWS_SENTIMENT&tickers=AAPL&apikey=demo)

*Querying news articles that simultaneously mention the Coinbase stock (COIN), Bitcoin (CRYPTO:BTC), and US Dollar (FOREX:USD) and are published on or after 2022-04-10, 1:30am UTC.*

[https://www.alphavantage.co/query?**function**=NEWS\_SENTIMENT&tickers=COIN,CRYPTO:BTC,FOREX:USD&time\_from=20220410T0130&limit=1000&apikey=demo](https://www.alphavantage.co/query?function=NEWS_SENTIMENT&tickers=COIN,CRYPTO:BTC,FOREX:USD&time_from=20220410T0130&limit=1000&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=NEWS\_SENTIMENT&tickers=AAPL&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Top Gainers, Losers, and Most Actively Traded Tickers (US Market)](https://www.alphavantage.co/documentation/#gainer-loser)

This endpoint returns the top 20 gainers, losers, and the most active traded tickers in the US market.

**API Parameters**

**❚ Required: function**

The API function of your choice. In this case, function=TOP\_GAINERS\_LOSERS

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=TOP\_GAINERS\_LOSERS&**apikey**=demo](https://www.alphavantage.co/query?function=TOP_GAINERS_LOSERS&apikey=demo)

💡 Tip: By default, the top gainers, losers, and the most active traded ticker information is updated at the end of each trading day for all users. If you would like to access realtime or 15-minute delayed data, please subscribe to a [premium membership plan](https://www.alphavantage.co/premium/) for your personal use. For commercial use, please [contact sales](mailto:support@alphavantage.co).

\* Realtime and 15-minute delayed US market data is regulated by the stock exchanges, FINRA, and the SEC. [Learn more](https://www.alphavantage.co/realtime_data_policy/) about the key market data policies you need to know as a data consumer.

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=TOP\_GAINERS\_LOSERS&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Advanced Analytics (Fixed Window) Trending](https://www.alphavantage.co/documentation/#analytics-fixed-window)

This endpoint returns a rich set of advanced analytics metrics (e.g., total return, variance, auto-correlation, etc.) for a given time series over a fixed temporal window.

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=ANALYTICS\_FIXED\_WINDOW

**❚ Required: SYMBOLS**

A list of symbols for the calculation. It can be a comma separated list of symbols as a string. Free API keys can specify up to 5 symbols per API request. Premium API keys can specify up to 50 symbols per API request.

**❚ Required: RANGE**

This is the date range for the series being requested. By default, the date range is the full set of data for the equity history. This can be further modified by the LIMIT variable.

RANGE can take certain text values as inputs. They are:

* full
* {N}day
* {N}week
* {N}month
* {N}year

For intraday time series, the following RANGE values are also accepted:

* {N}minute
* {N}hour

Aside from the “full” value which represents the entire time series, the other values specify an interval to return the series for as measured backwards from the current date/time.

To specify start & end dates for your analytics calcuation, simply add two RANGE parameters in your API request. For example: RANGE=2023-07-01&RANGE=2023-08-31 or RANGE=2020-12-01T00:04:00&RANGE=2020-12-06T23:59:59 with minute-level precision for intraday analytics. If the end date is missing, the end date is assumed to be the last trading date. In addition, you can request a full month of data by using YYYY-MM format like 2020-12. One day of intraday data can be requested by using YYYY-MM-DD format like 2020-12-06

**❚ Optional: OHLC**

This allows you to choose which open, high, low, or close field the calculation will be performed on. By default, OHLC=close. Valid values for these fields are open, high, low, close.

**❚ Required: INTERVAL**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, DAILY, WEEKLY, MONTHLY.

**❚ Required: CALCULATIONS**

A comma separated list of the analytics metrics you would like to calculate:

* MIN: The minimum return (largest negative or smallest positive) for all values in the series
* MAX: The maximum return for all values in the series
* MEAN: The mean of all returns in the series
* MEDIAN: The median of all returns in the series
* CUMULATIVE\_RETURN: The total return from the beginning to the end of the series range
* VARIANCE: The population variance of returns in the series range. Optionally, you can use VARIANCE(annualized=True)to normalized the output to an annual value. By default, the variance is not annualized.
* STDDEV: The population standard deviation of returns in the series range for each symbol. Optionally, you can use STDDEV(annualized=True)to normalized the output to an annual value. By default, the standard deviation is not annualized.
* MAX\_DRAWDOWN: Largest peak to trough interval for each symbol in the series range
* HISTOGRAM: For each symbol, place the observed total returns in bins. By default, bins=10. Use HISTOGRAM(bins=20) to specify a custom bin value (e.g., 20).
* AUTOCORRELATION: For each symbol place, calculate the autocorrelation for the given lag (e.g., the lag in neighboring points for the autocorrelation calculation). By default, lag=1. Use AUTOCORRELATION(lag=2) to specify a custom lag value (e.g., 2).
* COVARIANCE: Returns a covariance matrix for the input symbols. Optionally, you can use COVARIANCE(annualized=True)to normalized the output to an annual value. By default, the covariance is not annualized.
* CORRELATION: Returns a correlation matrix for the input symbols, using the PEARSON method as default. You can also specify the KENDALL or SPEARMAN method through CORRELATION(method=KENDALL) or CORRELATION(method=SPEARMAN), respectively.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

*For AAPL, MSFT, and IBM, calculate the mean & standard deviation of their returns based on daily close prices between 2023-07-01 and 2023-08-31, along with a correlation matrix among the three tickers.*

<https://www.alphavantage.co/query?function=ANALYTICS_FIXED_WINDOW&SYMBOLS=AAPL,MSFT,IBM&RANGE=2023-07-01&RANGE=2023-08-31&INTERVAL=DAILY&OHLC=close&CALCULATIONS=MEAN,STDDEV,CORRELATION&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://alphavantageapi.co/timeseries/analytics?SYMBOLS=AAPL,MSFT,IBM&RANGE=2023-07-01&RANGE=2023-08-31&INTERVAL=DAILY&OHLC=close&CALCULATIONS=MEAN,STDDEV,CORRELATION&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Advanced Analytics (Sliding Window) Trending](https://www.alphavantage.co/documentation/#analytics-sliding-window)

This endpoint returns a rich set of advanced analytics metrics (e.g., total return, variance, auto-correlation, etc.) for a given time series over sliding time windows. For example, we can calculate a moving variance over 5 years with a window of 100 points to see how the variance changes over time.

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=ANALYTICS\_SLIDING\_WINDOW

**❚ Required: SYMBOLS**

A list of symbols for the calculation. It can be a comma separated list of symbols as a string. Free API keys can specify up to 5 symbols per API request. Premium API keys can specify up to 50 symbols per API request.

**❚ Required: RANGE**

This is the date range for the series being requested. By default, the date range is the full set of data for the equity history. This can be further modified by the LIMIT variable.

RANGE can take certain text values as inputs. They are:

* full
* {N}day
* {N}week
* {N}month
* {N}year

For intraday time series, the following RANGE values are also accepted:

* {N}minute
* {N}hour

Aside from the “full” value which represents the entire time series, the other values specify an interval to return the series for as measured backwards from the current date/time.

To specify start & end dates for your analytics calcuation, simply add two RANGE parameters in your API request. For example: RANGE=2023-07-01&RANGE=2023-08-31 or RANGE=2020-12-01T00:04:00&RANGE=2020-12-06T23:59:59 with minute-level precision for intraday analytics. If the end date is missing, the end date is assumed to be the last trading date. In addition, you can request a full month of data by using YYYY-MM format like 2020-12. One day of intraday data can be requested by using YYYY-MM-DD format like 2020-12-06

**❚ Optional: OHLC**

This allows you to choose which open, high, low, or close field the calculation will be performed on. By default, OHLC=close. Valid values for these fields are open, high, low, close.

**❚ Required: INTERVAL**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, DAILY, WEEKLY, MONTHLY.

**❚ Required: WINDOW\_SIZE**

An integer representing the size of the moving window. A hard lower boundary of 10 has been set though it is recommended to make this window larger to make sure the running calculations are statistically significant.

**❚ Required: CALCULATIONS**

A comma separated list of the analytics metrics you would like to calculate. Free API keys can specify 1 metric to be calculated per API request. Premium API keys can specify multiple metrics to be calculated simultaneously per API request.

* MEAN: The mean of all returns in the series
* MEDIAN: The median of all returns in the series
* CUMULATIVE\_RETURN: The total return from the beginning to the end of the series range
* VARIANCE: The population variance of returns in the series range. Optionally, you can use VARIANCE(annualized=True)to normalized the output to an annual value. By default, the variance is not annualized.
* STDDEV: The population standard deviation of returns in the series range for each symbol. Optionally, you can use STDDEV(annualized=True)to normalized the output to an annual value. By default, the standard deviation is not annualized.
* COVARIANCE: Returns a covariance matrix for the input symbols. Optionally, you can use COVARIANCE(annualized=True)to normalized the output to an annual value. By default, the covariance is not annualized.
* CORRELATION: Returns a correlation matrix for the input symbols, using the PEARSON method as default. You can also specify the KENDALL or SPEARMAN method through CORRELATION(method=KENDALL) or CORRELATION(method=SPEARMAN), respectively.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

*For AAPL and IBM, calculate the running mean & annualized standard deviation of their returns based on daily close prices in the trailing 2 months, with a sliding window size of 20.*

<https://www.alphavantage.co/query?function=ANALYTICS_SLIDING_WINDOW&SYMBOLS=AAPL,IBM&RANGE=2month&INTERVAL=DAILY&OHLC=close&WINDOW_SIZE=20&CALCULATIONS=MEAN,STDDEV(annualized=True)&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://alphavantageapi.co/timeseries/running\_analytics?SYMBOLS=AAPL,IBM&RANGE=2month&INTERVAL=DAILY&OHLC=close&WINDOW\_SIZE=20&CALCULATIONS=MEAN,STDDEV(annualized=True)&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Fundamental Data](https://www.alphavantage.co/documentation/#fundamentals)

We offer the following set of fundamental data APIs in various temporal dimensions covering key financial metrics, income statements, balance sheets, cash flow, and other fundamental data points.

[Company Overview](https://www.alphavantage.co/documentation/#company-overview)

This API returns the company information, financial ratios, and other key metrics for the equity specified. Data is generally refreshed on the same day a company reports its latest earnings and financials.

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=OVERVIEW

**❚ Required: symbol**

The symbol of the ticker of your choice. For example: symbol=IBM.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=OVERVIEW&**symbol**=IBM&**apikey**=demo](https://www.alphavantage.co/query?function=OVERVIEW&symbol=IBM&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=OVERVIEW&symbol=IBM&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Corporate Action - Dividends](https://www.alphavantage.co/documentation/#dividends)

This API returns historical and future (declared) dividend distributions.

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=DIVIDENDS

**❚ Required: symbol**

The symbol of the ticker of your choice. For example: symbol=IBM.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=DIVIDENDS&**symbol**=IBM&**apikey**=demo](https://www.alphavantage.co/query?function=DIVIDENDS&symbol=IBM&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=DIVIDENDS&symbol=IBM&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Corporate Action - Splits](https://www.alphavantage.co/documentation/#splits)

This API returns historical split events.

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=SPLITS

**❚ Required: symbol**

The symbol of the ticker of your choice. For example: symbol=IBM.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=SPLITS&**symbol**=IBM&**apikey**=demo](https://www.alphavantage.co/query?function=SPLITS&symbol=IBM&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=SPLITS&symbol=IBM&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[INCOME\_STATEMENT Trending](https://www.alphavantage.co/documentation/#income-statement)

This API returns the annual and quarterly income statements for the company of interest, with normalized fields [mapped to GAAP and IFRS taxonomies](https://documentation.alphavantage.co/FundamentalDataDocs/index.html) of the SEC. Data is generally refreshed on the same day a company reports its latest earnings and financials.

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=INCOME\_STATEMENT

**❚ Required: symbol**

The symbol of the ticker of your choice. For example: symbol=IBM.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example - annual & quarterly income statements for IBM (click for JSON output)**

[https://www.alphavantage.co/query?**function**=INCOME\_STATEMENT&**symbol**=IBM&**apikey**=demo](https://www.alphavantage.co/query?function=INCOME_STATEMENT&symbol=IBM&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=INCOME\_STATEMENT&symbol=IBM&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[BALANCE\_SHEET Trending](https://www.alphavantage.co/documentation/#balance-sheet)

This API returns the annual and quarterly balance sheets for the company of interest, with normalized fields [mapped to GAAP and IFRS taxonomies](https://documentation.alphavantage.co/FundamentalDataDocs/index.html) of the SEC. Data is generally refreshed on the same day a company reports its latest earnings and financials.

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=BALANCE\_SHEET

**❚ Required: symbol**

The symbol of the ticker of your choice. For example: symbol=IBM.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example - annual & quarterly balance sheets for IBM (click for JSON output)**

[https://www.alphavantage.co/query?**function**=BALANCE\_SHEET&**symbol**=IBM&**apikey**=demo](https://www.alphavantage.co/query?function=BALANCE_SHEET&symbol=IBM&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=BALANCE\_SHEET&symbol=IBM&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[CASH\_FLOW Trending](https://www.alphavantage.co/documentation/#cash-flow)

This API returns the annual and quarterly cash flow for the company of interest, with normalized fields [mapped to GAAP and IFRS taxonomies](https://documentation.alphavantage.co/FundamentalDataDocs/index.html) of the SEC. Data is generally refreshed on the same day a company reports its latest earnings and financials.

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=CASH\_FLOW

**❚ Required: symbol**

The symbol of the ticker of your choice. For example: symbol=IBM.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example - annual & quarterly cash flows for IBM (click for JSON output)**

[https://www.alphavantage.co/query?**function**=CASH\_FLOW&**symbol**=IBM&**apikey**=demo](https://www.alphavantage.co/query?function=CASH_FLOW&symbol=IBM&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=CASH\_FLOW&symbol=IBM&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Earnings](https://www.alphavantage.co/documentation/#earnings)

This API returns the annual and quarterly earnings (EPS) for the company of interest. Quarterly data also includes analyst estimates and surprise metrics.

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=EARNINGS

**❚ Required: symbol**

The symbol of the ticker of your choice. For example: symbol=IBM.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=EARNINGS&**symbol**=IBM&**apikey**=demo](https://www.alphavantage.co/query?function=EARNINGS&symbol=IBM&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=EARNINGS&symbol=IBM&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Listing & Delisting Status](https://www.alphavantage.co/documentation/#listing-status)

This API returns a list of active or delisted US stocks and ETFs, either as of the latest trading day or at a specific time in history. The endpoint is positioned to facilitate equity research on asset lifecycle and survivorship.

**API Parameters**

**❚ Required: function**

The API function of your choice. In this case, function=LISTING\_STATUS

❚ Optional: date

If no date is set, the API endpoint will return a list of active or delisted symbols as of the latest trading day. If a date is set, the API endpoint will "travel back" in time and return a list of active or delisted symbols on that particular date in history. Any YYYY-MM-DD date later than 2010-01-01 is supported. For example, date=2013-08-03

❚ Optional: state

By default, state=active and the API will return a list of actively traded stocks and ETFs. Set state=delisted to query a list of delisted assets.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples**

To ensure optimal API response time, this endpoint uses the CSV format which is more memory-efficient than JSON.

Querying all active stocks and ETFs as of the latest trading day:

[https://www.alphavantage.co/query?**function**=LISTING\_STATUS&**apikey**=demo](https://www.alphavantage.co/query?function=LISTING_STATUS&apikey=demo)

Querying all delisted stocks and ETFs as of 2014-07-10:

[https://www.alphavantage.co/query?**function**=LISTING\_STATUS&**date**=2014-07-10&**state**=delisted&**apikey**=demo](https://www.alphavantage.co/query?function=LISTING_STATUS&date=2014-07-10&state=delisted&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** csv

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

CSV\_URL = 'https://www.alphavantage.co/query?function=LISTING\_STATUS&apikey=demo'

**with** requests.Session() **as** s:

download = s.get(CSV\_URL)

decoded\_content = download.content.decode('utf-8')

cr = csv.reader(decoded\_content.splitlines(), delimiter=',')

my\_list = list(cr)

**for** row **in** my\_list:

print(row)

[Earnings Calendar](https://www.alphavantage.co/documentation/#earnings-calendar)

This API returns a list of company earnings expected in the next 3, 6, or 12 months.

**API Parameters**

**❚ Required: function**

The API function of your choice. In this case, function=EARNINGS\_CALENDAR

❚ Optional: symbol

By default, no symbol will be set for this API. When no symbol is set, the API endpoint will return the full list of company earnings scheduled. If a symbol is set, the API endpoint will return the expected earnings for that specific symbol. For example, symbol=IBM

❚ Optional: horizon

By default, horizon=3month and the API will return a list of expected company earnings in the next 3 months. You may set horizon=6month or horizon=12month to query the earnings scheduled for the next 6 months or 12 months, respectively.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples**

To ensure optimal API response time, this endpoint uses the CSV format which is more memory-efficient than JSON.

Querying all the company earnings expected in the next 3 months:

[https://www.alphavantage.co/query?**function**=EARNINGS\_CALENDAR&**horizon**=3month&**apikey**=demo](https://www.alphavantage.co/query?function=EARNINGS_CALENDAR&horizon=3month&apikey=demo)

Querying all the earnings events for IBM in the next 12 months:

[https://www.alphavantage.co/query?**function**=EARNINGS\_CALENDAR&**symbol**=IBM&**horizon**=12month&**apikey**=demo](https://www.alphavantage.co/query?function=EARNINGS_CALENDAR&symbol=IBM&horizon=12month&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** csv

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

CSV\_URL = 'https://www.alphavantage.co/query?function=EARNINGS\_CALENDAR&horizon=3month&apikey=demo'

**with** requests.Session() **as** s:

download = s.get(CSV\_URL)

decoded\_content = download.content.decode('utf-8')

cr = csv.reader(decoded\_content.splitlines(), delimiter=',')

my\_list = list(cr)

**for** row **in** my\_list:

print(row)

[IPO Calendar](https://www.alphavantage.co/documentation/#ipo-calendar)

This API returns a list of IPOs expected in the next 3 months.

**API Parameters**

**❚ Required: function**

The API function of your choice. In this case, function=IPO\_CALENDAR

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples**

To ensure optimal API response time, this endpoint uses the CSV format which is more memory-efficient than JSON.

Querying all the company earnings expected in the next 3 months:

[https://www.alphavantage.co/query?**function**=IPO\_CALENDAR&**apikey**=demo](https://www.alphavantage.co/query?function=IPO_CALENDAR&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** csv

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

CSV\_URL = 'https://www.alphavantage.co/query?function=IPO\_CALENDAR&apikey=demo'

**with** requests.Session() **as** s:

download = s.get(CSV\_URL)

decoded\_content = download.content.decode('utf-8')

cr = csv.reader(decoded\_content.splitlines(), delimiter=',')

my\_list = list(cr)

**for** row **in** my\_list:

print(row)

[Foreign Exchange Rates (FX)](https://www.alphavantage.co/documentation/#fx)

APIs under this section provide a wide range of data feed for realtime and historical forex (FX) rates.

[CURRENCY\_EXCHANGE\_RATE Trending](https://www.alphavantage.co/documentation/#currency-exchange)

This API returns the realtime exchange rate for a pair of digital currency (e.g., Bitcoin) and physical currency (e.g., USD).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=CURRENCY\_EXCHANGE\_RATE

**❚ Required: from\_currency**

The currency you would like to get the exchange rate for. It can either be a [physical currency](https://www.alphavantage.co/physical_currency_list/) or [digital/crypto currency](https://www.alphavantage.co/digital_currency_list/). For example: from\_currency=USD or from\_currency=BTC.

**❚ Required: to\_currency**

The destination currency for the exchange rate. It can either be a [physical currency](https://www.alphavantage.co/physical_currency_list/) or [digital/crypto currency](https://www.alphavantage.co/digital_currency_list/). For example: to\_currency=USD or to\_currency=BTC.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

US Dollar to Japanese Yen:

<https://www.alphavantage.co/query?function=CURRENCY_EXCHANGE_RATE&from_currency=USD&to_currency=JPY&apikey=demo>

Bitcoin to Euro:

<https://www.alphavantage.co/query?function=CURRENCY_EXCHANGE_RATE&from_currency=BTC&to_currency=EUR&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=CURRENCY\_EXCHANGE\_RATE&from\_currency=USD&to\_currency=JPY&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[FX\_INTRADAY Premium Trending](https://www.alphavantage.co/documentation/#fx-intraday)

This API returns intraday time series (timestamp, open, high, low, close) of the FX currency pair specified, updated realtime.

**API Parameters**

**❚ Required: function**

The time series of your choice. In this case, function=FX\_INTRADAY

**❚ Required: from\_symbol**

A three-letter symbol from the [forex currency list](https://www.alphavantage.co/physical_currency_list/). For example: from\_symbol=EUR

**❚ Required: to\_symbol**

A three-letter symbol from the [forex currency list](https://www.alphavantage.co/physical_currency_list/). For example: to\_symbol=USD

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min

❚ Optional: outputsize

By default, outputsize=compact. Strings compact and full are accepted with the following specifications: compact returns only the latest 100 data points in the intraday time series; full returns the full-length intraday time series. The "compact" option is recommended if you would like to reduce the data size of each API call.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the intraday time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=FX\_INTRADAY&**from\_symbol**=EUR&**to\_symbol**=USD&**interval**=5min&**apikey**=demo](https://www.alphavantage.co/query?function=FX_INTRADAY&from_symbol=EUR&to_symbol=USD&interval=5min&apikey=demo)

[https://www.alphavantage.co/query?**function**=FX\_INTRADAY&**from\_symbol**=EUR&**to\_symbol**=USD&**interval**=5min&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=FX_INTRADAY&from_symbol=EUR&to_symbol=USD&interval=5min&outputsize=full&apikey=demo)

Downloadable CSV file:

[https://www.alphavantage.co/query?**function**=FX\_INTRADAY&**from\_symbol**=EUR&**to\_symbol**=USD&**interval**=5min&**apikey**=demo&**datatype**=csv](https://www.alphavantage.co/query?function=FX_INTRADAY&from_symbol=EUR&to_symbol=USD&interval=5min&apikey=demo&datatype=csv)

💡 Tip: this is a premium API function. Subscribe to a [premium membership plan](https://www.alphavantage.co/premium/) to instantly unlock all premium APIs.

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=FX\_INTRADAY&from\_symbol=EUR&to\_symbol=USD&interval=5min&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[FX\_DAILY](https://www.alphavantage.co/documentation/#fx-daily)

This API returns the daily time series (timestamp, open, high, low, close) of the FX currency pair specified, updated realtime.

**API Parameters**

**❚ Required: function**

The time series of your choice. In this case, function=FX\_DAILY

**❚ Required: from\_symbol**

A three-letter symbol from the [forex currency list](https://www.alphavantage.co/physical_currency_list/). For example: from\_symbol=EUR

**❚ Required: to\_symbol**

A three-letter symbol from the [forex currency list](https://www.alphavantage.co/physical_currency_list/). For example: to\_symbol=USD

❚ Optional: outputsize

By default, outputsize=compact. Strings compact and full are accepted with the following specifications: compact returns only the latest 100 data points in the daily time series; full returns the full-length daily time series. The "compact" option is recommended if you would like to reduce the data size of each API call.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=FX\_DAILY&**from\_symbol**=EUR&**to\_symbol**=USD&**apikey**=demo](https://www.alphavantage.co/query?function=FX_DAILY&from_symbol=EUR&to_symbol=USD&apikey=demo)

[https://www.alphavantage.co/query?**function**=FX\_DAILY&**from\_symbol**=EUR&**to\_symbol**=USD&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=FX_DAILY&from_symbol=EUR&to_symbol=USD&outputsize=full&apikey=demo)

Downloadable CSV file:

[https://www.alphavantage.co/query?**function**=FX\_DAILY&**from\_symbol**=EUR&**to\_symbol**=USD&**apikey**=demo&**datatype**=csv](https://www.alphavantage.co/query?function=FX_DAILY&from_symbol=EUR&to_symbol=USD&apikey=demo&datatype=csv)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=FX\_DAILY&from\_symbol=EUR&to\_symbol=USD&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[FX\_WEEKLY](https://www.alphavantage.co/documentation/#fx-weekly)

This API returns the weekly time series (timestamp, open, high, low, close) of the FX currency pair specified, updated realtime.

The latest data point is the price information for the week (or partial week) containing the current trading day, updated realtime.

**API Parameters**

**❚ Required: function**

The time series of your choice. In this case, function=FX\_WEEKLY

**❚ Required: from\_symbol**

A three-letter symbol from the [forex currency list](https://www.alphavantage.co/physical_currency_list/). For example: from\_symbol=EUR

**❚ Required: to\_symbol**

A three-letter symbol from the [forex currency list](https://www.alphavantage.co/physical_currency_list/). For example: to\_symbol=USD

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the weekly time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=FX\_WEEKLY&**from\_symbol**=EUR&**to\_symbol**=USD&**apikey**=demo](https://www.alphavantage.co/query?function=FX_WEEKLY&from_symbol=EUR&to_symbol=USD&apikey=demo)

Downloadable CSV file:

[https://www.alphavantage.co/query?**function**=FX\_WEEKLY&**from\_symbol**=EUR&**to\_symbol**=USD&**apikey**=demo&**datatype**=csv](https://www.alphavantage.co/query?function=FX_WEEKLY&from_symbol=EUR&to_symbol=USD&apikey=demo&datatype=csv)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=FX\_WEEKLY&from\_symbol=EUR&to\_symbol=USD&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[FX\_MONTHLY](https://www.alphavantage.co/documentation/#fx-monthly)

This API returns the monthly time series (timestamp, open, high, low, close) of the FX currency pair specified, updated realtime.

The latest data point is the prices information for the month (or partial month) containing the current trading day, updated realtime.

**API Parameters**

**❚ Required: function**

The time series of your choice. In this case, function=FX\_MONTHLY

**❚ Required: from\_symbol**

A three-letter symbol from the [forex currency list](https://www.alphavantage.co/physical_currency_list/). For example: from\_symbol=EUR

**❚ Required: to\_symbol**

A three-letter symbol from the [forex currency list](https://www.alphavantage.co/physical_currency_list/). For example: to\_symbol=USD

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the monthly time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=FX\_MONTHLY&**from\_symbol**=EUR&**to\_symbol**=USD&**apikey**=demo](https://www.alphavantage.co/query?function=FX_MONTHLY&from_symbol=EUR&to_symbol=USD&apikey=demo)

Downloadable CSV file:

[https://www.alphavantage.co/query?**function**=FX\_MONTHLY&**from\_symbol**=EUR&**to\_symbol**=USD&**apikey**=demo&**datatype**=csv](https://www.alphavantage.co/query?function=FX_MONTHLY&from_symbol=EUR&to_symbol=USD&apikey=demo&datatype=csv)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=FX\_MONTHLY&from\_symbol=EUR&to\_symbol=USD&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Digital & Crypto Currencies](https://www.alphavantage.co/documentation/#digital-currency)

APIs under this section provide a wide range of data feed for digital and crypto currencies such as Bitcoin.

[CURRENCY\_EXCHANGE\_RATE Trending](https://www.alphavantage.co/documentation/#crypto-exchange)

This API returns the realtime exchange rate for any pair of digital currency (e.g., Bitcoin) or physical currency (e.g., USD).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=CURRENCY\_EXCHANGE\_RATE

**❚ Required: from\_currency**

The currency you would like to get the exchange rate for. It can either be a [physical currency](https://www.alphavantage.co/physical_currency_list/) or [digital/crypto currency](https://www.alphavantage.co/digital_currency_list/). For example: from\_currency=USD or from\_currency=BTC.

**❚ Required: to\_currency**

The destination currency for the exchange rate. It can either be a [physical currency](https://www.alphavantage.co/physical_currency_list/) or [digital/crypto currency](https://www.alphavantage.co/digital_currency_list/). For example: to\_currency=USD or to\_currency=BTC.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

Bitcoin to Euro:

<https://www.alphavantage.co/query?function=CURRENCY_EXCHANGE_RATE&from_currency=BTC&to_currency=EUR&apikey=demo>

US Dollar to Japanese Yen:

<https://www.alphavantage.co/query?function=CURRENCY_EXCHANGE_RATE&from_currency=USD&to_currency=JPY&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=CURRENCY\_EXCHANGE\_RATE&from\_currency=BTC&to\_currency=EUR&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[CRYPTO\_INTRADAY Trending Premium](https://www.alphavantage.co/documentation/#crypto-intraday)

This API returns intraday time series (timestamp, open, high, low, close, volume) of the cryptocurrency specified, updated realtime.

**API Parameters**

**❚ Required: function**

The time series of your choice. In this case, function=CRYPTO\_INTRADAY

**❚ Required: symbol**

The digital/crypto currency of your choice. It can be any of the currencies in the [digital currency list](https://www.alphavantage.co/digital_currency_list/). For example: symbol=ETH.

**❚ Required: market**

The exchange market of your choice. It can be any of the market in the [market list](https://www.alphavantage.co/physical_currency_list/). For example: market=USD.

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min

❚ Optional: outputsize

By default, outputsize=compact. Strings compact and full are accepted with the following specifications: compact returns only the latest 100 data points in the intraday time series; full returns the full-length intraday time series. The "compact" option is recommended if you would like to reduce the data size of each API call.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the intraday time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=CRYPTO\_INTRADAY&**symbol**=ETH&**market**=USD&**interval**=5min&**apikey**=demo](https://www.alphavantage.co/query?function=CRYPTO_INTRADAY&symbol=ETH&market=USD&interval=5min&apikey=demo)

[https://www.alphavantage.co/query?**function**=CRYPTO\_INTRADAY&**symbol**=ETH&**market**=USD&**interval**=5min&**outputsize**=full&**apikey**=demo](https://www.alphavantage.co/query?function=CRYPTO_INTRADAY&symbol=ETH&market=USD&interval=5min&outputsize=full&apikey=demo)

Downloadable CSV file:

[https://www.alphavantage.co/query?**function**=CRYPTO\_INTRADAY&**symbol**=ETH&**market**=USD&**interval**=5min&**apikey**=demo&**datatype**=csv](https://www.alphavantage.co/query?function=CRYPTO_INTRADAY&symbol=ETH&market=USD&interval=5min&apikey=demo&datatype=csv)

💡 Tip: this is a premium API function. Subscribe to a [premium membership plan](https://www.alphavantage.co/premium/) to instantly unlock all premium APIs.

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=CRYPTO\_INTRADAY&symbol=ETH&market=USD&interval=5min&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[DIGITAL\_CURRENCY\_DAILY](https://www.alphavantage.co/documentation/#currency-daily)

This API returns the daily historical time series for a digital currency (e.g., BTC) traded on a specific market (e.g., EUR/Euro), refreshed daily at midnight (UTC). Prices and volumes are quoted in both the market-specific currency and USD.

**API Parameters**

**❚ Required: function**

The time series of your choice. In this case, function=DIGITAL\_CURRENCY\_DAILY

**❚ Required: symbol**

The digital/crypto currency of your choice. It can be any of the currencies in the [digital currency list](https://www.alphavantage.co/digital_currency_list/). For example: symbol=BTC.

**❚ Required: market**

The exchange market of your choice. It can be any of the market in the [market list](https://www.alphavantage.co/physical_currency_list/). For example: market=EUR.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=DIGITAL_CURRENCY_DAILY&symbol=BTC&market=EUR&apikey=demo>

Downloadable CSV file:

<https://www.alphavantage.co/query?function=DIGITAL_CURRENCY_DAILY&symbol=BTC&market=EUR&apikey=demo&datatype=csv>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=DIGITAL\_CURRENCY\_DAILY&symbol=BTC&market=EUR&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[DIGITAL\_CURRENCY\_WEEKLY Trending](https://www.alphavantage.co/documentation/#currency-weekly)

This API returns the weekly historical time series for a digital currency (e.g., BTC) traded on a specific market (e.g., EUR/Euro), refreshed daily at midnight (UTC). Prices and volumes are quoted in both the market-specific currency and USD.

**API Parameters**

**❚ Required: function**

The time series of your choice. In this case, function=DIGITAL\_CURRENCY\_WEEKLY

**❚ Required: symbol**

The digital/crypto currency of your choice. It can be any of the currencies in the [digital currency list](https://www.alphavantage.co/digital_currency_list/). For example: symbol=BTC.

**❚ Required: market**

The exchange market of your choice. It can be any of the market in the [market list](https://www.alphavantage.co/physical_currency_list/). For example: market=EUR.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=DIGITAL_CURRENCY_WEEKLY&symbol=BTC&market=EUR&apikey=demo>

Downloadable CSV file:

<https://www.alphavantage.co/query?function=DIGITAL_CURRENCY_WEEKLY&symbol=BTC&market=EUR&apikey=demo&datatype=csv>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=DIGITAL\_CURRENCY\_WEEKLY&symbol=BTC&market=EUR&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[DIGITAL\_CURRENCY\_MONTHLY Trending](https://www.alphavantage.co/documentation/#currency-monthly)

This API returns the monthly historical time series for a digital currency (e.g., BTC) traded on a specific market (e.g., EUR/Euro), refreshed daily at midnight (UTC). Prices and volumes are quoted in both the market-specific currency and USD.

**API Parameters**

**❚ Required: function**

The time series of your choice. In this case, function=DIGITAL\_CURRENCY\_MONTHLY

**❚ Required: symbol**

The digital/crypto currency of your choice. It can be any of the currencies in the [digital currency list](https://www.alphavantage.co/digital_currency_list/). For example: symbol=BTC.

**❚ Required: market**

The exchange market of your choice. It can be any of the market in the [market list](https://www.alphavantage.co/physical_currency_list/). For example: market=EUR.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=DIGITAL_CURRENCY_MONTHLY&symbol=BTC&market=EUR&apikey=demo>

Downloadable CSV file:

<https://www.alphavantage.co/query?function=DIGITAL_CURRENCY_MONTHLY&symbol=BTC&market=EUR&apikey=demo&datatype=csv>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=DIGITAL\_CURRENCY\_MONTHLY&symbol=BTC&market=EUR&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Commodities](https://www.alphavantage.co/documentation/#commodities)

APIs under this section provide historical data for major commodities such as crude oil, natural gas, copper, wheat, etc., spanning across various temporal horizons (daily, weekly, monthly, quarterly, etc.)

[Crude Oil Prices: West Texas Intermediate (WTI) Trending](https://www.alphavantage.co/documentation/#wti)

This API returns the West Texas Intermediate (WTI) crude oil prices in daily, weekly, and monthly horizons.

Source: U.S. Energy Information Administration, Crude Oil Prices: West Texas Intermediate (WTI) - Cushing, Oklahoma, retrieved from FRED, Federal Reserve Bank of St. Louis. This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=WTI

❚ Optional: interval

By default, interval=monthly. Strings daily, weekly, and monthly are accepted.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=WTI&interval=monthly&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=WTI&interval=monthly&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Crude Oil Prices (Brent) Trending](https://www.alphavantage.co/documentation/#brent)

This API returns the Brent (Europe) crude oil prices in daily, weekly, and monthly horizons.

Source: U.S. Energy Information Administration, Crude Oil Prices: Brent - Europe, retrieved from FRED, Federal Reserve Bank of St. Louis. This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=BRENT

❚ Optional: interval

By default, interval=monthly. Strings daily, weekly, and monthly are accepted.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=BRENT&interval=monthly&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=BRENT&interval=monthly&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Natural Gas](https://www.alphavantage.co/documentation/#natural-gas)

This API returns the Henry Hub natural gas spot prices in daily, weekly, and monthly horizons.

Source: U.S. Energy Information Administration, Henry Hub Natural Gas Spot Price, retrieved from FRED, Federal Reserve Bank of St. Louis. This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=NATURAL\_GAS

❚ Optional: interval

By default, interval=monthly. Strings daily, weekly, and monthly are accepted.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=NATURAL_GAS&interval=monthly&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=NATURAL\_GAS&interval=monthly&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Global Price of Copper Trending](https://www.alphavantage.co/documentation/#copper)

This API returns the global price of copper in monthly, quarterly, and annual horizons.

Source: International Monetary Fund ([IMF Terms of Use](https://www.imf.org/external/terms.htm)), Global price of Copper, retrieved from FRED, Federal Reserve Bank of St. Louis. This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=COPPER

❚ Optional: interval

By default, interval=monthly. Strings monthly, quarterly, and annual are accepted.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=COPPER&interval=monthly&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=COPPER&interval=monthly&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Global Price of Aluminum](https://www.alphavantage.co/documentation/#aluminum)

This API returns the global price of aluminum in monthly, quarterly, and annual horizons.

Source: International Monetary Fund ([IMF Terms of Use](https://www.imf.org/external/terms.htm)), Global price of Aluminum, retrieved from FRED, Federal Reserve Bank of St. Louis. This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=ALUMINUM

❚ Optional: interval

By default, interval=monthly. Strings monthly, quarterly, and annual are accepted.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=ALUMINUM&interval=monthly&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=ALUMINUM&interval=monthly&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Global Price of Wheat](https://www.alphavantage.co/documentation/#wheat)

This API returns the global price of wheat in monthly, quarterly, and annual horizons.

Source: International Monetary Fund ([IMF Terms of Use](https://www.imf.org/external/terms.htm)), Global price of Wheat, retrieved from FRED, Federal Reserve Bank of St. Louis. This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=WHEAT

❚ Optional: interval

By default, interval=monthly. Strings monthly, quarterly, and annual are accepted.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=WHEAT&interval=monthly&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=WHEAT&interval=monthly&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Global Price of Corn](https://www.alphavantage.co/documentation/#corn)

This API returns the global price of corn in monthly, quarterly, and annual horizons.

Source: International Monetary Fund ([IMF Terms of Use](https://www.imf.org/external/terms.htm)), Global price of Corn, retrieved from FRED, Federal Reserve Bank of St. Louis. This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=CORN

❚ Optional: interval

By default, interval=monthly. Strings monthly, quarterly, and annual are accepted.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=CORN&interval=monthly&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=CORN&interval=monthly&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Global Price of Cotton](https://www.alphavantage.co/documentation/#cotton)

This API returns the global price of cotton in monthly, quarterly, and annual horizons.

Source: International Monetary Fund ([IMF Terms of Use](https://www.imf.org/external/terms.htm)), Global price of Cotton, retrieved from FRED, Federal Reserve Bank of St. Louis. This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=COTTON

❚ Optional: interval

By default, interval=monthly. Strings monthly, quarterly, and annual are accepted.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=COTTON&interval=monthly&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=COTTON&interval=monthly&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Global Price of Sugar](https://www.alphavantage.co/documentation/#sugar)

This API returns the global price of sugar in monthly, quarterly, and annual horizons.

Source: International Monetary Fund ([IMF Terms of Use](https://www.imf.org/external/terms.htm)), Global price of Sugar, No. 11, World, retrieved from FRED, Federal Reserve Bank of St. Louis. This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=SUGAR

❚ Optional: interval

By default, interval=monthly. Strings monthly, quarterly, and annual are accepted.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=SUGAR&interval=monthly&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=SUGAR&interval=monthly&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Global Price of Coffee](https://www.alphavantage.co/documentation/#coffee)

This API returns the global price of coffee in monthly, quarterly, and annual horizons.

Source: International Monetary Fund ([IMF Terms of Use](https://www.imf.org/external/terms.htm)), Global price of Coffee, Other Mild Arabica, retrieved from FRED, Federal Reserve Bank of St. Louis. This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=COFFEE

❚ Optional: interval

By default, interval=monthly. Strings monthly, quarterly, and annual are accepted.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=COFFEE&interval=monthly&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=COFFEE&interval=monthly&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Global Price Index of All Commodities](https://www.alphavantage.co/documentation/#all-commodities)

This API returns the global price index of all commodities in monthly, quarterly, and annual temporal dimensions.

Source: International Monetary Fund ([IMF Terms of Use](https://www.imf.org/external/terms.htm)), Global Price Index of All Commodities, retrieved from FRED, Federal Reserve Bank of St. Louis. This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=ALL\_COMMODITIES

❚ Optional: interval

By default, interval=monthly. Strings monthly, quarterly, and annual are accepted.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=ALL_COMMODITIES&interval=monthly&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=ALL\_COMMODITIES&interval=monthly&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Economic Indicators](https://www.alphavantage.co/documentation/#economic-indicators)

APIs under this section provide key US economic indicators frequently used for investment strategy formulation and application development.

[REAL\_GDP Trending](https://www.alphavantage.co/documentation/#real-gdp)

This API returns the annual and quarterly Real GDP of the United States.

Source: U.S. Bureau of Economic Analysis, Real Gross Domestic Product, retrieved from FRED, Federal Reserve Bank of St. Louis. This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=REAL\_GDP

❚ Optional: interval

By default, interval=annual. Strings quarterly and annual are accepted.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=REAL_GDP&interval=annual&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=REAL\_GDP&interval=annual&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[REAL\_GDP\_PER\_CAPITA](https://www.alphavantage.co/documentation/#real-gdp-per-capita)

This API returns the quarterly Real GDP per Capita data of the United States.

Source: U.S. Bureau of Economic Analysis, Real gross domestic product per capita, retrieved from FRED, Federal Reserve Bank of St. Louis. This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=REAL\_GDP\_PER\_CAPITA

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=REAL_GDP_PER_CAPITA&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=REAL\_GDP\_PER\_CAPITA&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[TREASURY\_YIELD Trending](https://www.alphavantage.co/documentation/#treasury-yield)

This API returns the daily, weekly, and monthly US treasury yield of a given maturity timeline (e.g., 5 year, 30 year, etc).

Source: Board of Governors of the Federal Reserve System (US), Market Yield on U.S. Treasury Securities at 3-month, 2-year, 5-year, 7-year, 10-year, and 30-year Constant Maturities, Quoted on an Investment Basis, retrieved from FRED, Federal Reserve Bank of St. Louis. This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=TREASURY\_YIELD

❚ Optional: interval

By default, interval=monthly. Strings daily, weekly, and monthly are accepted.

❚ Optional: maturity

By default, maturity=10year. Strings 3month, 2year, 5year, 7year, 10year, and 30year are accepted.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=TREASURY_YIELD&interval=monthly&maturity=10year&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=TREASURY\_YIELD&interval=monthly&maturity=10year&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[FEDERAL\_FUNDS\_RATE](https://www.alphavantage.co/documentation/#interest-rate)

This API returns the daily, weekly, and monthly federal funds rate (interest rate) of the United States.

Source: Board of Governors of the Federal Reserve System (US), Federal Funds Effective Rate, retrieved from FRED, Federal Reserve Bank of St. Louis (https://fred.stlouisfed.org/series/FEDFUNDS). This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=FEDERAL\_FUNDS\_RATE

❚ Optional: interval

By default, interval=monthly. Strings daily, weekly, and monthly are accepted.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=FEDERAL_FUNDS_RATE&interval=monthly&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=FEDERAL\_FUNDS\_RATE&interval=monthly&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[CPI](https://www.alphavantage.co/documentation/#cpi)

This API returns the monthly and semiannual consumer price index (CPI) of the United States. CPI is widely regarded as the barometer of inflation levels in the broader economy.

Source: U.S. Bureau of Labor Statistics, Consumer Price Index for All Urban Consumers: All Items in U.S. City Average, retrieved from FRED, Federal Reserve Bank of St. Louis. This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=CPI

❚ Optional: interval

By default, interval=monthly. Strings monthly and semiannual are accepted.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=CPI&interval=monthly&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=CPI&interval=monthly&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[INFLATION](https://www.alphavantage.co/documentation/#inflation)

This API returns the annual inflation rates (consumer prices) of the United States.

Source: World Bank, Inflation, consumer prices for the United States, retrieved from FRED, Federal Reserve Bank of St. Louis. This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=INFLATION

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=INFLATION&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=INFLATION&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[RETAIL\_SALES](https://www.alphavantage.co/documentation/#retail-sales)

This API returns the monthly Advance Retail Sales: Retail Trade data of the United States.

Source: U.S. Census Bureau, Advance Retail Sales: Retail Trade, retrieved from FRED, Federal Reserve Bank of St. Louis (https://fred.stlouisfed.org/series/RSXFSN). This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=RETAIL\_SALES

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=RETAIL_SALES&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=RETAIL\_SALES&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[DURABLES](https://www.alphavantage.co/documentation/#durable-goods)

This API returns the monthly manufacturers' new orders of durable goods in the United States.

Source: U.S. Census Bureau, Manufacturers' New Orders: Durable Goods, retrieved from FRED, Federal Reserve Bank of St. Louis (https://fred.stlouisfed.org/series/UMDMNO). This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=DURABLES

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=DURABLES&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=DURABLES&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[UNEMPLOYMENT](https://www.alphavantage.co/documentation/#unemployment)

This API returns the monthly unemployment data of the United States. The unemployment rate represents the number of unemployed as a percentage of the labor force. Labor force data are restricted to people 16 years of age and older, who currently reside in 1 of the 50 states or the District of Columbia, who do not reside in institutions (e.g., penal and mental facilities, homes for the aged), and who are not on active duty in the Armed Forces ([source](https://fred.stlouisfed.org/series/UNRATE)).

Source: U.S. Bureau of Labor Statistics, Unemployment Rate, retrieved from FRED, Federal Reserve Bank of St. Louis. This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=UNEMPLOYMENT

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=UNEMPLOYMENT&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=UNEMPLOYMENT&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[NONFARM\_PAYROLL](https://www.alphavantage.co/documentation/#nonfarm-payroll)

This API returns the monthly US All Employees: Total Nonfarm (commonly known as Total Nonfarm Payroll), a measure of the number of U.S. workers in the economy that excludes proprietors, private household employees, unpaid volunteers, farm employees, and the unincorporated self-employed.

Source: U.S. Bureau of Labor Statistics, All Employees, Total Nonfarm, retrieved from FRED, Federal Reserve Bank of St. Louis. This data feed uses the FRED® API but is not endorsed or certified by the Federal Reserve Bank of St. Louis. By using this data feed, you agree to be bound by the [FRED® API Terms of Use](https://fred.stlouisfed.org/docs/api/terms_of_use.html).

**API Parameters**

**❚ Required: function**

The function of your choice. In this case, function=NONFARM\_PAYROLL

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

<https://www.alphavantage.co/query?function=NONFARM_PAYROLL&apikey=demo>

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=NONFARM\_PAYROLL&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[Technical Indicators](https://www.alphavantage.co/documentation/#technical-indicators)

Technical indicator APIs for a given equity or currency exchange pair, derived from the underlying time series based stock API and forex data. All indicators are calculated from adjusted time series data to eliminate artificial price/volume perturbations from historical split and dividend events.

[SMA Trending](https://www.alphavantage.co/documentation/#sma)

This API returns the simple moving average (SMA) values. See also: [SMA explainer](https://www.alphavantage.co/simple_moving_average_sma/) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=SimpleMA.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=SMA

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each moving average value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

Equity:

[https://www.alphavantage.co/query?**function**=SMA&**symbol**=IBM&**interval**=weekly&**time\_period**=10&**series\_type**=open&**apikey**=demo](https://www.alphavantage.co/query?function=SMA&symbol=IBM&interval=weekly&time_period=10&series_type=open&apikey=demo)

Forex (FX) or cryptocurrency pair:

[https://www.alphavantage.co/query?**function**=SMA&**symbol**=USDEUR&**interval**=weekly&**time\_period**=10&**series\_type**=open&**apikey**=demo](https://www.alphavantage.co/query?function=SMA&symbol=USDEUR&interval=weekly&time_period=10&series_type=open&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=SMA&symbol=IBM&interval=weekly&time\_period=10&series\_type=open&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[EMA Trending](https://www.alphavantage.co/documentation/#ema)

This API returns the exponential moving average (EMA) values. See also: [EMA explainer](https://www.alphavantage.co/exponential_moving_average_ema/) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=ExpMA.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=EMA

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each moving average value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

Equity:

[https://www.alphavantage.co/query?**function**=EMA&**symbol**=IBM&**interval**=weekly&**time\_period**=10&**series\_type**=open&**apikey**=demo](https://www.alphavantage.co/query?function=EMA&symbol=IBM&interval=weekly&time_period=10&series_type=open&apikey=demo)

Forex (FX) or cryptocurrency pair:

[https://www.alphavantage.co/query?**function**=EMA&**symbol**=USDEUR&**interval**=weekly&**time\_period**=10&**series\_type**=open&**apikey**=demo](https://www.alphavantage.co/query?function=EMA&symbol=USDEUR&interval=weekly&time_period=10&series_type=open&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=EMA&symbol=IBM&interval=weekly&time\_period=10&series\_type=open&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[WMA](https://www.alphavantage.co/documentation/#wma)

This API returns the weighted moving average (WMA) values. See also: [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=WeightedMA.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=WMA

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each moving average value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=WMA&**symbol**=IBM&**interval**=weekly&**time\_period**=10&**series\_type**=open&**apikey**=demo](https://www.alphavantage.co/query?function=WMA&symbol=IBM&interval=weekly&time_period=10&series_type=open&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=WMA&symbol=IBM&interval=weekly&time\_period=10&series\_type=open&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[DEMA](https://www.alphavantage.co/documentation/#dema)

This API returns the double exponential moving average (DEMA) values. See also: [Investopedia article](http://www.investopedia.com/articles/trading/10/double-exponential-moving-average.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=DEMA.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=DEMA

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each moving average value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=DEMA&**symbol**=IBM&**interval**=weekly&**time\_period**=10&**series\_type**=open&**apikey**=demo](https://www.alphavantage.co/query?function=DEMA&symbol=IBM&interval=weekly&time_period=10&series_type=open&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=DEMA&symbol=IBM&interval=weekly&time\_period=10&series\_type=open&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[TEMA](https://www.alphavantage.co/documentation/#tema)

This API returns the triple exponential moving average (TEMA) values. See also: [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=TEMA.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=TEMA

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each moving average value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=TEMA&**symbol**=IBM&**interval**=weekly&**time\_period**=10&**series\_type**=open&**apikey**=demo](https://www.alphavantage.co/query?function=TEMA&symbol=IBM&interval=weekly&time_period=10&series_type=open&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=TEMA&symbol=IBM&interval=weekly&time\_period=10&series\_type=open&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[TRIMA](https://www.alphavantage.co/documentation/#trima)

This API returns the triangular moving average (TRIMA) values. See also: [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=TriangularMA.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=TRIMA

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each moving average value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=TRIMA&**symbol**=IBM&**interval**=weekly&**time\_period**=10&**series\_type**=open&**apikey**=demo](https://www.alphavantage.co/query?function=TRIMA&symbol=IBM&interval=weekly&time_period=10&series_type=open&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=TRIMA&symbol=IBM&interval=weekly&time\_period=10&series\_type=open&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[KAMA](https://www.alphavantage.co/documentation/#kama)

This API returns the Kaufman adaptive moving average (KAMA) values.

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=KAMA

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each moving average value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=KAMA&**symbol**=IBM&**interval**=weekly&**time\_period**=10&**series\_type**=open&**apikey**=demo](https://www.alphavantage.co/query?function=KAMA&symbol=IBM&interval=weekly&time_period=10&series_type=open&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=KAMA&symbol=IBM&interval=weekly&time\_period=10&series\_type=open&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[MAMA](https://www.alphavantage.co/documentation/#mama)

This API returns the MESA adaptive moving average (MAMA) values.

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=MAMA

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: fastlimit

Positive floats are accepted. By default, fastlimit=0.01.

❚ Optional: slowlimit

Positive floats are accepted. By default, slowlimit=0.01.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=MAMA&**symbol**=IBM&**interval**=daily&**series\_type**=close&**fastlimit**=0.02&**apikey**=demo](https://www.alphavantage.co/query?function=MAMA&symbol=IBM&interval=daily&series_type=close&fastlimit=0.02&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=MAMA&symbol=IBM&interval=daily&series\_type=close&fastlimit=0.02&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[VWAP Trending Premium](https://www.alphavantage.co/documentation/#vwap)

This API returns the volume weighted average price (VWAP) for intraday time series. See also: [Investopedia article](https://www.investopedia.com/terms/v/vwap.asp).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=VWAP

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. In keeping with mainstream investment literatures on VWAP, the following intraday intervals are supported: 1min, 5min, 15min, 30min, 60min

❚ Optional: month

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=VWAP&**symbol**=IBM&**interval**=15min&**apikey**=demo](https://www.alphavantage.co/query?function=VWAP&symbol=IBM&interval=15min&apikey=demo)

💡 Tip: this is a premium API function. Subscribe to a [premium membership plan](https://www.alphavantage.co/premium/) to instantly unlock all premium APIs.

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=VWAP&symbol=IBM&interval=15min&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[T3](https://www.alphavantage.co/documentation/#t3)

This API returns the triple exponential moving average (T3) values. See also: [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=T3.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=T3

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each moving average value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=T3&**symbol**=IBM&**interval**=weekly&**time\_period**=10&**series\_type**=open&**apikey**=demo](https://www.alphavantage.co/query?function=T3&symbol=IBM&interval=weekly&time_period=10&series_type=open&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=T3&symbol=IBM&interval=weekly&time\_period=10&series\_type=open&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[MACD Trending Premium](https://www.alphavantage.co/documentation/#macd)

This API returns the moving average convergence / divergence (MACD) values. See also: [Investopedia article](http://www.investopedia.com/articles/forex/05/macddiverge.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=MACD.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=MACD

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: fastperiod

Positive integers are accepted. By default, fastperiod=12.

❚ Optional: slowperiod

Positive integers are accepted. By default, slowperiod=26.

❚ Optional: signalperiod

Positive integers are accepted. By default, signalperiod=9.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

Equity:

[https://www.alphavantage.co/query?**function**=MACD&**symbol**=IBM&**interval**=daily&**series\_type**=open&**apikey**=demo](https://www.alphavantage.co/query?function=MACD&symbol=IBM&interval=daily&series_type=open&apikey=demo)

Forex (FX) or cryptocurrency pair:

[https://www.alphavantage.co/query?**function**=MACD&**symbol**=USDEUR&**interval**=weekly&**series\_type**=open&**apikey**=demo](https://www.alphavantage.co/query?function=MACD&symbol=USDEUR&interval=weekly&series_type=open&apikey=demo)

💡 Tip: this is a premium API function. Subscribe to a [premium membership plan](https://www.alphavantage.co/premium/) to instantly unlock all premium APIs.

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=MACD&symbol=IBM&interval=daily&series\_type=open&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[MACDEXT](https://www.alphavantage.co/documentation/#macdext)

This API returns the moving average convergence / divergence values with controllable moving average type. See also: [Investopedia article](http://www.investopedia.com/articles/forex/05/macddiverge.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=MACD.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=MACDEXT

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: fastperiod

Positive integers are accepted. By default, fastperiod=12.

❚ Optional: slowperiod

Positive integers are accepted. By default, slowperiod=26.

❚ Optional: signalperiod

Positive integers are accepted. By default, signalperiod=9.

❚ Optional: fastmatype

Moving average type for the faster moving average. By default, fastmatype=0. Integers 0 - 8 are accepted with the following mappings. 0 = Simple Moving Average (SMA), 1 = Exponential Moving Average (EMA), 2 = Weighted Moving Average (WMA), 3 = Double Exponential Moving Average (DEMA), 4 = Triple Exponential Moving Average (TEMA), 5 = Triangular Moving Average (TRIMA), 6 = T3 Moving Average, 7 = Kaufman Adaptive Moving Average (KAMA), 8 = MESA Adaptive Moving Average (MAMA).

❚ Optional: slowmatype

Moving average type for the slower moving average. By default, slowmatype=0. Integers 0 - 8 are accepted with the following mappings. 0 = Simple Moving Average (SMA), 1 = Exponential Moving Average (EMA), 2 = Weighted Moving Average (WMA), 3 = Double Exponential Moving Average (DEMA), 4 = Triple Exponential Moving Average (TEMA), 5 = Triangular Moving Average (TRIMA), 6 = T3 Moving Average, 7 = Kaufman Adaptive Moving Average (KAMA), 8 = MESA Adaptive Moving Average (MAMA).

❚ Optional: signalmatype

Moving average type for the signal moving average. By default, signalmatype=0. Integers 0 - 8 are accepted with the following mappings. 0 = Simple Moving Average (SMA), 1 = Exponential Moving Average (EMA), 2 = Weighted Moving Average (WMA), 3 = Double Exponential Moving Average (DEMA), 4 = Triple Exponential Moving Average (TEMA), 5 = Triangular Moving Average (TRIMA), 6 = T3 Moving Average, 7 = Kaufman Adaptive Moving Average (KAMA), 8 = MESA Adaptive Moving Average (MAMA).

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=MACDEXT&**symbol**=IBM&**interval**=daily&**series\_type**=open&**apikey**=demo](https://www.alphavantage.co/query?function=MACDEXT&symbol=IBM&interval=daily&series_type=open&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=MACDEXT&symbol=IBM&interval=daily&series\_type=open&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[STOCH Trending](https://www.alphavantage.co/documentation/#stoch)

This API returns the stochastic oscillator (STOCH) values. See also: [Investopedia article](https://www.investopedia.com/terms/s/stochasticoscillator.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=StochasticOscillator.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=STOCH

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

❚ Optional: fastkperiod

The time period of the fastk moving average. Positive integers are accepted. By default, fastkperiod=5.

❚ Optional: slowkperiod

The time period of the slowk moving average. Positive integers are accepted. By default, slowkperiod=3.

❚ Optional: slowdperiod

The time period of the slowd moving average. Positive integers are accepted. By default, slowdperiod=3.

❚ Optional: slowkmatype

Moving average type for the slowk moving average. By default, slowkmatype=0. Integers 0 - 8 are accepted with the following mappings. 0 = Simple Moving Average (SMA), 1 = Exponential Moving Average (EMA), 2 = Weighted Moving Average (WMA), 3 = Double Exponential Moving Average (DEMA), 4 = Triple Exponential Moving Average (TEMA), 5 = Triangular Moving Average (TRIMA), 6 = T3 Moving Average, 7 = Kaufman Adaptive Moving Average (KAMA), 8 = MESA Adaptive Moving Average (MAMA).

❚ Optional: slowdmatype

Moving average type for the slowd moving average. By default, slowdmatype=0. Integers 0 - 8 are accepted with the following mappings. 0 = Simple Moving Average (SMA), 1 = Exponential Moving Average (EMA), 2 = Weighted Moving Average (WMA), 3 = Double Exponential Moving Average (DEMA), 4 = Triple Exponential Moving Average (TEMA), 5 = Triangular Moving Average (TRIMA), 6 = T3 Moving Average, 7 = Kaufman Adaptive Moving Average (KAMA), 8 = MESA Adaptive Moving Average (MAMA).

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

Equity:

[https://www.alphavantage.co/query?**function**=STOCH&**symbol**=IBM&**interval**=daily&**apikey**=demo](https://www.alphavantage.co/query?function=STOCH&symbol=IBM&interval=daily&apikey=demo)

Forex (FX) or cryptocurrency pair:

[https://www.alphavantage.co/query?**function**=STOCH&**symbol**=USDEUR&**interval**=weekly&**apikey**=demo](https://www.alphavantage.co/query?function=STOCH&symbol=USDEUR&interval=weekly&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=STOCH&symbol=IBM&interval=daily&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[STOCHF](https://www.alphavantage.co/documentation/#stochf)

This API returns the stochastic fast (STOCHF) values. See also: [Investopedia article](http://www.investopedia.com/university/indicator_oscillator/ind_osc8.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=StochasticOscillator.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=STOCHF

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

❚ Optional: fastkperiod

The time period of the fastk moving average. Positive integers are accepted. By default, fastkperiod=5.

❚ Optional: fastdperiod

The time period of the fastd moving average. Positive integers are accepted. By default, fastdperiod=3.

❚ Optional: fastdmatype

Moving average type for the fastd moving average. By default, fastdmatype=0. Integers 0 - 8 are accepted with the following mappings. 0 = Simple Moving Average (SMA), 1 = Exponential Moving Average (EMA), 2 = Weighted Moving Average (WMA), 3 = Double Exponential Moving Average (DEMA), 4 = Triple Exponential Moving Average (TEMA), 5 = Triangular Moving Average (TRIMA), 6 = T3 Moving Average, 7 = Kaufman Adaptive Moving Average (KAMA), 8 = MESA Adaptive Moving Average (MAMA).

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=STOCHF&**symbol**=IBM&**interval**=daily&**apikey**=demo](https://www.alphavantage.co/query?function=STOCHF&symbol=IBM&interval=daily&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=STOCHF&symbol=IBM&interval=daily&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[RSI Trending](https://www.alphavantage.co/documentation/#rsi)

This API returns the relative strength index (RSI) values. See also: [RSI explainer](https://www.alphavantage.co/relative_strength_index_rsi/) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=RSI.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=RSI

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each RSI value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

Equity:

[https://www.alphavantage.co/query?**function**=RSI&**symbol**=IBM&**interval**=weekly&**time\_period**=10&**series\_type**=open&**apikey**=demo](https://www.alphavantage.co/query?function=RSI&symbol=IBM&interval=weekly&time_period=10&series_type=open&apikey=demo)

Forex (FX) or cryptocurrency pair:

[https://www.alphavantage.co/query?**function**=RSI&**symbol**=USDEUR&**interval**=weekly&**time\_period**=10&**series\_type**=open&**apikey**=demo](https://www.alphavantage.co/query?function=RSI&symbol=USDEUR&interval=weekly&time_period=10&series_type=open&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=RSI&symbol=IBM&interval=weekly&time\_period=10&series\_type=open&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[STOCHRSI](https://www.alphavantage.co/documentation/#stochrsi)

This API returns the stochastic relative strength index (STOCHRSI) values. See also: [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=StochRSI.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=STOCHRSI

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each STOCHRSI value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: fastkperiod

The time period of the fastk moving average. Positive integers are accepted. By default, fastkperiod=5.

❚ Optional: fastdperiod

The time period of the fastd moving average. Positive integers are accepted. By default, fastdperiod=3.

❚ Optional: fastdmatype

Moving average type for the fastd moving average. By default, fastdmatype=0. Integers 0 - 8 are accepted with the following mappings. 0 = Simple Moving Average (SMA), 1 = Exponential Moving Average (EMA), 2 = Weighted Moving Average (WMA), 3 = Double Exponential Moving Average (DEMA), 4 = Triple Exponential Moving Average (TEMA), 5 = Triangular Moving Average (TRIMA), 6 = T3 Moving Average, 7 = Kaufman Adaptive Moving Average (KAMA), 8 = MESA Adaptive Moving Average (MAMA).

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=STOCHRSI&**symbol**=IBM&**interval**=daily&**time\_period**=10&**series\_type**=close&**fastkperiod**=6&**fastdmatype**=1&**apikey**=demo](https://www.alphavantage.co/query?function=STOCHRSI&symbol=IBM&interval=daily&time_period=10&series_type=close&fastkperiod=6&fastdmatype=1&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=STOCHRSI&symbol=IBM&interval=daily&time\_period=10&series\_type=close&fastkperiod=6&fastdmatype=1&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[WILLR](https://www.alphavantage.co/documentation/#willr)

This API returns the Williams' %R (WILLR) values. See also: [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=WilliamsR.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=WILLR

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each WILLR value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=WILLR&**symbol**=IBM&**interval**=daily&**time\_period**=10&**apikey**=demo](https://www.alphavantage.co/query?function=WILLR&symbol=IBM&interval=daily&time_period=10&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=WILLR&symbol=IBM&interval=daily&time\_period=10&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[ADX Trending](https://www.alphavantage.co/documentation/#adx)

This API returns the average directional movement index (ADX) values. See also: [Investopedia article](http://www.investopedia.com/articles/trading/07/adx-trend-indicator.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=ADX.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=ADX

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each ADX value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

Equity:

[https://www.alphavantage.co/query?**function**=ADX&**symbol**=IBM&**interval**=daily&**time\_period**=10&**apikey**=demo](https://www.alphavantage.co/query?function=ADX&symbol=IBM&interval=daily&time_period=10&apikey=demo)

Forex (FX) or cryptocurrency pair:

[https://www.alphavantage.co/query?**function**=ADX&**symbol**=USDEUR&**interval**=weekly&**time\_period**=10&**apikey**=demo](https://www.alphavantage.co/query?function=ADX&symbol=USDEUR&interval=weekly&time_period=10&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=ADX&symbol=IBM&interval=daily&time\_period=10&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[ADXR](https://www.alphavantage.co/documentation/#adxr)

This API returns the average directional movement index rating (ADXR) values. See also: [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=ADXR.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=ADXR

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each ADXR value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=ADXR&**symbol**=IBM&**interval**=daily&**time\_period**=10&**apikey**=demo](https://www.alphavantage.co/query?function=ADXR&symbol=IBM&interval=daily&time_period=10&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=ADXR&symbol=IBM&interval=daily&time\_period=10&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[APO](https://www.alphavantage.co/documentation/#apo)

This API returns the absolute price oscillator (APO) values. See also: [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=PriceOscillator.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=APO

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: fastperiod

Positive integers are accepted. By default, fastperiod=12.

❚ Optional: slowperiod

Positive integers are accepted. By default, slowperiod=26.

❚ Optional: matype

Moving average type. By default, matype=0. Integers 0 - 8 are accepted with the following mappings. 0 = Simple Moving Average (SMA), 1 = Exponential Moving Average (EMA), 2 = Weighted Moving Average (WMA), 3 = Double Exponential Moving Average (DEMA), 4 = Triple Exponential Moving Average (TEMA), 5 = Triangular Moving Average (TRIMA), 6 = T3 Moving Average, 7 = Kaufman Adaptive Moving Average (KAMA), 8 = MESA Adaptive Moving Average (MAMA).

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=APO&**symbol**=IBM&**interval**=daily&**series\_type**=close&**fastperiod**=10&**matype**=1&**apikey**=demo](https://www.alphavantage.co/query?function=APO&symbol=IBM&interval=daily&series_type=close&fastperiod=10&matype=1&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=APO&symbol=IBM&interval=daily&series\_type=close&fastperiod=10&matype=1&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[PPO](https://www.alphavantage.co/documentation/#ppo)

This API returns the percentage price oscillator (PPO) values. See also: [Investopedia article](http://www.investopedia.com/articles/investing/051214/use-percentage-price-oscillator-elegant-indicator-picking-stocks.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=PriceOscillatorPct.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=PPO

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: fastperiod

Positive integers are accepted. By default, fastperiod=12.

❚ Optional: slowperiod

Positive integers are accepted. By default, slowperiod=26.

❚ Optional: matype

Moving average type. By default, matype=0. Integers 0 - 8 are accepted with the following mappings. 0 = Simple Moving Average (SMA), 1 = Exponential Moving Average (EMA), 2 = Weighted Moving Average (WMA), 3 = Double Exponential Moving Average (DEMA), 4 = Triple Exponential Moving Average (TEMA), 5 = Triangular Moving Average (TRIMA), 6 = T3 Moving Average, 7 = Kaufman Adaptive Moving Average (KAMA), 8 = MESA Adaptive Moving Average (MAMA).

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=PPO&**symbol**=IBM&**interval**=daily&**series\_type**=close&**fastperiod**=10&**matype**=1&**apikey**=demo](https://www.alphavantage.co/query?function=PPO&symbol=IBM&interval=daily&series_type=close&fastperiod=10&matype=1&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=PPO&symbol=IBM&interval=daily&series\_type=close&fastperiod=10&matype=1&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[MOM](https://www.alphavantage.co/documentation/#mom)

This API returns the momentum (MOM) values. See also: [Investopedia article](http://www.investopedia.com/articles/technical/03/070203.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=Momentum.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=MOM

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each MOM value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=MOM&**symbol**=IBM&**interval**=daily&**time\_period**=10&**series\_type**=close&**apikey**=demo](https://www.alphavantage.co/query?function=MOM&symbol=IBM&interval=daily&time_period=10&series_type=close&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=MOM&symbol=IBM&interval=daily&time\_period=10&series\_type=close&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[BOP](https://www.alphavantage.co/documentation/#bop)

This API returns the balance of power (BOP) values.

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=BOP

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=BOP&**symbol**=IBM&**interval**=daily&**apikey**=demo](https://www.alphavantage.co/query?function=BOP&symbol=IBM&interval=daily&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=BOP&symbol=IBM&interval=daily&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[CCI Trending](https://www.alphavantage.co/documentation/#cci)

This API returns the commodity channel index (CCI) values. See also: [Investopedia article](http://www.investopedia.com/articles/trading/05/041805.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=CCI.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=CCI

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each CCI value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

Equity:

[https://www.alphavantage.co/query?**function**=CCI&**symbol**=IBM&**interval**=daily&**time\_period**=10&**apikey**=demo](https://www.alphavantage.co/query?function=CCI&symbol=IBM&interval=daily&time_period=10&apikey=demo)

Forex (FX) or cryptocurrency pair:

[https://www.alphavantage.co/query?**function**=CCI&**symbol**=USDEUR&**interval**=weekly&**time\_period**=10&**apikey**=demo](https://www.alphavantage.co/query?function=CCI&symbol=USDEUR&interval=weekly&time_period=10&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=CCI&symbol=IBM&interval=daily&time\_period=10&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[CMO](https://www.alphavantage.co/documentation/#cmo)

This API returns the Chande momentum oscillator (CMO) values. See also: [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=CMO.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=CMO

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each CMO value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=CMO&**symbol**=IBM&**interval**=weekly&**time\_period**=10&**series\_type**=close&**apikey**=demo](https://www.alphavantage.co/query?function=CMO&symbol=IBM&interval=weekly&time_period=10&series_type=close&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=CMO&symbol=IBM&interval=weekly&time\_period=10&series\_type=close&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[ROC](https://www.alphavantage.co/documentation/#roc)

This API returns the rate of change (ROC) values. See also: [Investopedia article](http://www.investopedia.com/articles/technical/092401.asp).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=ROC

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each ROC value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=ROC&**symbol**=IBM&**interval**=weekly&**time\_period**=10&**series\_type**=close&**apikey**=demo](https://www.alphavantage.co/query?function=ROC&symbol=IBM&interval=weekly&time_period=10&series_type=close&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=ROC&symbol=IBM&interval=weekly&time\_period=10&series\_type=close&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[ROCR](https://www.alphavantage.co/documentation/#rocr)

This API returns the rate of change ratio (ROCR) values. See also: [Investopedia article](http://www.investopedia.com/articles/technical/092401.asp).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=ROCR

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each ROCR value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=ROCR&**symbol**=IBM&**interval**=daily&**time\_period**=10&**series\_type**=close&**apikey**=demo](https://www.alphavantage.co/query?function=ROCR&symbol=IBM&interval=daily&time_period=10&series_type=close&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=ROCR&symbol=IBM&interval=daily&time\_period=10&series\_type=close&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[AROON Trending](https://www.alphavantage.co/documentation/#aroon)

This API returns the Aroon (AROON) values. See also: [Investopedia article](http://www.investopedia.com/articles/trading/06/aroon.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=Aroon.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=AROON

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each AROON value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

Equity:

[https://www.alphavantage.co/query?**function**=AROON&**symbol**=IBM&**interval**=daily&**time\_period**=14&**apikey**=demo](https://www.alphavantage.co/query?function=AROON&symbol=IBM&interval=daily&time_period=14&apikey=demo)

Forex (FX) or cryptocurrency pair:

[https://www.alphavantage.co/query?**function**=AROON&**symbol**=USDEUR&**interval**=weekly&**time\_period**=14&**apikey**=demo](https://www.alphavantage.co/query?function=AROON&symbol=USDEUR&interval=weekly&time_period=14&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=AROON&symbol=IBM&interval=daily&time\_period=14&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[AROONOSC](https://www.alphavantage.co/documentation/#aroonosc)

This API returns the Aroon oscillator (AROONOSC) values. See also: [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=AroonOscillator.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=AROONOSC

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each AROONOSC value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=AROONOSC&**symbol**=IBM&**interval**=daily&**time\_period**=10&**apikey**=demo](https://www.alphavantage.co/query?function=AROONOSC&symbol=IBM&interval=daily&time_period=10&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=AROONOSC&symbol=IBM&interval=daily&time\_period=10&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[MFI](https://www.alphavantage.co/documentation/#mfi)

This API returns the money flow index (MFI) values. See also: [Investopedia article](http://www.investopedia.com/articles/technical/03/072303.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=MoneyFlowIndex.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=MFI

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each MFI value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=MFI&**symbol**=IBM&**interval**=weekly&**time\_period**=10&**apikey**=demo](https://www.alphavantage.co/query?function=MFI&symbol=IBM&interval=weekly&time_period=10&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=MFI&symbol=IBM&interval=weekly&time\_period=10&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[TRIX](https://www.alphavantage.co/documentation/#trix)

This API returns the 1-day rate of change of a triple smooth exponential moving average (TRIX) values. See also: [Investopedia article](http://www.investopedia.com/articles/technical/02/092402.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=TRIX.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=TRIX

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each TRIX value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=TRIX&**symbol**=IBM&**interval**=daily&**time\_period**=10&**series\_type**=close&**apikey**=demo](https://www.alphavantage.co/query?function=TRIX&symbol=IBM&interval=daily&time_period=10&series_type=close&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=TRIX&symbol=IBM&interval=daily&time\_period=10&series\_type=close&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[ULTOSC](https://www.alphavantage.co/documentation/#ultosc)

This API returns the ultimate oscillator (ULTOSC) values. See also: [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=UltimateOsc.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=ULTOSC

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

❚ Optional: timeperiod1

The first time period for the indicator. Positive integers are accepted. By default, timeperiod1=7.

❚ Optional: timeperiod2

The second time period for the indicator. Positive integers are accepted. By default, timeperiod2=14.

❚ Optional: timeperiod3

The third time period for the indicator. Positive integers are accepted. By default, timeperiod3=28.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Examples (click for JSON output)**

[https://www.alphavantage.co/query?**function**=ULTOSC&**symbol**=IBM&**interval**=daily&**timeperiod1**=8&**apikey**=demo](https://www.alphavantage.co/query?function=ULTOSC&symbol=IBM&interval=daily&timeperiod1=8&apikey=demo)

[https://www.alphavantage.co/query?**function**=ULTOSC&**symbol**=IBM&**interval**=daily&**apikey**=demo](https://www.alphavantage.co/query?function=ULTOSC&symbol=IBM&interval=weekly&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=ULTOSC&symbol=IBM&interval=daily&timeperiod1=8&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[DX](https://www.alphavantage.co/documentation/#dx)

This API returns the directional movement index (DX) values. See also: [Investopedia article](http://www.investopedia.com/articles/technical/02/050602.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=DX.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=DX

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each DX value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=DX&**symbol**=IBM&**interval**=daily&**time\_period**=10&**apikey**=demo](https://www.alphavantage.co/query?function=DX&symbol=IBM&interval=daily&time_period=10&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=DX&symbol=IBM&interval=daily&time\_period=10&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[MINUS\_DI](https://www.alphavantage.co/documentation/#minusdi)

This API returns the minus directional indicator (MINUS\_DI) values. See also: [Investopedia article](http://www.investopedia.com/articles/technical/02/050602.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=DI.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=MINUS\_DI

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each MINUS\_DI value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=MINUS\_DI&**symbol**=IBM&**interval**=weekly&**time\_period**=10&**apikey**=demo](https://www.alphavantage.co/query?function=MINUS_DI&symbol=IBM&interval=weekly&time_period=10&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=MINUS\_DI&symbol=IBM&interval=weekly&time\_period=10&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[PLUS\_DI](https://www.alphavantage.co/documentation/#plusdi)

This API returns the plus directional indicator (PLUS\_DI) values. See also: [Investopedia article](http://www.investopedia.com/articles/technical/02/050602.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=DI.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=PLUS\_DI

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each PLUS\_DI value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=PLUS\_DI&**symbol**=IBM&**interval**=daily&**time\_period**=10&**apikey**=demo](https://www.alphavantage.co/query?function=PLUS_DI&symbol=IBM&interval=daily&time_period=10&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=PLUS\_DI&symbol=IBM&interval=daily&time\_period=10&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[MINUS\_DM](https://www.alphavantage.co/documentation/#minusdm)

This API returns the minus directional movement (MINUS\_DM) values. See also: [Investopedia article](http://www.investopedia.com/articles/technical/02/050602.asp)

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=MINUS\_DM

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each MINUS\_DM value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=MINUS\_DM&**symbol**=IBM&**interval**=daily&**time\_period**=10&**apikey**=demo](https://www.alphavantage.co/query?function=MINUS_DM&symbol=IBM&interval=daily&time_period=10&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=MINUS\_DM&symbol=IBM&interval=daily&time\_period=10&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[PLUS\_DM](https://www.alphavantage.co/documentation/#plusdm)

This API returns the plus directional movement (PLUS\_DM) values. See also: [Investopedia article](http://www.investopedia.com/articles/technical/02/050602.asp)

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=PLUS\_DM

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each PLUS\_DM value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=PLUS\_DM&**symbol**=IBM&**interval**=daily&**time\_period**=10&**apikey**=demo](https://www.alphavantage.co/query?function=PLUS_DM&symbol=IBM&interval=daily&time_period=10&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=PLUS\_DM&symbol=IBM&interval=daily&time\_period=10&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[BBANDS Trending](https://www.alphavantage.co/documentation/#bbands)

This API returns the Bollinger bands (BBANDS) values. See also: [Investopedia article](http://www.investopedia.com/articles/technical/04/030304.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=Bollinger.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=BBANDS

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each BBANDS value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: nbdevup

The standard deviation multiplier of the upper band. Positive integers are accepted. By default, nbdevup=2.

❚ Optional: nbdevdn

The standard deviation multiplier of the lower band. Positive integers are accepted. By default, nbdevdn=2.

❚ Optional: matype

Moving average type of the time series. By default, matype=0. Integers 0 - 8 are accepted with the following mappings. 0 = Simple Moving Average (SMA), 1 = Exponential Moving Average (EMA), 2 = Weighted Moving Average (WMA), 3 = Double Exponential Moving Average (DEMA), 4 = Triple Exponential Moving Average (TEMA), 5 = Triangular Moving Average (TRIMA), 6 = T3 Moving Average, 7 = Kaufman Adaptive Moving Average (KAMA), 8 = MESA Adaptive Moving Average (MAMA).

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

Equity:

[https://www.alphavantage.co/query?**function**=BBANDS&**symbol**=IBM&**interval**=weekly&**time\_period**=5&**series\_type**=close&**nbdevup**=3&**nbdevdn**=3&**apikey**=demo](https://www.alphavantage.co/query?function=BBANDS&symbol=IBM&interval=weekly&time_period=5&series_type=close&nbdevup=3&nbdevdn=3&apikey=demo)

Forex (FX) or cryptocurrency pair:

[https://www.alphavantage.co/query?**function**=BBANDS&**symbol**=USDEUR&**interval**=weekly&**time\_period**=5&**series\_type**=close&**nbdevup**=3&**nbdevdn**=3&**apikey**=demo](https://www.alphavantage.co/query?function=BBANDS&symbol=USDEUR&interval=weekly&time_period=5&series_type=close&nbdevup=3&nbdevdn=3&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=BBANDS&symbol=IBM&interval=weekly&time\_period=5&series\_type=close&nbdevup=3&nbdevdn=3&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[MIDPOINT](https://www.alphavantage.co/documentation/#midpoint)

This API returns the midpoint (MIDPOINT) values. MIDPOINT = (highest value + lowest value)/2.

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=MIDPOINT

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each MIDPOINT value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=MIDPOINT&**symbol**=IBM&**interval**=daily&**time\_period**=10&**series\_type**=close&**apikey**=demo](https://www.alphavantage.co/query?function=MIDPOINT&symbol=IBM&interval=daily&time_period=10&series_type=close&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=MIDPOINT&symbol=IBM&interval=daily&time\_period=10&series\_type=close&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[MIDPRICE](https://www.alphavantage.co/documentation/#midprice)

This API returns the midpoint price (MIDPRICE) values. MIDPRICE = (highest high + lowest low)/2.

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=MIDPRICE

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each MIDPRICE value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=MIDPRICE&**symbol**=IBM&**interval**=daily&**time\_period**=10&**apikey**=demo](https://www.alphavantage.co/query?function=MIDPRICE&symbol=IBM&interval=daily&time_period=10&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=MIDPRICE&symbol=IBM&interval=daily&time\_period=10&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[SAR](https://www.alphavantage.co/documentation/#sar)

This API returns the parabolic SAR (SAR) values. See also: [Investopedia article](http://www.investopedia.com/articles/technical/02/042202.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=SAR.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=SAR

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

❚ Optional: acceleration

The acceleration factor. Positive floats are accepted. By default, acceleration=0.01.

❚ Optional: maximum

The acceleration factor maximum value. Positive floats are accepted. By default, maximum=0.20.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=SAR&**symbol**=IBM&**interval**=weekly&**acceleration**=0.05&**maximum**=0.25&**apikey**=demo](https://www.alphavantage.co/query?function=SAR&symbol=IBM&interval=weekly&acceleration=0.05&maximum=0.25&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=SAR&symbol=IBM&interval=weekly&acceleration=0.05&maximum=0.25&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[TRANGE](https://www.alphavantage.co/documentation/#trange)

This API returns the true range (TRANGE) values. See also: [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=TR.htm)

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=TRANGE

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=TRANGE&**symbol**=IBM&**interval**=daily&**apikey**=demo](https://www.alphavantage.co/query?function=TRANGE&symbol=IBM&interval=daily&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=TRANGE&symbol=IBM&interval=daily&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[ATR](https://www.alphavantage.co/documentation/#atr)

This API returns the average true range (ATR) values. See also: [Investopedia article](http://www.investopedia.com/articles/trading/08/average-true-range.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=ATR.htm)

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=ATR

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each ATR value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=ATR&**symbol**=IBM&**interval**=daily&**time\_period**=14&**apikey**=demo](https://www.alphavantage.co/query?function=ATR&symbol=IBM&interval=daily&time_period=14&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=ATR&symbol=IBM&interval=daily&time\_period=14&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[NATR](https://www.alphavantage.co/documentation/#natr)

This API returns the normalized average true range (NATR) values.

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=NATR

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required:time\_period**

Number of data points used to calculate each NATR value. Positive integers are accepted (e.g., time\_period=60, time\_period=200)

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=NATR&**symbol**=IBM&**interval**=weekly&**time\_period**=14&**apikey**=demo](https://www.alphavantage.co/query?function=NATR&symbol=IBM&interval=weekly&time_period=14&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=NATR&symbol=IBM&interval=weekly&time\_period=14&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[AD Trending](https://www.alphavantage.co/documentation/#ad)

This API returns the Chaikin A/D line (AD) values. See also: [Investopedia article](http://www.investopedia.com/articles/active-trading/031914/understanding-chaikin-oscillator.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=AccumDist.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=AD

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=AD&**symbol**=IBM&**interval**=daily&**apikey**=demo](https://www.alphavantage.co/query?function=AD&symbol=IBM&interval=daily&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=AD&symbol=IBM&interval=daily&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[ADOSC](https://www.alphavantage.co/documentation/#adosc)

This API returns the Chaikin A/D oscillator (ADOSC) values. See also: [Investopedia article](http://www.investopedia.com/articles/active-trading/031914/understanding-chaikin-oscillator.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=AccumDist.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=ADOSC

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

❚ Optional: fastperiod

The time period of the fast EMA. Positive integers are accepted. By default, fastperiod=3.

❚ Optional: slowperiod

The time period of the slow EMA. Positive integers are accepted. By default, slowperiod=10.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example(click for JSON output)**

[https://www.alphavantage.co/query?**function**=ADOSC&**symbol**=IBM&**interval**=daily&**fastperiod**=5&**apikey**=demo](https://www.alphavantage.co/query?function=ADOSC&symbol=IBM&interval=daily&fastperiod=5&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=ADOSC&symbol=IBM&interval=daily&fastperiod=5&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[OBV Trending](https://www.alphavantage.co/documentation/#obv)

This API returns the on balance volume (OBV) values. See also: [Investopedia article](http://www.investopedia.com/articles/technical/100801.asp) and [mathematical reference](http://www.fmlabs.com/reference/default.htm?url=OBV.htm).

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=OBV

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=OBV&**symbol**=IBM&**interval**=weekly&**apikey**=demo](https://www.alphavantage.co/query?function=OBV&symbol=IBM&interval=weekly&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=OBV&symbol=IBM&interval=weekly&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[HT\_TRENDLINE](https://www.alphavantage.co/documentation/#httrendline)

This API returns the Hilbert transform, instantaneous trendline (HT\_TRENDLINE) values.

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=HT\_TRENDLINE

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=HT\_TRENDLINE&**symbol**=IBM&**interval**=daily&**series\_type**=close&**apikey**=demo](https://www.alphavantage.co/query?function=HT_TRENDLINE&symbol=IBM&interval=daily&series_type=close&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=HT\_TRENDLINE&symbol=IBM&interval=daily&series\_type=close&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[HT\_SINE](https://www.alphavantage.co/documentation/#htsine)

This API returns the Hilbert transform, sine wave (HT\_SINE) values.

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=HT\_SINE

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=HT\_SINE&**symbol**=IBM&**interval**=daily&**series\_type**=close&**apikey**=demo](https://www.alphavantage.co/query?function=HT_SINE&symbol=IBM&interval=daily&series_type=close&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=HT\_SINE&symbol=IBM&interval=daily&series\_type=close&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[HT\_TRENDMODE](https://www.alphavantage.co/documentation/#httrendmode)

This API returns the Hilbert transform, trend vs cycle mode (HT\_TRENDMODE) values.

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=HT\_TRENDMODE

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=HT\_TRENDMODE&**symbol**=IBM&**interval**=weekly&**series\_type**=close&**apikey**=demo](https://www.alphavantage.co/query?function=HT_TRENDMODE&symbol=IBM&interval=weekly&series_type=close&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=HT\_TRENDMODE&symbol=IBM&interval=weekly&series\_type=close&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[HT\_DCPERIOD](https://www.alphavantage.co/documentation/#htdcperiod)

This API returns the Hilbert transform, dominant cycle period (HT\_DCPERIOD) values.

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=HT\_DCPERIOD

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=HT\_DCPERIOD&**symbol**=IBM&**interval**=daily&**series\_type**=close&**apikey**=demo](https://www.alphavantage.co/query?function=HT_DCPERIOD&symbol=IBM&interval=daily&series_type=close&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=HT\_DCPERIOD&symbol=IBM&interval=daily&series\_type=close&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[HT\_DCPHASE](https://www.alphavantage.co/documentation/#htdcphase)

This API returns the Hilbert transform, dominant cycle phase (HT\_DCPHASE) values.

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=HT\_DCPHASE

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=HT\_DCPHASE&**symbol**=IBM&**interval**=daily&**series\_type**=close&**apikey**=demo](https://www.alphavantage.co/query?function=HT_DCPHASE&symbol=IBM&interval=daily&series_type=close&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=HT\_DCPHASE&symbol=IBM&interval=daily&series\_type=close&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

[HT\_PHASOR](https://www.alphavantage.co/documentation/#htphasor)

This API returns the Hilbert transform, phasor components (HT\_PHASOR) values.

**API Parameters**

**❚ Required: function**

The technical indicator of your choice. In this case, function=HT\_PHASOR

**❚ Required: symbol**

The name of the ticker of your choice. For example: symbol=IBM

**❚ Required: interval**

Time interval between two consecutive data points in the time series. The following values are supported: 1min, 5min, 15min, 30min, 60min, daily, weekly, monthly

❚ Optional: month

Note: this parameter is ONLY applicable to intraday intervals (1min, 5min, 15min, 30min, and 60min) for the equity markets. The daily/weekly/monthly intervals are agnostic to this parameter.

By default, this parameter is not set and the technical indicator values will be calculated based on the most recent 30 days of intraday data. You can use the month parameter (in YYYY-MM format) to compute intraday technical indicators for a specific month in history. For example, month=2009-01. Any month equal to or later than 2000-01 (January 2000) is supported.

**❚ Required: series\_type**

The desired price type in the time series. Four types are supported: close, open, high, low

❚ Optional: datatype

By default, datatype=json. Strings json and csv are accepted with the following specifications: json returns the daily time series in JSON format; csv returns the time series as a CSV (comma separated value) file.

**❚ Required: apikey**

Your API key. Claim your free API key [here](https://www.alphavantage.co/support/#api-key).

**Example (click for JSON output)**

[https://www.alphavantage.co/query?**function**=HT\_PHASOR&**symbol**=IBM&**interval**=weekly&**series\_type**=close&**apikey**=demo](https://www.alphavantage.co/query?function=HT_PHASOR&symbol=IBM&interval=weekly&series_type=close&apikey=demo)

**Language-specific guides**

Python NodeJS PHP C#/.NET Other

**import** requests

*# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-key*

url = 'https://www.alphavantage.co/query?function=HT\_PHASOR&symbol=IBM&interval=weekly&series\_type=close&apikey=demo'

r = requests.get(url)

data = r.json()

print(data)

Copyright © Alpha Vantage Inc. 2017-2024