Table of Contents

Alpha Vantage API Documentation

Our stock APIs © 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. Examples in this documentation are for demo purposes. Claim your free API key today to explore our full API offerings!

Time Series Stock Data APIs

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

This API returns current and 20+ years of historical intraday OHLCV time series of the equity specified, covering pre-market and post-market 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 (pre-market and post-market) 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.

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`

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`

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`

Downloadable CSV file:

`https://www.alphavantage.co/query?function=TIME_SERIES_INTRADAY&symbol=IBM&interval=5min&apikey=demo&datatype=csv`

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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

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, 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.

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&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

Sample ticker traded in Canada - Toronto Stock Exchange

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

Sample ticker traded in Germany - XETRA

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

Sample ticker traded in China - Shanghai Stock Exchange

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

The above is just a small sample of the 100,000+ symbols we support. Please refer to our Search Endpoint 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

Language-specific guides

Python NodeJS PHP C#/.NET Other

```python
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

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.

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&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

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

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

Sample ticker traded in Germany - XETRA

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

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

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

The above is just a small sample of the 100,000+ symbols we support. Please refer to our Search Endpoint 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

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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

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.

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=TSCO.LON&apikey=demo

Downloadable CSV file:
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-ke
y
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

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.

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=TSCO.LON&apikey=demo

Downloadable CSV file:
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

```python
import requests

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

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.

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=TSCO.LON&apikey=demo

Downloadable CSV file:

https://www.alphavantage.co/query?function=TIME_SERIES_MONTHLY&symbol=IBM&apikey=demo&datatype=csv

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
import requests

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

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.

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=TSCO.LON&apikey=demo

Downloadable CSV file:

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

```python
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

This endpoint returns the latest price and volume information for a ticker of your choice. You can specify one ticker per API request.

If you would like to query a large universe of tickers in bulk, you may want to try out our Realtime Bulk Quotes API, which accepts up to 100 tickers per API request.

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.

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=300135.SHZ&apikey=demo

Downloadable CSV file:
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 for your personal use. For commercial use, please contact sales.

* Realtime and 15-minute delayed US market data is regulated by the stock exchanges, FINRA, and the SEC. Learn more 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)
```

Realtime Bulk Quotes Premium

This API returns realtime quotes for US-traded symbols in bulk, accepting up to 100 symbols per API request and covering both regular and extended (pre-market and post-market) trading hours. You can use this endpoint as a high-throughput alternative to the Global Quote API, which accepts one symbol per API request.

API Parameters

Required: `function`

The time series of your choice. In this case, `function=REALTIME_BULK_QUOTES`

Required: `symbol`

Up to 100 symbols separated by comma. For example: `symbol=MSFT,AAPL,IBM`. If more than 100 symbols are provided, only the first 100 symbols will be honored as part of the API input.

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.

Examples (click for JSON output)

`https://www.alphavantage.co/query?function=REALTIME_BULK_QUOTES&symbol=MSFT,AAPL,IBM&apikey=demo`

>> Tip: this is a premium API function. Please subscribe to any premium membership plan that mentions "Realtime US Market Data" in its description to unlock this endpoint for your personal use. For commercial use, please contact sales.

Language-specific guides

Python NodeJS PHP C#/.NET Other

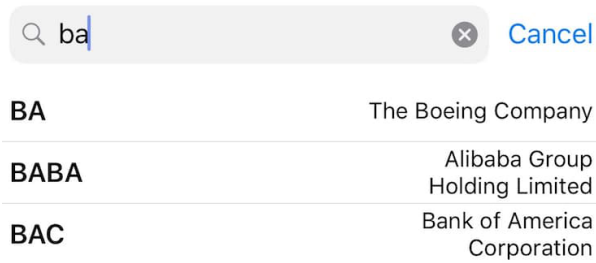
```python
import requests

# 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_BULK_QUOTES&symbol=MSFT,AAPL,IBM&apikey=demo'
r = requests.get(url)
data = r.json()

print(data)
```

Search Endpoint Utility

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.

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=tencent&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

Downloadable CSV file:

https://www.alphavantage.co/query?function=SYMBOL_SEARCH&keywords=BA&apikey=demo&datatype=csv

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
import requests

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

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.

Examples (click for JSON output)

https://www.alphavantage.co/query?function=MARKET_STATUS&apikey=demo

Language-specific guides

Python NodeJS PHP C#/.NET Other

```python
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

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

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: `require_greeks`

Enable greeks & implied volatility (IV) fields. By default, `require_greeks=false`. Set `require_greeks=true` to enable greeks & IVs in the API response.

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.

Examples (click for JSON output)

By default, the entire realtime option chain is returned

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

Set require_greeks=true to enable greeks & implied volatility (IV) fields in the API response

https://www.alphavantage.co/query?function=REALTIME_OPTIONS&symbol=IBM&require_greeks=true&apikey=demo

Query a specific contract (instead of the entire option chain) with greeks & IVs enabled

https://www.alphavantage.co/query?function=REALTIME_OPTIONS&symbol=IBM&require_greeks=true&contract=IBM270115C00390000&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 to unlock realtime options data.

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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

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.

Example (click for JSON output)

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

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 2008-01-01)

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

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
import requests

# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-ke
y
url = 'https://www.alphavantage.co/query?function=HISTORICAL_OPTIONS&symbol=IBM&apikey=demo'
r = requests.get(url)
data = r.json()

print(data)
```

Alpha 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

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.

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`

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`

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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)
```

Earnings Call Transcript Trending

This API returns the earnings call transcript for a given company in a specific quarter, covering over 15 years of history and enriched with LLM-based sentiment signals.

API Parameters

Required: `function`

The function of your choice. In this case, `function=EARNINGS_CALL_TRANSCRIPT`

Required: `symbol`

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

Required: `quarter`

Fiscal quarter in YYYYQM format. For example: `quarter=2024Q1`. Any quarter since 2010Q1 is supported.

Required: `apikey`

Your API key. Claim your free API key here.

Example (click for JSON output)

https://www.alphavantage.co/query?function=EARNINGS_CALL_TRANSCRIPT&symbol=IBM&quarter=2024Q1&apikey=demo

Language-specific guides

Python NodeJS PHP C#/.NET Other

```python
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=INSIDER_TRANSACTIONS&symbol=IBM&apikey=demo'
r = requests.get(url)
data = r.json()

print(data)
```

Top Gainers, Losers, and Most Actively Traded Tickers (US Market)

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.

Examples (click for JSON output)

`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 for your personal use. For commercial use, please contact sales.

* Realtime and 15-minute delayed US market data is regulated by the stock exchanges, FINRA, and the SEC. Learn more about the key market data policies you need to know as a data consumer.

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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)
```

Insider Transactions Trending

This API returns the latest and historical insider transactions made by key stakeholders (e.g., founders, executives, board members, etc.) of a specific company.

API Parameters

Required: `function`

The function of your choice. In this case, `function=INSIDER_TRANSACTIONS`

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.

Example (click for JSON output)

`https://www.alphavantage.co/query?function=INSIDER_TRANSACTIONS&symbol=IBM&apikey=demo`

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
import requests

# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-ke
y
url = 'https://www.alphavantage.co/query?function=INSIDER_TRANSACTIONS&symbol=IBM&apikey=demo'
r = requests.get(url)
data = r.json()

print(data)
```

Advanced 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 normalize 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 normalize 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 normalize 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.

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

```python
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

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 normalize 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 normalize 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 normalize 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.

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

```python
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

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

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.

Example (click for JSON output)

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

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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)
```

ETF Profile & Holdings

This API returns key ETF metrics (e.g., net assets, expense ratio, and turnover), along with the corresponding ETF holdings / constituents with allocation by asset types and sectors.

API Parameters

Required: `function`

The function of your choice. In this case, `function=ETF_PROFILE`

Required: `symbol`

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

Required: `apikey`

Your API key. Claim your free API key here.

Example (click for JSON output)

https://www.alphavantage.co/query?function=ETF_PROFILE&symbol=QQQ&apikey=demo

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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=ETF_PROFILE&symbol=QQQ&apikey=demo'
r = requests.get(url)
data = r.json()

print(data)
```

Corporate Action - Dividends Trending

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.

Example (click for JSON output)

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

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.

Example (click for JSON output)

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

This API returns the annual and quarterly income statements for the company of interest, with normalized fields mapped to GAAP and IFRS taxonomies 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.

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

`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-ke
y
url = 'https://www.alphavantage.co/query?function=INCOME_STATEMENT&symbol=IBM&apikey=demo'
r = requests.get(url)
data = r.json()

print(data)
```

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 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.

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

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

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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

This API returns the annual and quarterly cash flow for the company of interest, with normalized fields mapped to GAAP and IFRS taxonomies 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.

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

`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-ke
y
url = 'https://www.alphavantage.co/query?function=CASH_FLOW&symbol=IBM&apikey=demo'
r = requests.get(url)
data = r.json()

print(data)
```

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.

Examples (click for JSON output)

`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-ke
y
```

```
url = 'https://www.alphavantage.co/query?function=EARNINGS&symbol=IBM&apikey=demo'
r = requests.get(url)
data = r.json()

print(data)
```

Listing & Delisting 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.

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`

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`

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-ke
y
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

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.

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`

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

Language-specific guides

Python NodeJS PHP C#/.NET Other

```python
import csv
import requests

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

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.

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

Language-specific guides

Python NodeJS PHP C#/.NET Other

```python
import csv
import requests
```

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

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

CURRENCY_EXCHANGE_RATE Trending

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 or digital/crypto currency. 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 or digital/crypto currency. For example: `to_currency=USD` or `to_currency=BTC`.

Required: `apikey`

Your API key. Claim your free API key here.

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

```python
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

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. For example: `from_symbol=EUR`

Required: `to_symbol`

A three-letter symbol from the forex 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.

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&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

>> Tip: this is a premium API function. Subscribe to a premium membership plan to instantly unlock all premium APIs.

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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

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. For example: `from_symbol=EUR`

Required: `to_symbol`

A three-letter symbol from the forex 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.

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&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

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-ke
y
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

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. For example: `from_symbol=EUR`

Required: `to_symbol`

A three-letter symbol from the forex 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.

Examples (click for JSON output)

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

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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

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. For example: `from_symbol=EUR`

Required: `to_symbol`

A three-letter symbol from the forex 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.

Examples (click for JSON output)

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

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-ke
y
```

```
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

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

CURRENCY_EXCHANGE_RATE Trending

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 or digital/crypto currency. 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 or digital/crypto currency. For example: `to_currency=USD` or `to_currency=BTC`.

Required: `apikey`

Your API key. Claim your free API key here.

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

```python
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

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. For example: `symbol=ETH`.

Required: `market`

The exchange market of your choice. It can be any of the market in the market 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.

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&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

>> Tip: this is a premium API function. Subscribe to a premium membership plan to instantly unlock all premium APIs.

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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

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. For example: `symbol=BTC`.

Required: `market`

The exchange market of your choice. It can be any of the market in the market list. For example: `market=EUR`.

Required: `apikey`

Your API key. Claim your free API key here.

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

```python
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

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. For example: `symbol=BTC`.

Required: `market`

The exchange market of your choice. It can be any of the market in the market list. For example: `market=EUR`.

Required: `apikey`

Your API key. Claim your free API key here.

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

```python
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

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. For

example: `symbol=BTC`.

Required: `market`

The exchange market of your choice. It can be any of the market in the market list. For example: `market=EUR`.

Required: `apikey`

Your API key. Claim your free API key here.

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

```python
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

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

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.

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.

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

```python
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

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.

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.

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

```python
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

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.

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.

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

```python
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

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

Source: International Monetary Fund (IMF Terms of Use), 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.

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.

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

```python
import requests

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

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

Source: International Monetary Fund (IMF Terms of Use), 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.

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.

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

```python
import requests

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

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

Source: International Monetary Fund (IMF Terms of Use), 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.

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.

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

```python
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

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

Source: International Monetary Fund (IMF Terms of Use), 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.

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.

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

```python
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

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

Source: International Monetary Fund (IMF Terms of Use), 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.

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, . Strings and are accepted with the following specifications:

`datatype=json` `json` `csv` `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.

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

```python
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

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

Source: International Monetary Fund (IMF Terms of Use), 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.

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.

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

```python
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

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

Source: International Monetary Fund (IMF Terms of Use), 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.

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.

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

```python
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

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), 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.

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.

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

```python
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

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

REAL_GDP Trending

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.

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.

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

```python
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

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.

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.

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

```python
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

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.

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.

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

```python
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

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.

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.

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

```python
import requests

# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-ke
y
url = 'https://www.alphavantage.co/query?function=FEDERAL_FUNDS_RATE&interval=monthly&apikey=demo'
r = requests.get(url)
data = r.json()

print(data)
```

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.

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.

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

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.

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.

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-ke
y
url = 'https://www.alphavantage.co/query?function=INFLATION&apikey=demo'
r = requests.get(url)
data = r.json()

print(data)
```

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.

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.

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-ke
y
url = 'https://www.alphavantage.co/query?function=RETAIL_SALES&apikey=demo'
r = requests.get(url)
data = r.json()
```

```
print(data)
```

DURABLES

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.

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.

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-ke
y
url = 'https://www.alphavantage.co/query?function=DURABLES&apikey=demo'
r = requests.get(url)
data = r.json()

print(data)
```

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).

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.

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.

Examples (click for JSON output)

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

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
import requests

# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-ke
y
url = 'https://www.alphavantage.co/query?function=UNEMPLOYMENT&apikey=demo'
r = requests.get(url)
data = r.json()

print(data)
```

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.

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.

Examples (click for JSON output)

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

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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

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

This API returns the simple moving average (SMA) values. See also: SMA explainer and mathematical reference.

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.

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`

Forex (FX) or cryptocurrency pair:

`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

```python
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

This API returns the exponential moving average (EMA) values. See also: EMA explainer and mathematical reference.

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.

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

Forex (FX) or cryptocurrency pair:

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

```python
import requests

# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-ke
y
url = 'https://www.alphavantage.co/query?function=EMA&symbol=IBM&interval=weekly&time_period=10&seri
es_type=open&apikey=demo'
r = requests.get(url)
data = r.json()

print(data)
```

WMA

This API returns the weighted moving average (WMA) values. See also: mathematical reference.

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.

Examples (click for JSON output)

`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

This API returns the double exponential moving average (DEMA) values. See also: Investopedia article and mathematical reference.

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.

Examples (click for JSON output)

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

```python
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

This API returns the triple exponential moving average (TEMA) values. See also: mathematical reference.

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.

Examples (click for JSON output)

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

```python
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

This API returns the triangular moving average (TRIMA) values. See also: mathematical reference.

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.

Examples (click for JSON output)

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-ke
y
```

```
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

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.

Examples (click for JSON output)

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

```python
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

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.

Example (click for JSON output)

`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

```python
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

This API returns the volume weighted average price (VWAP) for intraday time series. See also: Investopedia article.

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.

Examples (click for JSON output)

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

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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

This API returns the triple exponential moving average (T3) values. See also: mathematical reference.

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.

Examples (click for JSON output)

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

```python
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

This API returns the moving average convergence / divergence (MACD) values. See also: Investopedia article and mathematical reference.

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.

Examples (click for JSON output)

Equity:

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

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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

This API returns the moving average convergence / divergence values with controllable moving average type. See also: Investopedia article and mathematical reference.

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.

Examples (click for JSON output)

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

```python
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

This API returns the stochastic oscillator (STOCH) values. See also: Investopedia article and mathematical reference.

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.

Examples (click for JSON output)

Equity:

`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`

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
import requests

# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-ke
y
url = 'https://www.alphavantage.co/query?function=STOCH&symbol=IBM&interval=daily&apikey=demo'
r = requests.get(url)
data = r.json()

print(data)
```

STOCHF

This API returns the stochastic fast (STOCHF) values. See also: Investopedia article and mathematical reference.

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.

Examples (click for JSON output)

`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-ke
y
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

This API returns the relative strength index (RSI) values. See also: RSI explainer and mathematical reference.

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.

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`

Forex (FX) or cryptocurrency pair:

`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

```python
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

This API returns the stochastic relative strength index (STOCHRSI) values. See also: mathematical reference.

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.

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

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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

This API returns the Williams' %R (WILLR) values. See also: mathematical reference.

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.

Example (click for JSON output)

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

```python
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

This API returns the average directional movement index (ADX) values. See also: Investopedia article and mathematical reference.

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.

Example (click for JSON output)

Equity:

`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`

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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

This API returns the average directional movement index rating (ADXR) values. See also: mathematical reference.

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.

Example (click for JSON output)

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

```python
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

This API returns the absolute price oscillator (APO) values. See also: mathematical reference.

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.

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`

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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

This API returns the percentage price oscillator (PPO) values. See also: Investopedia article and mathematical reference.

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.

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

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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

This API returns the momentum (MOM) values. See also: Investopedia article and mathematical reference.

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.

Example (click for JSON output)

`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

```python
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

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.

Example (click for JSON output)

`https://www.alphavantage.co/query?function=BOP&symbol=IBM&interval=daily&apikey=demo`

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
import requests

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

This API returns the commodity channel index (CCI) values. See also: Investopedia article and mathematical reference.

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.

Example (click for JSON output)

Equity:

`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`

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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

This API returns the Chande momentum oscillator (CMO) values. See also: mathematical reference.

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.

Example (click for JSON output)

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-ke
y
url = 'https://www.alphavantage.co/query?function=CMO&symbol=IBM&interval=weekly&time_period=10&seri
```

```
es_type=close&apikey=demo'
r = requests.get(url)
data = r.json()

print(data)
```

ROC

This API returns the rate of change (ROC) values. See also: Investopedia article.

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.

Example (click for JSON output)

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

```python
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

This API returns the rate of change ratio (ROCR) values. See also: Investopedia article.

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.

Example (click for JSON output)

`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

```python
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

This API returns the Aroon (AROON) values. See also: Investopedia article and mathematical reference.

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.

Example (click for JSON output)

Equity:

`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`

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
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

This API returns the Aroon oscillator (AROONOSC) values. See also: mathematical reference.

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.

Example (click for JSON output)

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

```python
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

This API returns the money flow index (MFI) values. See also: Investopedia article and mathematical reference.

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.

Example (click for JSON output)

`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

```python
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

This API returns the 1-day rate of change of a triple smooth exponential moving average (TRIX) values. See also: Investopedia article and mathematical reference.

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.

Example (click for JSON output)

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

```python
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

This API returns the ultimate oscillator (ULTOSC) values. See also: mathematical reference.

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.

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&apikey=demo

Language-specific guides

Python NodeJS PHP C#/.NET Other

```python
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

This API returns the directional movement index (DX) values. See also: Investopedia article and mathematical reference.

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.

Example (click for JSON output)

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

```python
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

This API returns the minus directional indicator (MINUS_DI) values. See also: Investopedia article and mathematical reference.

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.

Example (click for JSON output)

`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

```python
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

This API returns the plus directional indicator (PLUS_DI) values. See also: Investopedia article and mathematical reference.

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.

Example (click for JSON output)

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

```python
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

This API returns the minus directional movement (MINUS_DM) values. See also: Investopedia article

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.

Example (click for JSON output)

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

```python
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

This API returns the plus directional movement (PLUS_DM) values. See also: Investopedia article

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.

Example (click for JSON output)

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

```python
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

This API returns the Bollinger bands (BBANDS) values. See also: Investopedia article and mathematical reference.

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.

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`

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`

Language-specific guides

Python NodeJS PHP C#/.NET Other

```python
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

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.

Example (click for JSON output)

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

```python
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

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.

Example (click for JSON output)

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

```python
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

This API returns the parabolic SAR (SAR) values. See also: Investopedia article and mathematical reference.

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.

Example (click for JSON output)

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-ke
```

```
y
url = 'https://www.alphavantage.co/query?function=SAR&symbol=IBM&interval=weekly&acceleration=0.05&m
aximum=0.25&apikey=demo'
r = requests.get(url)
data = r.json()

print(data)
```

TRANGE

This API returns the true range (TRANGE) values. See also: mathematical reference

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.

Example (click for JSON output)

`https://www.alphavantage.co/query?function=TRANGE&symbol=IBM&interval=daily&apikey=demo`

Language-specific guides

Python NodeJS PHP C#/.NET Other

```python
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

This API returns the average true range (ATR) values. See also: Investopedia article and mathematical reference

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.

Example (click for JSON output)

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

```python
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

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.

Example (click for JSON output)

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

```python
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

This API returns the Chaikin A/D line (AD) values. See also: Investopedia article and mathematical reference.

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.

Example (click for JSON output)

https://www.alphavantage.co/query?function=AD&symbol=IBM&interval=daily&apikey=demo

Language-specific guides

Python | NodeJS | PHP | C#/.NET | Other

```python
import requests

# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-ke
y
url = 'https://www.alphavantage.co/query?function=AD&symbol=IBM&interval=daily&apikey=demo'
r = requests.get(url)
data = r.json()

print(data)
```

ADOSC

This API returns the Chaikin A/D oscillator (ADOSC) values. See also: Investopedia article and mathematical reference.

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.

Example(click for JSON output)

`https://www.alphavantage.co/query?function=ADOSC&symbol=IBM&interval=daily&fastperiod=5&apikey=demo`

Language-specific guides

Python NodeJS PHP C#/.NET Other

```python
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

This API returns the on balance volume (OBV) values. See also: Investopedia article and mathematical reference.

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.

Example (click for JSON output)

https://www.alphavantage.co/query?function=OBV&symbol=IBM&interval=weekly&apikey=demo

Language-specific guides

Python NodeJS PHP C#/.NET Other

```python
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

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.

Example (click for JSON output)

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

```python
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

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.

Example (click for JSON output)

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

```python
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

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.

Example (click for JSON output)

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

```python
import requests

# replace the "demo" apikey below with your own key from https://www.alphavantage.co/support/#api-ke
y
url = 'https://www.alphavantage.co/query?function=HT_TRENDMODE&symbol=IBM&interval=weekly&series_typ
e=close&apikey=demo'
r = requests.get(url)
```

```
data = r.json()

print(data)
```

HT_DCPERIOD

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.

Example (click for JSON output)

`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

```python
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

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.

Example (click for JSON output)

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

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.

Example (click for JSON output)

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

```python
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)
```