polygon.io

Documentation

Stocks Options Indices Forex Crypto

Key: Default

Getting Started

Stocks API Documentation

The Polygon.io Stocks API provides REST endpoints that let you query the latest market data from all US stock exchanges. You can also find data on company financials, stock market holidays, corporate actions, and more.

Authentication

Pass your API key in the query string like follows:

```
https://api.polygon.io/v2/aggs/ticker/AAPL/range/1/day/2023-01-09/2023-01-09?apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

Alternatively, you can add an Authorization header to the request with your API Key as the token in the following form:

```
Authorization: Bearer 1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

Usage

Many of Polygon.io's REST endpoints allow you to extend query parameters with inequalities like `date.lt=2023-01-01` (less than) and `date.gte=2023-01-01` (greater than or equal to) to search ranges of values. You can also use the field name without any extension to query for exact equality. Fields that support extensions will have an "Additional filter parameters" dropdown beneath them in the docs that detail the supported extensions for that parameter.

Response Types

By default, all endpoints return a JSON response. Users with Stocks Starter plan and above can request a CSV response by including `'Accept': 'text/csv'` as a request parameter.

Your Plan

Stocks Basic

- 5 API Calls / Minute
- End of Day Data
- 2 Years Historical Data

Upgrade

Client Libraries

Python
client-python

Go
client-go

Javascript
client-js

PHP
client-php

Kotlin
client-jvm

Aggregates (Bars)

GET /v2/aggs/ticker/{stocksTicker}/range/{multiplier}/{timespan}/{from}/{to}

Get aggregate bars for a stock over a given date range in custom time window sizes.

For example, if timespan = ‘minute’ and multiplier = ‘5’ then 5-minute bars will be returned.

Parameters

stocksTicker*	AAPL

Specify a case-sensitive ticker symbol. For example, AAPL represents Apple Inc.

multiplier*	1

The size of the timespan multiplier.

timespan*	day

The size of the time window.

from*	2023-01-09

The start of the aggregate time window. Either a date with the format YYYY-MM-DD or a millisecond timestamp.

to*	2023-01-09

The end of the aggregate time window. Either a date with the format YYYY-MM-DD or a millisecond timestamp.

adjusted	true

Whether or not the results are adjusted for splits. By default, results are adjusted. Set this to false to get results that are NOT adjusted for splits.

sort	asc

Sort the results by timestamp. `asc` will return results in ascending order (oldest at the top), `desc` will return results in descending order (newest at the top).

limit	120

Limits the number of base aggregates queried to create the aggregate results. Max 50000 and Default 5000. Read more about how limit is used to calculate aggregate results in our article on Aggregate Data API Improvements.

```
https://api.polygon.io/v2/aggs/ticker/AAPL/range/1/day/2023-01-09/2023-01-09?adju
sted=true&sort=asc&limit=120&apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

JSON

Run Query

Response Attributes

`ticker*` string

The exchange symbol that this item is traded under.

`adjusted*` boolean

Whether or not this response was adjusted for splits.

`queryCount*` integer

The number of aggregates (minute or day) used to generate the response.

`request_id*` string

A request id assigned by the server.

`resultsCount*` integer

The total number of results for this request.

`status*` string

The status of this request's response.

`results` array

`c*` number

The close price for the symbol in the given time period.

`h*` number

The highest price for the symbol in the given time period.

`l*` number

The lowest price for the symbol in the given time period.

`n` integer

The number of transactions in the aggregate window.

`o*` number

The open price for the symbol in the given time period.

`otc` boolean

Whether or not this aggregate is for an OTC ticker. This field will be left off if false.

`t*` integer

The Unix Msec timestamp for the start of the aggregate window.

`v*` number

The trading volume of the symbol in the given time period.

`vw` number

The volume weighted average price.

`next_url` string

If present, this value can be used to fetch the next page of data.

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "adjusted": true,
  "next_url": "https://api.polygon.io/v2/aggs/ticker/AAPL/range/1/day/1578114000000/2020-
  "queryCount": 2,
  "request_id": "6a7e466379af0a71039d60cc78e72282",
  "results": [
    {
      "c": 75.0875,
      "h": 75.15,
      "l": 73.7975,
      "n": 1,
      "o": 74.06,
```

```
      "t": 1577941200000,
      "v": 135647456,
      "vw": 74.6099
    },
    {
      "c": 74.3575,
      "h": 75.145,
      "l": 74.125,
      "n": 1,
      "o": 74.2875,
      "t": 1578027600000,
      "v": 146535512,
      "vw": 74.7026
    }
  ],
  "resultsCount": 2,
  "status": "OK",
  "ticker": "AAPL"
}
```


Grouped Daily (Bars)

GET /v2/aggs/grouped/locale/us/market/stocks/{date}

Get the daily open, high, low, and close (OHLC) for the entire stocks/equities markets.

Parameters

date*	2023-01-09

The beginning date for the aggregate window.

adjusted	true

Whether or not the results are adjusted for splits. By default, results are adjusted. Set this to false to get results that are NOT adjusted for splits.

include_otc	

Include OTC securities in the response. Default is false (don't include OTC securities).

```
https://api.polygon.io/v2/aggs/grouped/locale/us/market/stocks/2023-01-09?adjusted=true&apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

JSON

Run Query

Response Attributes

`adjusted`* boolean

Whether or not this response was adjusted for splits.

`queryCount`* integer

The number of aggregates (minute or day) used to generate the response.

`request_id`* string

A request id assigned by the server.

`resultsCount`* integer

The total number of results for this request.

`status`* string

The status of this request's response.

`results` array

`T`* string

The exchange symbol that this item is traded under.

`c`* number

The close price for the symbol in the given time period.

`h`* number

The highest price for the symbol in the given time period.

`l`* number

The lowest price for the symbol in the given time period.

`n` integer

The number of transactions in the aggregate window.

`o*` number

The open price for the symbol in the given time period.

`otc` boolean

Whether or not this aggregate is for an OTC ticker. This field will be left off if false.

`t*` integer

The Unix Msec timestamp for the end of the aggregate window.

`v*` number

The trading volume of the symbol in the given time period.

`vw` number

The volume weighted average price.

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "adjusted": true,
  "queryCount": 3,
  "results": [
    {
      "T": "KIMpL",
      "c": 25.9102,
      "h": 26.25,
      "l": 25.91,
      "n": 74,
      "o": 26.07,
      "t": 1602705600000,
      "v": 4369,
      "vw": 26.0407
    },
    {
      "T": "TANH",
      "c": 23.4,
      "h": 24.763,
      "l": 22.65,
      "n": 1096,
      "o": 24.5,
      "t": 1602705600000,
      "v": 25933.6,
      "vw": 23.493
```

```
    },
    {
      "T": "VSAT",
      "c": 34.24,
      "h": 35.47,
      "l": 34.21,
      "n": 4966,
      "o": 34.9,
      "t": 1602705600000,
      "v": 312583,
      "vw": 34.4736
    }
  ],
  "resultsCount": 3,
  "status": "OK"
}
```


Daily Open/Close

GET /v1/open-close/{stocksTicker}/{date}

Get the open, close and afterhours prices of a stock symbol on a certain date.

Parameters

stocksTicker*	AAPL

Specify a case-sensitive ticker symbol. For example, AAPL represents Apple Inc.

date*	

The date of the requested open/close in the format YYYY-MM-DD.

adjusted	true

Whether or not the results are adjusted for splits. By default, results are adjusted. Set this to false to get results that are NOT adjusted for splits.

```
https://api.polygon.io/v1/open-close/AAPL/2023-01-09?adjusted=true&apiKey=1RQBxJj
R6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

JSON

Run Query

Response Attributes

`afterHours` number

The close price of the ticker symbol in after hours trading.

`close*` number

The close price for the symbol in the given time period.

`from*` string

The requested date.

`high*` number

The highest price for the symbol in the given time period.

`low*` number

The lowest price for the symbol in the given time period.

`open*` number

The open price for the symbol in the given time period.

`otc` boolean

Whether or not this aggregate is for an OTC ticker. This field will be left off if false.

`preMarket` integer

The open price of the ticker symbol in pre-market trading.

`status*` string

The status of this request's response.

`symbol*` string

The exchange symbol that this item is traded under.

`volume*` number

The trading volume of the symbol in the given time period.

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "afterHours": 322.1,
  "close": 325.12,
  "from": "2023-01-09",
  "high": 326.2,
  "low": 322.3,
  "open": 324.66,
  "preMarket": 324.5,
  "status": "OK",
  "symbol": "AAPL",
  "volume": 26122646
}
```


Previous Close

GET /v2/aggs/ticker/{stocksTicker}/prev

Get the previous day's open, high, low, and close (OHLC) for the specified stock ticker.

Parameters

stocksTicker*	AAPL

Specify a case-sensitive ticker symbol. For example, AAPL represents Apple Inc.

adjusted	true

Whether or not the results are adjusted for splits. By default, results are adjusted. Set this to false to get results that are NOT adjusted for splits.

```
https://api.polygon.io/v2/aggs/ticker/AAPL/prev?adjusted=true&apiKey=1RQBxJjR6uKq
zsfPUpWtrHCSsFMOwSjj
```

Copy

JSON

Run Query

Response Attributes

`ticker`* string

The exchange symbol that this item is traded under.

`adjusted`* boolean

Whether or not this response was adjusted for splits.

`queryCount`* integer

The number of aggregates (minute or day) used to generate the response.

`request_id`* string

A request id assigned by the server.

`resultsCount`* integer

The total number of results for this request.

`status`* string

The status of this request's response.

`results` array

- `c`* number

  The close price for the symbol in the given time period.

- `h`* number

  The highest price for the symbol in the given time period.

- `l`* number

  The lowest price for the symbol in the given time period.

- `n` integer

  The number of transactions in the aggregate window.

- `o`* number

  The open price for the symbol in the given time period.

- `t`* integer

  The Unix Msec timestamp for the start of the aggregate window.

- `v`* number

  The trading volume of the symbol in the given time period.

vw number

The volume weighted average price.

Was this helpful?

Help us improve

Yes No

Response Object


```json
{
  "adjusted": true,
  "queryCount": 1,
  "request_id": "6a7e466379af0a71039d60cc78e72282",
  "results": [
    {
      "T": "AAPL",
      "c": 115.97,
      "h": 117.59,
      "l": 114.13,
      "o": 115.55,
      "t": 1605042000000,
      "v": 131704427,
      "vw": 116.3058
    }
  ],
  "resultsCount": 1,
  "status": "OK",
  "ticker": "AAPL"
}
```


Trades

GET /v3/trades/{stockTicker}

Get trades for a ticker symbol in a given time range.

Requires a "Stocks Developer" subscription Upgrade

Parameters

`stockTicker*`	AAPL

The ticker symbol to get trades for.

`timestamp`	

Query by trade timestamp. Either a date with the format YYYY-MM-DD or a nanosecond timestamp.

Additional filter parameters

`order`	

Order results based on the `sort` field.

`limit`	

Limit the number of results returned, default is 10 and max is 50000.

`sort`	

Sort field used for ordering.

```
https://api.polygon.io/v3/trades/AAPL?apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

JSON

Run Query

Response Attributes

`next_url` string

If present, this value can be used to fetch the next page of data.

`results` array

`conditions` array [integer]

A list of condition codes.

`correction` integer

The trade correction indicator.

`exchange*` integer

The exchange ID. See Exchanges for Polygon.io's mapping of exchange IDs.

`id*` string

The Trade ID which uniquely identifies a trade. These are unique per combination of ticker, exchange, and TRF. For example: A trade for AAPL executed on NYSE and a trade for AAPL executed on NASDAQ could potentially have the same Trade ID.

`participant_timestamp`* integer

The nanosecond accuracy Participant/Exchange Unix Timestamp. This is the timestamp of when the trade was actually generated at the exchange.

`price`* number

The price of the trade. This is the actual dollar value per whole share of this trade. A trade of 100 shares with a price of $2.00 would be worth a total dollar value of $200.00.

`sequence_number`* integer

The sequence number represents the sequence in which trade events happened. These are increasing and unique per ticker symbol, but will not always be sequential (e.g., 1, 2, 6, 9, 10, 11). Values reset after each trading session/day.

`sip_timestamp`* integer

The nanosecond accuracy SIP Unix Timestamp. This is the timestamp of when the SIP received this trade from the exchange which produced it.

`size`* number

The size of a trade (also known as volume).

`tape` integer

There are 3 tapes which define which exchange the ticker is listed on. These are integers in our objects which represent the letter of the alphabet. Eg: 1 = A, 2 = B, 3 = C.
Tape A is NYSE listed securities
Tape B is NYSE ARCA / NYSE American
Tape C is NASDAQ

`trf_id` integer

The ID for the Trade Reporting Facility where the trade took place.

`trf_timestamp` integer

The nanosecond accuracy TRF (Trade Reporting Facility) Unix Timestamp. This is the timestamp of when the trade reporting facility received this trade.

`status`* string

The status of this request's response.

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "next_url": "https://api.polygon.io/v3/trades/AAPL?cursor=YWN0aXZlPXRydWUmZGF0ZT0yMDIxL
  "request_id": "a47d1beb8c11b6ae897ab76cdbbf35a3",
  "results": [
    {
      "conditions": [
        12,
        41
      ],
      "exchange": 11,
      "id": "1",
      "participant_timestamp": 1517562000015577000,
      "price": 171.55,
      "sequence_number": 1063,
      "sip_timestamp": 1517562000016036600,
      "size": 100,
      "tape": 3
    },
    {
      "conditions": [
        12,
        41
      ],
      "exchange": 11,
      "id": "2",
      "participant_timestamp": 1517562000015577600,
      "price": 171.55,
      "sequence_number": 1064,
      "sip_timestamp": 1517562000016038100,
      "size": 100,
      "tape": 3
    }
  ],
  "status": "OK"
}
```


Last Trade

GET /v2/last/trade/{stocksTicker}

Get the most recent trade for a given stock.

Requires a "Stocks Developer" subscription

Upgrade

Parameters

stocksTicker*	AAPL

The ticker symbol of the stock/equity.

`https://api.polygon.io/v2/last/trade/AAPL?apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj` Copy

JSON

Run Query

Response Attributes

`request_id*` string

A request id assigned by the server.

`results` object

`T*` string

The exchange symbol that this item is traded under.

`c` array [integer]

A list of condition codes.

`e` integer

The trade correction indicator.

`f` integer

The nanosecond accuracy TRF(Trade Reporting Facility) Unix Timestamp. This is the timestamp of when the trade reporting facility received this message.

`i*` string

The Trade ID which uniquely identifies a trade. These are unique per combination of ticker, exchange, and TRF. For example: A trade for AAPL executed on NYSE and a trade for AAPL executed on NASDAQ could potentially have the same Trade ID.

`p*` number

The price of the trade. This is the actual dollar value per whole share of this trade. A trade of 100 shares with a price of $2.00 would be worth a total dollar value of $200.00.

`q`* integer

The sequence number represents the sequence in which message events happened. These are increasing and unique per ticker symbol, but will not always be sequential (e.g., 1, 2, 6, 9, 10, 11).

`r` integer

The ID for the Trade Reporting Facility where the trade took place.

`s` number

The size of a trade (also known as volume).

`t`* integer

The nanosecond accuracy SIP Unix Timestamp. This is the timestamp of when the SIP received this message from the exchange which produced it.

`x`* integer

The exchange ID. See Exchanges for Polygon.io's mapping of exchange IDs.

`y`* integer

The nanosecond accuracy Participant/Exchange Unix Timestamp. This is the timestamp of when the quote was actually generated at the exchange.

`z` integer

There are 3 tapes which define which exchange the ticker is listed on. These are integers in our objects which represent the letter of the alphabet. Eg: 1 = A, 2 = B, 3 = C.
Tape A is NYSE listed securities
Tape B is NYSE ARCA / NYSE American
Tape C is NASDAQ

`status`* string

The status of this request's response.

Was this helpful?
Help us improve

Yes No

Response Object


```
{
  "request_id": "f05562305bd26ced64b98ed68b3c5d96",
  "results": {
```

```
      "T": "AAPL",
      "c": [
        37
      ],
      "f": 1617901342969796400,
      "i": "118749",
      "p": 129.8473,
      "q": 3135876,
      "r": 202,
      "s": 25,
      "t": 1617901342969834000,
      "x": 4,
      "y": 1617901342968000000,
      "z": 3
    },
    "status": "OK"
  }
```


Quotes (NBBO)

GET /v3/quotes/{stockTicker}

Get NBBO quotes for a ticker symbol in a given time range.

Requires a "Stocks Advanced" subscription

Upgrade

Parameters

stockTicker*	AAPL

The ticker symbol to get quotes for.

timestamp	

Query by timestamp. Either a date with the format YYYY-MM-DD or a nanosecond timestamp.

Additional filter parameters

order	

Order results based on the `sort` field.

limit

Limit the number of results returned, default is 10 and max is 50000.

sort

Sort field used for ordering.

```
https://api.polygon.io/v3/quotes/AAPL?apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

JSON

Run Query

Response Attributes

`next_url` string

If present, this value can be used to fetch the next page of data.

`results` array

`ask_exchange` integer

The ask exchange ID

`ask_price` number

The ask price.

`ask_size` number

The ask size. This represents the number of round lot orders at the given ask price. The normal round lot size is 100 shares. An ask size of 2 means there are 200 shares available to purchase at the given ask price.

`bid_exchange` integer

The bid exchange ID

`bid_price` number

The bid price.

`bid_size` number

The bid size. This represents the number of round lot orders at the given bid price. The normal round lot size is 100 shares. A bid size of 2 means there are 200 shares for purchase at the given bid price.

`conditions` array [integer]

A list of condition codes.

`indicators` array [integer]

A list of indicator codes.

`participant_timestamp`* integer

The nanosecond accuracy Participant/Exchange Unix Timestamp. This is the timestamp of when the quote was actually generated at the exchange.

`sequence_number`* integer

The sequence number represents the sequence in which quote events happened. These are increasing and unique per ticker symbol, but will not always be sequential (e.g., 1, 2, 6, 9, 10, 11). Values reset after each trading session/day.

`sip_timestamp`* integer

The nanosecond accuracy SIP Unix Timestamp. This is the timestamp of when the SIP received this quote from the exchange which produced it.

`tape` integer

There are 3 tapes which define which exchange the ticker is listed on. These are integers in our objects which represent the letter of the alphabet. Eg: 1 = A, 2 = B, 3 = C.
Tape A is NYSE listed securities
Tape B is NYSE ARCA / NYSE American
Tape C is NASDAQ

`trf_timestamp` integer

The nanosecond accuracy TRF (Trade Reporting Facility) Unix Timestamp. This is the timestamp of when the trade reporting facility received this quote.

`status`* string

The status of this request's response.

Was this helpful?
Help us improve

Yes No

Response Object

```
{
  "next_url": "https://api.polygon.io/v3/quotes/AAPL?cursor=YWN0aXZlPXRydWUmZGF0ZT0yMDIxL
  "request_id": "a47d1beb8c11b6ae897ab76cdbbf35a3",
  "results": [
    {
      "ask_exchange": 0,
      "ask_price": 0,
```

```
      "ask_size": 0,
      "bid_exchange": 11,
      "bid_price": 102.7,
      "bid_size": 60,
      "conditions": [
        1
      ],
      "participant_timestamp": 1517562000065321200,
      "sequence_number": 2060,
      "sip_timestamp": 1517562000065700400,
      "tape": 3
    },
    {
      "ask_exchange": 0,
      "ask_price": 0,
      "ask_size": 0,
      "bid_exchange": 11,
      "bid_price": 170,
      "bid_size": 2,
      "conditions": [
        1
      ],
      "participant_timestamp": 1517562000065408300,
      "sequence_number": 2061,
      "sip_timestamp": 1517562000065791500,
      "tape": 3
    }
  ],
  "status": "OK"
}
```


Last Quote

GET /v2/last/nbbo/{stocksTicker}

Get the most recent NBBO (Quote) tick for a given stock.

Requires a "Stocks Advanced" subscription

Upgrade

Parameters

stocksTicker*	AAPL

The ticker symbol of the stock/equity.

```
https://api.polygon.io/v2/last/nbbo/AAPL?apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

JSON

Run Query

Response Attributes

`request_id`* string

A request id assigned by the server.

`results` object

`P` number

The ask price.

`S` integer

The ask size. This represents the number of round lot orders at the given ask price. The normal round lot size is 100 shares. An ask size of 2 means there are 200 shares available to purchase at the given ask price.

`T`* string

The exchange symbol that this item is traded under.

`X` integer

The exchange ID. See Exchanges for Polygon.io's mapping of exchange IDs.

`c` array [integer]

A list of condition codes.

`f` integer

The nanosecond accuracy TRF(Trade Reporting Facility) Unix Timestamp. This is the timestamp of when the trade reporting facility received this message.

`i` array [integer]

A list of indicator codes.

`p` number

The bid price.

q* integer

The sequence number represents the sequence in which message events happened. These are increasing and unique per ticker symbol, but will not always be sequential (e.g., 1, 2, 6, 9, 10, 11).

s integer

The bid size. This represents the number of round lot orders at the given bid price. The normal round lot size is 100 shares. A bid size of 2 means there are 200 shares for purchase at the given bid price.

t* integer

The nanosecond accuracy SIP Unix Timestamp. This is the timestamp of when the SIP received this message from the exchange which produced it.

x integer

The exchange ID. See Exchanges for Polygon.io's mapping of exchange IDs.

y* integer

The nanosecond accuracy Participant/Exchange Unix Timestamp. This is the timestamp of when the quote was actually generated at the exchange.

z integer

There are 3 tapes which define which exchange the ticker is listed on. These are integers in our objects which represent the letter of the alphabet. Eg: 1 = A, 2 = B, 3 = C.
Tape A is NYSE listed securities
Tape B is NYSE ARCA / NYSE American
Tape C is NASDAQ

status* string

The status of this request's response.

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "request_id": "b84e24636301f19f88e0dfbf9a45ed5c",
  "results": {
    "P": 127.98,
    "S": 7,
    "T": "AAPL",
    "X": 19,
    "p": 127.96,
```

```
      "q": 83480742,
      "s": 1,
      "t": 1617827221349730300,
      "x": 11,
      "y": 1617827221349366000,
      "z": 3
    },
    "status": "OK"
  }
```


All Tickers

GET /v2/snapshot/locale/us/markets/stocks/tickers

Get the most up-to-date market data for all traded stock symbols.

Note: Snapshot data is cleared at 3:30am EST and gets populated as data is received from the exchanges. This can happen as early as 4am EST.

Requires a "Stocks Starter" subscription Upgrade

Parameters

tickers

A case-sensitive comma separated list of tickers to get snapshots for. For example, AAPL,TSLA,GOOG. Empty string defaults to querying all tickers.

include_otc

Include OTC securities in the response. Default is false (don't include OTC securities).

```
https://api.polygon.io/v2/snapshot/locale/us/markets/stocks/tickers?apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

JSON Run Query

Response Attributes

`count` integer

The total number of results for this request.

`status*` string

The status of this request's response.

`tickers` array

`day` object

The most recent daily bar for this ticker.

`c*` number

The close price for the symbol in the given time period.

`h*` number

The highest price for the symbol in the given time period.

`l*` number

The lowest price for the symbol in the given time period.

`o*` number

The open price for the symbol in the given time period.

`otc` boolean

Whether or not this aggregate is for an OTC ticker. This field will be left off if false.

`v*` number

The trading volume of the symbol in the given time period.

`vw*` number

The volume weighted average price.

`fmv` number

Fair market value is only available on Business plans. It is our proprietary algorithm to generate a real-time, accurate, fair market value of a tradable security. For more information, contact us.

`lastQuote` object

The most recent quote for this ticker. This is only returned if your current plan includes quotes.

`P*` number

The ask price.

`S*` integer

The ask size in lots.

`p*` number

The bid price.

`s*` integer

The bid size in lots.

`t*` integer

The nanosecond accuracy SIP Unix Timestamp. This is the timestamp of when the SIP received this message from the exchange which produced it.

`lastTrade` object

The most recent trade for this ticker. This is only returned if your current plan includes trades.

`c*` array [integer]

The trade conditions.

`i*` string

The Trade ID which uniquely identifies a trade. These are unique per combination of ticker, exchange, and TRF. For example: A trade for AAPL executed on NYSE and a trade for AAPL executed on NASDAQ could potentially have the same Trade ID.

`p*` number

The price of the trade. This is the actual dollar value per whole share of this trade. A trade of 100 shares with a price of $2.00 would be worth a total dollar value of $200.00.

`s*` integer

The size (volume) of the trade.

`t*` integer

The nanosecond accuracy SIP Unix Timestamp. This is the timestamp of when the SIP received this message from the exchange which produced it.

`x*` integer

The exchange ID. See Exchanges for Polygon.io's mapping of exchange IDs.

`min` object

The most recent minute bar for this ticker.

`av*` integer

The accumulated volume.

`c*` number

The close price for the symbol in the given time period.

`h*` number

The highest price for the symbol in the given time period.

`l*` number

The lowest price for the symbol in the given time period.

`n*` integer

The number of transactions in the aggregate window.

`o*` number

The open price for the symbol in the given time period.

`otc` boolean

Whether or not this aggregate is for an OTC ticker. This field will be left off if false.

`t*` integer

The Unix Msec timestamp for the start of the aggregate window.

`v*` number

The trading volume of the symbol in the given time period.

`vw*` number

The volume weighted average price.

`prevDay` object

The previous day's bar for this ticker.

`c*` number

The close price for the symbol in the given time period.

`h*` number

The highest price for the symbol in the given time period.

`l*` number

The lowest price for the symbol in the given time period.

`o*` number

The open price for the symbol in the given time period.

`otc` boolean

Whether or not this aggregate is for an OTC ticker. This field will be left off if false.

`v*` number

The trading volume of the symbol in the given time period.

`vw*` number

The volume weighted average price.

`ticker` string

The exchange symbol that this item is traded under.

`todaysChange` number

The value of the change from the previous day.

`todaysChangePerc` number

The percentage change since the previous day.

`updated` integer

The last updated timestamp.

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "count": 1,
  "status": "OK",
  "tickers": [
    {
      "day": {
        "c": 20.506,
        "h": 20.64,
        "l": 20.506,
        "o": 20.64,
        "v": 37216,
        "vw": 20.616
      },
      "lastQuote": {
```

```
        "P": 20.6,
        "S": 22,
        "p": 20.5,
        "s": 13,
        "t": 1605192959994246100
      },
      "lastTrade": {
        "c": [
          14,
          41
        ],
        "i": "71675577320245",
        "p": 20.506,
        "s": 2416,
        "t": 1605192894630916600,
        "x": 4
      },
      "min": {
        "av": 37216,
        "c": 20.506,
        "h": 20.506,
        "l": 20.506,
        "n": 1,
        "o": 20.506,
        "t": 1684428600000,
        "v": 5000,
        "vw": 20.5105
      },
      "prevDay": {
        "c": 20.63,
        "h": 21,
        "l": 20.5,
        "o": 20.79,
        "v": 292738,
        "vw": 20.6939
      },
      "ticker": "BCAT",
      "todaysChange": -0.124,
      "todaysChangePerc": -0.601,
      "updated": 1605192894630916600
    }
  ]
}
```


Gainers/Losers

GET /v2/snapshot/locale/us/markets/stocks/{direction}

Get the most up-to-date market data for the current top 20 gainers or losers of the day in the stocks/equities markets.

Top gainers are those tickers whose price has increased by the highest percentage since the previous day's close. Top losers are those tickers whose price has decreased by the highest percentage since the previous day's close.

Note: Snapshot data is cleared at 3:30am EST and gets populated as data is received from the exchanges.

Requires a "Stocks Starter" subscription

Upgrade

Parameters

direction*	gainers

The direction of the snapshot results to return.

include_otc	

Include OTC securities in the response. Default is false (don't include OTC securities).

```
https://api.polygon.io/v2/snapshot/locale/us/markets/stocks/gainers?apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

JSON

Run Query

Response Attributes

`status`* string

The status of this request's response.

`tickers` array

`day` object

The most recent daily bar for this ticker.

`c`* number

The close price for the symbol in the given time period.

h* number

The highest price for the symbol in the given time period.

l* number

The lowest price for the symbol in the given time period.

o* number

The open price for the symbol in the given time period.

otc boolean

Whether or not this aggregate is for an OTC ticker. This field will be left off if false.

v* number

The trading volume of the symbol in the given time period.

vw* number

The volume weighted average price.

fmv number

Fair market value is only available on Business plans. It is our proprietary algorithm to generate a real-time, accurate, fair market value of a tradable security. For more information, contact us.

lastQuote object

The most recent quote for this ticker. This is only returned if your current plan includes quotes.

P* number

The ask price.

S* integer

The ask size in lots.

p* number

The bid price.

s* integer

The bid size in lots.

t* integer

The nanosecond accuracy SIP Unix Timestamp. This is the timestamp of when the SIP received this message from the exchange which produced it.

`lastTrade` object

The most recent trade for this ticker. This is only returned if your current plan includes trades.

`c*` array [integer]

The trade conditions.

`i*` string

The Trade ID which uniquely identifies a trade. These are unique per combination of ticker, exchange, and TRF. For example: A trade for AAPL executed on NYSE and a trade for AAPL executed on NASDAQ could potentially have the same Trade ID.

`p*` number

The price of the trade. This is the actual dollar value per whole share of this trade. A trade of 100 shares with a price of $2.00 would be worth a total dollar value of $200.00.

`s*` integer

The size (volume) of the trade.

`t*` integer

The nanosecond accuracy SIP Unix Timestamp. This is the timestamp of when the SIP received this message from the exchange which produced it.

`x*` integer

The exchange ID. See Exchanges for Polygon.io's mapping of exchange IDs.

`min` object

The most recent minute bar for this ticker.

`av*` integer

The accumulated volume.

`c*` number

The close price for the symbol in the given time period.

`h*` number

The highest price for the symbol in the given time period.

`l*` number

The lowest price for the symbol in the given time period.

`n*` integer

The number of transactions in the aggregate window.

o* number

The open price for the symbol in the given time period.

otc boolean

Whether or not this aggregate is for an OTC ticker. This field will be left off if false.

t* integer

The Unix Msec timestamp for the start of the aggregate window.

v* number

The trading volume of the symbol in the given time period.

vw* number

The volume weighted average price.

prevDay object

The previous day's bar for this ticker.

c* number

The close price for the symbol in the given time period.

h* number

The highest price for the symbol in the given time period.

l* number

The lowest price for the symbol in the given time period.

o* number

The open price for the symbol in the given time period.

otc boolean

Whether or not this aggregate is for an OTC ticker. This field will be left off if false.

v* number

The trading volume of the symbol in the given time period.

vw* number

The volume weighted average price.

ticker string

The exchange symbol that this item is traded under.

`todaysChange` number

The value of the change from the previous day.

`todaysChangePerc` number

The percentage change since the previous day.

`updated` integer

The last updated timestamp.

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "status": "OK",
  "tickers": [
    {
      "day": {
        "c": 14.2284,
        "h": 15.09,
        "l": 14.2,
        "o": 14.33,
        "v": 133963,
        "vw": 14.5311
      },
      "lastQuote": {
        "P": 14.44,
        "S": 11,
        "p": 14.2,
        "s": 25,
        "t": 1605195929997325600
      },
      "lastTrade": {
        "c": [
          63
        ],
        "i": "79372124707124",
        "p": 14.2284,
        "s": 536,
        "t": 1605195848258266000,
        "x": 4
      },
      "min": {
```

```
          "av": 133963,
          "c": 14.2284,
          "h": 14.325,
          "l": 14.2,
          "n": 5,
          "o": 14.28,
          "t": 1684428600000,
          "v": 6108,
          "vw": 14.2426
        },
        "prevDay": {
          "c": 0.73,
          "h": 0.799,
          "l": 0.73,
          "o": 0.75,
          "v": 1568097,
          "vw": 0.7721
        },
        "ticker": "PDS",
        "todaysChange": 13.498,
        "todaysChangePerc": 1849.096,
        "updated": 1605195848258266000
      }
    ]
  }
```


Ticker

GET /v2/snapshot/locale/us/markets/stocks/tickers/{stocksTicker}

Get the most up-to-date market data for a single traded stock ticker.

Note: Snapshot data is cleared at 3:30am EST and gets populated as data is received from the exchanges. This can happen as early as 4am EST.

Requires a "Stocks Starter" subscription Upgrade

Parameters

stocksTicker*	AAPL

Specify a case-sensitive ticker symbol. For example, AAPL represents Apple Inc.

```
https://api.polygon.io/v2/snapshot/locale/us/markets/stocks/tickers/AAPL?apiKey=1
RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

JSON

Run Query

Response Attributes

`status`* string

The status of this request's response.

`request_id`* string

A request id assigned by the server.

`ticker` object

`day` object

The most recent daily bar for this ticker.

`c`* number

The close price for the symbol in the given time period.

`h`* number

The highest price for the symbol in the given time period.

`l`* number

The lowest price for the symbol in the given time period.

`o`* number

The open price for the symbol in the given time period.

`otc` boolean

Whether or not this aggregate is for an OTC ticker. This field will be left off if false.

`v`* number

The trading volume of the symbol in the given time period.

`vw`* number

The volume weighted average price.

`fmv` number

Fair market value is only available on Business plans. It is our proprietary algorithm to generate a real-time, accurate, fair market value of a tradable security. For more information, contact us.

`lastQuote` object

The most recent quote for this ticker. This is only returned if your current plan includes quotes.

`P*` number

The ask price.

`S*` integer

The ask size in lots.

`p*` number

The bid price.

`s*` integer

The bid size in lots.

`t*` integer

The nanosecond accuracy SIP Unix Timestamp. This is the timestamp of when the SIP received this message from the exchange which produced it.

`lastTrade` object

The most recent trade for this ticker. This is only returned if your current plan includes trades.

`c*` array [integer]

The trade conditions.

`i*` string

The Trade ID which uniquely identifies a trade. These are unique per combination of ticker, exchange, and TRF. For example: A trade for AAPL executed on NYSE and a trade for AAPL executed on NASDAQ could potentially have the same Trade ID.

`p*` number

The price of the trade. This is the actual dollar value per whole share of this trade. A trade of 100 shares with a price of $2.00 would be worth a total dollar value of $200.00.

`s*` integer

The size (volume) of the trade.

`t*` integer

The nanosecond accuracy SIP Unix Timestamp. This is the timestamp of when the SIP received this message from the exchange which produced it.

`x`* integer

The exchange ID. See Exchanges for Polygon.io's mapping of exchange IDs.

`min` object

The most recent minute bar for this ticker.

`av`* integer

The accumulated volume.

`c`* number

The close price for the symbol in the given time period.

`h`* number

The highest price for the symbol in the given time period.

`l`* number

The lowest price for the symbol in the given time period.

`n`* integer

The number of transactions in the aggregate window.

`o`* number

The open price for the symbol in the given time period.

`otc` boolean

Whether or not this aggregate is for an OTC ticker. This field will be left off if false.

`t`* integer

The Unix Msec timestamp for the start of the aggregate window.

`v`* number

The trading volume of the symbol in the given time period.

`vw`* number

The volume weighted average price.

`prevDay` object

The previous day's bar for this ticker.

`c`* number

The close price for the symbol in the given time period.

`h`* number

The highest price for the symbol in the given time period.

`l`* number

The lowest price for the symbol in the given time period.

`o`* number

The open price for the symbol in the given time period.

`otc` boolean

Whether or not this aggregate is for an OTC ticker. This field will be left off if false.

`v`* number

The trading volume of the symbol in the given time period.

`vw`* number

The volume weighted average price.

`ticker` string

The exchange symbol that this item is traded under.

`todaysChange` number

The value of the change from the previous day.

`todaysChangePerc` number

The percentage change since the previous day.

`updated` integer

The last updated timestamp.

Was this helpful?

Help us improve

Yes No

Response Object

```
{
  "request_id": "657e430f1ae768891f018e08e03598d8",
```

```
    "status": "OK",
    "ticker": {
      "day": {
        "c": 120.4229,
        "h": 120.53,
        "l": 118.81,
        "o": 119.62,
        "v": 28727868,
        "vw": 119.725
      },
      "lastQuote": {
        "P": 120.47,
        "S": 4,
        "p": 120.46,
        "s": 8,
        "t": 1605195918507251700
      },
      "lastTrade": {
        "c": [
          14,
          41
        ],
        "i": "4046",
        "p": 120.47,
        "s": 236,
        "t": 1605195918306274000,
        "x": 10
      },
      "min": {
        "av": 28724441,
        "c": 120.4201,
        "h": 120.468,
        "l": 120.37,
        "n": 762,
        "o": 120.435,
        "t": 1684428720000,
        "v": 270796,
        "vw": 120.4129
      },
      "prevDay": {
        "c": 119.49,
        "h": 119.63,
        "l": 116.44,
        "o": 117.19,
        "v": 110597265,
        "vw": 118.4998
      },
      "ticker": "AAPL",
      "todaysChange": 0.98,
      "todaysChangePerc": 0.82,
      "updated": 1605195918306274000
```

```
    }
  }
```

Universal Snapshot

GET /v3/snapshot

Get snapshots for assets of all types

Requires a "Stocks Starter" subscription

Upgrade

Parameters

ticker.any_of	NCLH,O:SPY250321C00380000,C:EURUSD,X:BTCUSD,I:SPX

Comma separated list of tickers, up to a maximum of 250. If no tickers are passed then all results will be returned in a paginated manner.

Warning: The maximum number of characters allowed in a URL are subject to your technology stack.

Additional filter parameters

`type`

Query by the type of asset.

`order`

Order results based on the `sort` field.

`limit`

Limit the number of results returned, default is 10 and max is 250.

`sort`

Sort field used for ordering.

```
https://api.polygon.io/v3/snapshot?ticker.any_of=NCLH,O:SPY250321C00380000,C:EURUSD,X:BTCUSD,I:SPX&apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

Run Query

Response Attributes

`next_url` string

If present, this value can be used to fetch the next page of data.

`request_id`* string

`results` array

`break_even_price` number

The price of the underlying asset for the contract to break even. For a call, this value is (strike price + premium paid). For a put, this value is (strike price - premium paid).

`details` object

The details for this contract.

`contract_type`* enum [put, call, other]

The type of contract. Can be "put", "call", or in some rare cases, "other".

`exercise_style`* enum [american, european, bermudan]

The exercise style of this contract. See this link for more details on exercise styles.

`expiration_date`* string

The contract's expiration date in YYYY-MM-DD format.

`shares_per_contract`* number

The number of shares per contract for this contract.

`strike_price`* number

The strike price of the option contract.

`error` string

The error while looking for this ticker.

`fmv` number

Fair market value is only available on Business plans. It's it our proprietary algorithm to generate a real-time, accurate, fair market value of a tradable security. For more information, contact us.

`greeks` object

The greeks for this contract. There are certain circumstances where greeks will not be returned, such as options contracts that are deep in the money. See this article for more information.

`delta`* number

The change in the option's price per $0.01 increment in the price of the underlying asset.

`gamma`* number

The change in delta per $0.01 change in the price of the underlying asset.

`theta`* number

The change in the option's price per day.

`vega`* number

The change in the option's price per 1% increment in volatility.

`implied_volatility` number

The market's forecast for the volatility of the underlying asset, based on this option's current price.

`last_quote` object

The most recent quote for this contract. This is only returned if your current plan includes quotes.

`ask`* number

The ask price.

`ask_exchange` integer

The ask side exchange ID. See Exchanges for Polygon.io's mapping of exchange IDs.

`ask_size` number

The ask size. This represents the number of round lot orders at the given ask price. The normal round lot size is 100 shares. An ask size of 2 means there are 200 shares available to purchase at the given ask price.

`bid`* number

The bid price.

`bid_exchange` integer

The bid side exchange ID. See Exchanges for Polygon.io's mapping of exchange IDs.

`bid_size` number

The bid size. This represents the number of round lot orders at the given bid price. The normal round lot size is 100 shares. A bid size of 2 means there are 200 shares for purchase at the given bid price.

`last_updated`* integer

The nanosecond timestamp of when this information was updated.

`midpoint` number

The average of the bid and ask price.

`timeframe`* enum [DELAYED, REAL-TIME]

The time relevance of the data.

`last_trade` object

The most recent quote for this contract. This is only returned if your current plan includes trades.

`conditions` array [integer]

A list of condition codes.

`exchange` integer

The exchange ID. See Exchanges for Polygon.io's mapping of exchange IDs.

`id` string

The Trade ID which uniquely identifies a trade. These are unique per combination of ticker, exchange, and TRF. For example: A trade for AAPL executed on NYSE and a trade for AAPL executed on NASDAQ could potentially have the same Trade ID.

`last_updated`* integer

The nanosecond timestamp of when this information was updated.

`participant_timestamp`* integer

The nanosecond Exchange Unix Timestamp. This is the timestamp of when the trade was generated at the exchange.

`price`* number

The price of the trade. This is the actual dollar value per whole share of this trade. A trade of 100 shares with a price of $2.00 would be worth a total dollar value of $200.00.

`sip_timestamp` integer

The nanosecond accuracy SIP Unix Timestamp. This is the timestamp of when the SIP received this trade from the exchange which produced it.

`size`* integer

The size of a trade (also known as volume).

`timeframe`* enum [DELAYED, REAL-TIME]

The time relevance of the data.

`market_status` string

The market status for the market that trades this ticker. Possible values for stocks, options, crypto, and forex snapshots are open, closed, early_trading, or late_trading. Possible values for indices snapshots are regular_trading, closed, early_trading, and late_trading.

`message` string

The error message while looking for this ticker.

`name` string

The name of this contract.

`open_interest` number

The quantity of this contract held at the end of the last trading day.

`session` object

`change*` number

The value of the price change for the asset from the previous trading day.

`change_percent*` number

The percent of the price change for the asset from the previous trading day.

`close*` number

The closing price of the asset for the day.

`early_trading_change` number

Today's early trading change amount, difference between price and previous close if in early trading hours, otherwise difference between last price during early trading and previous close.

`early_trading_change_percent` number

Today's early trading change as a percentage.

`high*` number

The highest price of the asset for the day.

`late_trading_change` number

Today's late trading change amount, difference between price and today's close if in late trading hours, otherwise difference between last price during late trading and today's close.

`late_trading_change_percent` number

Today's late trading change as a percentage.

low* number

The lowest price of the asset for the day.

open* number

The open price of the asset for the day.

previous_close* number

The closing price of the asset for the previous trading day.

price number

The price of the most recent trade or bid price for this asset.

volume number

The trading volume for the asset for the day.

ticker* string

The ticker symbol for the asset.

type enum [stocks, options, fx, crypto, indices]

The asset class for this ticker.

underlying_asset object

Information on the underlying stock for this options contract. The market data returned depends on your current stocks plan.

change_to_break_even* number

The change in price for the contract to break even.

last_updated* integer

The nanosecond timestamp of when this information was updated.

price number

The price of the trade. This is the actual dollar value per whole share of this trade. A trade of 100 shares with a price of $2.00 would be worth a total dollar value of $200.00.

ticker* string

The ticker symbol for the contract's underlying asset.

timeframe* enum [DELAYED, REAL-TIME]

The time relevance of the data.

`value` number

The value of the underlying index.

`value` number

Value of Index.

`status*` string

The status of this request's response.

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "request_id": "abc123",
  "results": [
    {
      "break_even_price": 171.075,
      "details": {
        "contract_type": "call",
        "exercise_style": "american",
        "expiration_date": "2022-10-14",
        "shares_per_contract": 100,
        "strike_price": 5,
        "underlying_ticker": "NCLH"
      },
      "fmv": 0.05,
      "greeks": {
        "delta": 0.5520187372272933,
        "gamma": 0.00706756515659829,
        "theta": -0.018532772783847958,
        "vega": 0.7274811132998142
      },
      "implied_volatility": 0.3048997097864957,
      "last_quote": {
        "ask": 21.25,
        "ask_exchange": 12,
        "ask_size": 110,
        "bid": 20.9,
        "bid_exchange": 10,
        "bid_size": 172,
        "last_updated": 1636573458756383500,
        "midpoint": 21.075,
        "timeframe": "REAL-TIME"
      },
```

```
      "last_trade": {
        "conditions": [
          209
        ],
        "exchange": 316,
        "price": 0.05,
        "sip_timestamp": 1675280958783136800,
        "size": 2,
        "timeframe": "REAL-TIME"
      },
      "market_status": "closed",
      "name": "NCLH $5 Call",
      "open_interest": 8921,
      "session": {
        "change": -0.05,
        "change_percent": -1.07,
        "close": 6.65,
        "early_trading_change": -0.01,
        "early_trading_change_percent": -0.03,
        "high": 7.01,
        "late_trading_change": -0.4,
        "late_trading_change_percent": -0.02,
        "low": 5.42,
        "open": 6.7,
        "previous_close": 6.71,
        "volume": 67
      },
      "ticker": "O:NCLH221014C00005000",
      "type": "options",
      "underlying_asset": {
        "change_to_break_even": 23.123999999999995,
        "last_updated": 1636573459862384600,
        "price": 147.951,
        "ticker": "AAPL",
        "timeframe": "REAL-TIME"
      }
    },
    {
      "fmv": 0.05,
      "last_quote": {
        "ask": 21.25,
        "ask_exchange": 300,
        "ask_size": 110,
        "bid": 20.9,
        "bid_exchange": 323,
        "bid_size": 172,
        "last_updated": 1636573458756383500,
        "timeframe": "REAL-TIME"
      },
      "last_trade": {
        "conditions": [
          209
```

```
        ],
        "exchange": 316,
        "id": "4064",
        "last_updated": 1675280958783136800,
        "price": 0.05,
        "size": 2,
        "timeframe": "REAL-TIME"
      },
      "market_status": "closed",
      "name": "Apple Inc.",
      "session": {
        "change": -1.05,
        "change_percent": -4.67,
        "close": 21.4,
        "early_trading_change": -0.39,
        "early_trading_change_percent": -0.07,
        "high": 22.49,
        "late_trading_change": 1.2,
        "late_trading_change_percent": 3.92,
        "low": 21.35,
        "open": 22.49,
        "previous_close": 22.45,
        "volume": 37
      },
      "ticker": "AAPL",
      "type": "stocks"
    },
    {
      "error": "NOT_FOUND",
      "message": "Ticker not found.",
      "ticker": "TSLAAPL"
    }
  ],
  "status": "OK"
}
```


Simple Moving Average (SMA)

GET /v1/indicators/sma/{stockTicker}

Get the simple moving average (SMA) for a ticker symbol over a given time range.

Parameters

stockTicker*	AAPL

The ticker symbol for which to get simple moving average (SMA) data.

timestamp	

Query by timestamp. Either a date with the format YYYY-MM-DD or a millisecond timestamp.

Additional filter parameters

timespan	day

The size of the aggregate time window.

adjusted	true

Whether or not the aggregates used to calculate the simple moving average are adjusted for splits. By default, aggregates are adjusted. Set this to false to get results that are NOT adjusted for splits.

window	50

The window size used to calculate the simple moving average (SMA). i.e. a window size of 10 with daily aggregates would result in a 10 day moving average.

series_type	close

The price in the aggregate which will be used to calculate the simple moving average. i.e. 'close' will result in using close prices to calculate the simple moving average (SMA).

expand_underlying	

Whether or not to include the aggregates used to calculate this indicator in the response.

order	desc

The order in which to return the results, ordered by timestamp.

limit	

Limit the number of results returned, default is 10 and max is 5000

```
https://api.polygon.io/v1/indicators/sma/AAPL?timespan=day&adjusted=true&window=50&series_type=close&order=desc&apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

JSON

Run Query

Response Attributes

`next_url` string

If present, this value can be used to fetch the next page of data.

`request_id` string

A request id assigned by the server.

`results` object

`underlying` object

`aggregates` array

`c*` number

The close price for the symbol in the given time period.

`h*` number

The highest price for the symbol in the given time period.

`l*` number

The lowest price for the symbol in the given time period.

`n*` integer

The number of transactions in the aggregate window.

`o*` number

The open price for the symbol in the given time period.

`otc` boolean

Whether or not this aggregate is for an OTC ticker. This field will be left off if false.

`t*` number

The Unix Msec timestamp for the start of the aggregate window.

`v*` number

The trading volume of the symbol in the given time period.

`vw*` number

The volume weighted average price.

`url` string

The URL which can be used to request the underlying aggregates used in this request.

`values` array

`timestamp` integer

The Unix Msec timestamp from the last aggregate used in this calculation.

`value` number

The indicator value for this period.

`status` string

The status of this request's response.

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "next_url": "https://api.polygon.io/v1/indicators/sma/AAPL?cursor=YWN0aXZlPXRydWUmZGF0Z
  "request_id": "a47d1beb8c11b6ae897ab76cdbbf35a3",
  "results": {
    "underlying": {
      "aggregates": [
        {
          "c": 75.0875,
          "h": 75.15,
          "l": 73.7975,
          "n": 1,
          "o": 74.06,
          "t": 1577941200000,
          "v": 135647456,
          "vw": 74.6099
        },
        {
          "c": 74.3575,
          "h": 75.145,
          "l": 74.125,
          "n": 1,
          "o": 74.2875,
          "t": 1578027600000,
          "v": 146535512,
          "vw": 74.7026
        }
      ],
      "url": "https://api.polygon.io/v2/aggs/ticker/AAPL/range/1/day/2003-01-01/2022-07-2
    },
    "values": [
      {
        "timestamp": 1517562000016,
        "value": 140.139
```

```
        }
      ]
    },
    "status": "OK"
  }
```


Exponential Moving Average (EMA)

GET /v1/indicators/ema/{stockTicker}

Get the exponential moving average (EMA) for a ticker symbol over a given time range.

Parameters

stockTicker*	AAPL

The ticker symbol for which to get exponential moving average (EMA) data.

timestamp	

Query by timestamp. Either a date with the format YYYY-MM-DD or a millisecond timestamp.

Additional filter parameters

timespan	day

The size of the aggregate time window.

adjusted	true

Whether or not the aggregates used to calculate the exponential moving average are adjusted for splits. By default, aggregates are adjusted. Set this to false to get results that are NOT adjusted for splits.

window	50

The window size used to calculate the exponential moving average (EMA). i.e. a window size of 10 with daily aggregates would result in a 10 day moving average.

series_type	close

The price in the aggregate which will be used to calculate the exponential moving average. i.e. 'close' will result in using close prices to calculate the exponential moving average (EMA).

expand_underlying	

Whether or not to include the aggregates used to calculate this indicator in the response.

order	desc

The order in which to return the results, ordered by timestamp.

limit	

Limit the number of results returned, default is 10 and max is 5000

```
https://api.polygon.io/v1/indicators/ema/AAPL?timespan=day&adjusted=true&window=50&series_type=close&order=desc&apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

JSON Run Query

Response Attributes

`next_url` string

If present, this value can be used to fetch the next page of data.

`request_id` string

A request id assigned by the server.

`results` object

`underlying` object

`aggregates` array

`c*` number

The close price for the symbol in the given time period.

`h*` number

The highest price for the symbol in the given time period.

`l*` number

The lowest price for the symbol in the given time period.

`n*` integer

The number of transactions in the aggregate window.

`o`* number

The open price for the symbol in the given time period.

`otc` boolean

Whether or not this aggregate is for an OTC ticker. This field will be left off if false.

`t`* number

The Unix Msec timestamp for the start of the aggregate window.

`v`* number

The trading volume of the symbol in the given time period.

`vw`* number

The volume weighted average price.

`url` string

The URL which can be used to request the underlying aggregates used in this request.

`values` array

`timestamp` integer

The Unix Msec timestamp from the last aggregate used in this calculation.

`value` number

The indicator value for this period.

`status` string

The status of this request's response.

Was this helpful?

Help us improve

Yes No

Response Object

```
{
  "next_url": "https://api.polygon.io/v1/indicators/ema/AAPL?cursor=YWN0aXZlPXRydWUmZGF0Z
  "request_id": "a47d1beb8c11b6ae897ab76cdbbf35a3",
  "results": {
    "underlying": {
      "url": "https://api.polygon.io/v2/aggs/ticker/AAPL/range/1/day/2003-01-01/2022-07-2
    },
    "values": [
      {
```

```
        "timestamp": 1517562000016,
        "value": 140.139
      }
    ]
  },
  "status": "OK"
}
```


Moving Average Convergence/Divergence (MACD)

GET /v1/indicators/macd/{stockTicker}

Get moving average convergence/divergence (MACD) data for a ticker symbol over a given time range.

Parameters

stockTicker*	AAPL

The ticker symbol for which to get MACD data.

timestamp	

Query by timestamp. Either a date with the format YYYY-MM-DD or a millisecond timestamp.

Additional filter parameters

timespan	day

The size of the aggregate time window.

adjusted	true

Whether or not the aggregates used to calculate the MACD are adjusted for splits. By default, aggregates are adjusted. Set this to false to get results that are NOT adjusted for splits.

short_window	12

The short window size used to calculate MACD data.

long_window	26

The long window size used to calculate MACD data.

signal_window	9

The window size used to calculate the MACD signal line.

series_type	close

The price in the aggregate which will be used to calculate the MACD. i.e. 'close' will result in using close prices to calculate the MACD.

expand_underlying	

Whether or not to include the aggregates used to calculate this indicator in the response.

order	desc

The order in which to return the results, ordered by timestamp.

limit	

Limit the number of results returned, default is 10 and max is 5000

```
https://api.polygon.io/v1/indicators/macd/AAPL?timespan=day&adjusted=true&short_window=12&long_window=26&signal_window=9&series_type=close&order=desc&apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

JSON

Run Query

Response Attributes

next_url string

If present, this value can be used to fetch the next page of data.

request_id string

A request id assigned by the server.

results object

underlying object

aggregates array

c* number

The close price for the symbol in the given time period.

h* number

The highest price for the symbol in the given time period.

`l`* number

The lowest price for the symbol in the given time period.

`n`* integer

The number of transactions in the aggregate window.

`o`* number

The open price for the symbol in the given time period.

`otc` boolean

Whether or not this aggregate is for an OTC ticker. This field will be left off if false.

`t`* number

The Unix Msec timestamp for the start of the aggregate window.

`v`* number

The trading volume of the symbol in the given time period.

`vw`* number

The volume weighted average price.

`url` string

The URL which can be used to request the underlying aggregates used in this request.

`values` array

`histogram` number

The indicator value for this period.

`signal` number

The indicator value for this period.

`timestamp` integer

The Unix Msec timestamp from the last aggregate used in this calculation.

`value` number

The indicator value for this period.

`status` string

The status of this request's response.

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "next_url": "https://api.polygon.io/v1/indicators/macd/AAPL?cursor=YWN0aXZlPXRydWUmZGF0
  "request_id": "a47d1beb8c11b6ae897ab76cdbbf35a3",
  "results": {
    "underlying": {
      "url": "https://api.polygon.io/v2/aggs/ticker/AAPL/range/1/day/2003-01-01/2022-07-2
    },
    "values": [
      {
        "histogram": 38.3801666667,
        "signal": 106.9811666667,
        "timestamp": 1517562000016,
        "value": 145.3613333333
      },
      {
        "histogram": 41.098859136,
        "signal": 102.7386283473,
        "timestamp": 1517562001016,
        "value": 143.8374874833
      }
    ]
  },
  "status": "OK"
}
```


Relative Strength Index (RSI)

GET /v1/indicators/rsi/{stockTicker}

Get the relative strength index (RSI) for a ticker symbol over a given time range.

Parameters

stockTicker*	AAPL

The ticker symbol for which to get relative strength index (RSI) data.

`timestamp`	

Query by timestamp. Either a date with the format YYYY-MM-DD or a millisecond timestamp.

Additional filter parameters

`timespan`	day

The size of the aggregate time window.

`adjusted`	true

Whether or not the aggregates used to calculate the relative strength index are adjusted for splits. By default, aggregates are adjusted. Set this to false to get results that are NOT adjusted for splits.

`window`	14

The window size used to calculate the relative strength index (RSI).

`series_type`	close

The price in the aggregate which will be used to calculate the relative strength index. i.e. 'close' will result in using close prices to calculate the relative strength index (RSI).

`expand_underlying`	

Whether or not to include the aggregates used to calculate this indicator in the response.

`order`	desc

The order in which to return the results, ordered by timestamp.

`limit`	

Limit the number of results returned, default is 10 and max is 5000

```
https://api.polygon.io/v1/indicators/rsi/AAPL?timespan=day&adjusted=true&window=14&series_type=close&order=desc&apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

JSON

Run Query

Response Attributes

`next_url` string

If present, this value can be used to fetch the next page of data.

`request_id` string

A request id assigned by the server.

`results` object

`underlying` object

`aggregates` array

`c*` number

The close price for the symbol in the given time period.

`h*` number

The highest price for the symbol in the given time period.

`l*` number

The lowest price for the symbol in the given time period.

`n*` integer

The number of transactions in the aggregate window.

`o*` number

The open price for the symbol in the given time period.

`otc` boolean

Whether or not this aggregate is for an OTC ticker. This field will be left off if false.

`t*` number

The Unix Msec timestamp for the start of the aggregate window.

`v*` number

The trading volume of the symbol in the given time period.

`vw*` number

The volume weighted average price.

`url` string

The URL which can be used to request the underlying aggregates used in this request.

`values` array

`timestamp` integer

The Unix Msec timestamp from the last aggregate used in this calculation.

`value` number

The indicator value for this period.

`status` string

The status of this request's response.

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "next_url": "https://api.polygon.io/v1/indicators/rsi/AAPL?cursor=YWN0aXZlPXRydWUmZGF0Z
  "request_id": "a47d1beb8c11b6ae897ab76cdbbf35a3",
  "results": {
    "underlying": {
      "url": "https://api.polygon.io/v2/aggs/ticker/AAPL/range/1/day/2003-01-01/2022-07-2
    },
    "values": [
      {
        "timestamp": 1517562000016,
        "value": 82.19
      }
    ]
  },
  "status": "OK"
}
```


Tickers

GET /v3/reference/tickers

Query all ticker symbols which are supported by Polygon.io. This API currently includes Stocks/Equities, Indices, Forex, and Crypto.

Parameters

`ticker`

Specify a ticker symbol. Defaults to empty string which queries all tickers.

Additional filter parameters

`type`

Specify the type of the tickers. Find the types that we support via our Ticker Types API. Defaults to empty string which queries all types.

`market`

Filter by market type. By default all markets are included.

`exchange`

Specify the primary exchange of the asset in the ISO code format. Find more information about the ISO codes at the ISO org website. Defaults to empty string which queries all exchanges.

`cusip`

Specify the CUSIP code of the asset you want to search for. Find more information about CUSIP codes at their website. Defaults to empty string which queries all CUSIPs.

Note: Although you can query by CUSIP, due to legal reasons we do not return the CUSIP in the response.

`cik`

Specify the CIK of the asset you want to search for. Find more information about CIK codes at their website. Defaults to empty string which queries all CIKs.

`date`

Specify a point in time to retrieve tickers available on that date. Defaults to the most recent available date.

`search`

Search for terms within the ticker and/or company name.

`active` true

Specify if the tickers returned should be actively traded on the queried date. Default is true.

`order`

Order results based on the `sort` field.

`limit`

Limit the number of results returned, default is 100 and max is 1000.

`sort`

Sort field used for ordering.

```
https://api.polygon.io/v3/reference/tickers?active=true&apiKey=1RQBxJjR6uKqzsfPUp
WtrHCSsFMOwSjj
```

Copy

JSON

Run Query

Response Attributes

`count` integer

The total number of results for this request.

`next_url` string

If present, this value can be used to fetch the next page of data.

`request_id` string

A request id assigned by the server.

`results` array

An array of tickers that match your query.
Note: Although you can query by CUSIP, due to legal reasons we do not return the CUSIP in the response.

`active` boolean

Whether or not the asset is actively traded. False means the asset has been delisted.

`cik` string

The CIK number for this ticker. Find more information here.

`composite_figi` string

The composite OpenFIGI number for this ticker. Find more information here

---

`currency_name` string

The name of the currency that this asset is traded with.

`delisted_utc` string

The last date that the asset was traded.

`last_updated_utc` string

The information is accurate up to this time.

`locale`* enum [us, global]

The locale of the asset.

`market*` enum [stocks, crypto, fx, otc, indices]

The market type of the asset.

`name*` string

The name of the asset. For stocks/equities this will be the companies registered name. For crypto/fx this will be the name of the currency or coin pair.

`primary_exchange` string

The ISO code of the primary listing exchange for this asset.

`share_class_figi` string

The share Class OpenFIGI number for this ticker. Find more information here

---

`ticker*` string

The exchange symbol that this item is traded under.

`type` string

The type of the asset. Find the types that we support via our Ticker Types API.

`status` string

The status of this request's response.

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "count": 1,
  "next_url": "https://api.polygon.io/v3/reference/tickers?cursor=YWN0aXZlPXRydWUmZGF0ZT0
  "request_id": "e70013d92930de90e089dc8fa098888e",
  "results": [
    {
      "active": true,
      "cik": "0001090872",
      "composite_figi": "BBG000BWQYZ5",
      "currency_name": "usd",
      "last_updated_utc": "2021-04-25T00:00:00Z",
      "locale": "us",
      "market": "stocks",
      "name": "Agilent Technologies Inc.",
      "primary_exchange": "XNYS",
```

```
      "share_class_figi": "BBG001SCTQY4",
      "ticker": "A",
      "type": "CS"
    }
  ],
  "status": "OK"
}
```


Ticker Details v3

GET /v3/reference/tickers/{ticker}

Get a single ticker supported by Polygon.io. This response will have detailed information about the ticker and the company behind it.

Parameters

ticker*	AAPL

The ticker symbol of the asset.

date	

Specify a point in time to get information about the ticker available on that date. When retrieving information from SEC filings, we compare this date with the period of report date on the SEC filing.

For example, consider an SEC filing submitted by AAPL on 2019-07-31, with a period of report date ending on 2019-06-29. That means that the filing was submitted on 2019-07-31, but the filing was created based on information from 2019-06-29. If you were to query for AAPL details on 2019-06-29, the ticker details would include information from the SEC filing.

Defaults to the most recent available date.

```
https://api.polygon.io/v3/reference/tickers/AAPL?apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

JSON

Run Query

Response Attributes

`count` integer

The total number of results for this request.

`request_id` string

A request id assigned by the server.

`results` object

Ticker with details.

`active*` boolean

Whether or not the asset is actively traded. False means the asset has been delisted.

`address` object

`address1` string

The first line of the company's headquarters address.

`city` string

The city of the company's headquarters address.

`postal_code` string

The postal code of the company's headquarters address.

`state` string

The state of the company's headquarters address.

`branding` object

`icon_url` string

A link to this ticker's company's icon. Icon's are generally smaller, square images that represent the company at a glance. Note that you must provide an API key when accessing this URL. See the "Authentication" section at the top of this page for more details.

`logo_url` string

A link to this ticker's company's logo. Note that you must provide an API key when accessing this URL. See the "Authentication" section at the top of this page for more details.

`cik` string

The CIK number for this ticker. Find more information here.

`composite_figi` string

The composite OpenFIGI number for this ticker. Find more information here

---

`currency_name*` string

The name of the currency that this asset is traded with.

`delisted_utc` string

The last date that the asset was traded.

`description` string

A description of the company and what they do/offer.

`homepage_url` string

The URL of the company's website homepage.

`list_date` string

The date that the symbol was first publicly listed in the format YYYY-MM-DD.

`locale*` enum [us, global]

The locale of the asset.

`market*` enum [stocks, crypto, fx, otc, indices]

The market type of the asset.

`market_cap` number

The most recent close price of the ticker multiplied by weighted outstanding shares.

`name*` string

The name of the asset. For stocks/equities this will be the companies registered name. For crypto/fx this will be the name of the currency or coin pair.

`phone_number` string

The phone number for the company behind this ticker.

`primary_exchange` string

The ISO code of the primary listing exchange for this asset.

`round_lot` number

Round lot size of this security.

`share_class_figi` string

The share Class OpenFIGI number for this ticker. Find more information here

---

`share_class_shares_outstanding` number

The recorded number of outstanding shares for this particular share class.

`sic_code` string

The standard industrial classification code for this ticker. For a list of SIC Codes, see the SEC's SIC Code List.

`sic_description` string

A description of this ticker's SIC code.

`ticker*` string

The exchange symbol that this item is traded under.

`ticker_root` string

The root of a specified ticker. For example, the root of BRK.A is BRK.

`ticker_suffix` string

The suffix of a specified ticker. For example, the suffix of BRK.A is A.

`total_employees` number

The approximate number of employees for the company.

`type` string

The type of the asset. Find the types that we support via our Ticker Types API.

`weighted_shares_outstanding` number

The shares outstanding calculated assuming all shares of other share classes are converted to this share class.

`status` string

The status of this request's response.

Was this helpful?

Help us improve

Yes No

Response Object

```
{
  "request_id": "31d59dda-80e5-4721-8496-d0d32a654afe",
  "results": {
    "active": true,
```

```
    "address": {
      "address1": "One Apple Park Way",
      "city": "Cupertino",
      "postal_code": "95014",
      "state": "CA"
    },
    "branding": {
      "icon_url": "https://api.polygon.io/v1/reference/company-branding/d3d3LmFwcGxlLmNvb
      "logo_url": "https://api.polygon.io/v1/reference/company-branding/d3d3LmFwcGxlLmNvb
    },
    "cik": "0000320193",
    "composite_figi": "BBG000B9XRY4",
    "currency_name": "usd",
    "description": "Apple designs a wide variety of consumer electronic devices, includin
    "homepage_url": "https://www.apple.com",
    "list_date": "1980-12-12",
    "locale": "us",
    "market": "stocks",
    "market_cap": 2771126040150,
    "name": "Apple Inc.",
    "phone_number": "(408) 996-1010",
    "primary_exchange": "XNAS",
    "round_lot": 100,
    "share_class_figi": "BBG001S5N8V8",
    "share_class_shares_outstanding": 16406400000,
    "sic_code": "3571",
    "sic_description": "ELECTRONIC COMPUTERS",
    "ticker": "AAPL",
    "ticker_root": "AAPL",
    "total_employees": 154000,
    "type": "CS",
    "weighted_shares_outstanding": 16334371000
  },
  "status": "OK"
}
```


Ticker Events

GET /vX/reference/tickers/{id}/events

Get a timeline of events for the entity associated with the given ticker, CUSIP, or Composite FIGI.

This API is experimental.

Parameters

id*	META

Identifier of an asset. This can currently be a Ticker, CUSIP, or Composite FIGI. When given a ticker, we return events for the entity currently represented by that ticker. To find events for entities previously associated with a ticker, find the relevant identifier using the Ticker Details Endpoint

types	

A comma-separated list of the types of event to include. Currently ticker_change is the only supported event_type. Leave blank to return all supported event_types.

```
https://api.polygon.io/vX/reference/tickers/META/events?apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

Run Query

Response Attributes

`request_id` string

A request id assigned by the server.

`results` object

`events` array [undefined]

`name` string

`status` string

The status of this request's response.

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "request_id": "31d59dda-80e5-4721-8496-d0d32a654afe",
  "results": {
    "events": [
```

```
        {
          "date": "2022-06-09",
          "ticker_change": {
            "ticker": "META"
          },
          "type": "ticker_change"
        },
        {
          "date": "2012-05-18",
          "ticker_change": {
            "ticker": "FB"
          },
          "type": "ticker_change"
        }
      ],
      "name": "Meta Platforms, Inc. Class A Common Stock"
    },
    "status": "OK"
  }
```


Ticker News

GET /v2/reference/news

Get the most recent news articles relating to a stock ticker symbol, including a summary of the article and a link to the original source.

Parameters

`ticker`

Return results that contain this ticker.

Additional filter parameters

`published_utc`

Return results published on, before, or after this date.

Additional filter parameters

`order`

Order results based on the `sort` field.

`limit`

Limit the number of results returned, default is 10 and max is 1000.

`sort`

Sort field used for ordering.

`https://api.polygon.io/v2/reference/news?apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj` Copy

JSON Run Query

Response Attributes

`count` integer

The total number of results for this request.

`next_url` string

If present, this value can be used to fetch the next page of data.

`request_id` string

A request id assigned by the server.

`results` array

`amp_url` string

The mobile friendly Accelerated Mobile Page (AMP) URL.

`article_url*` string

A link to the news article.

`author*` string

The article's author.

`description` string

A description of the article.

`id*` string

Unique identifier for the article.

`image_url` string

The article's image URL.

`keywords` array [string]

The keywords associated with the article (which will vary depending on the publishing source).

`published_utc`* string

The date the article was published on.

`publisher`* object

`favicon_url` string

The publisher's homepage favicon URL.

`homepage_url`* string

The publisher's homepage URL.

`logo_url`* string

The publisher's logo URL.

`name`* string

The publisher's name.

`tickers`* array [string]

The ticker symbols associated with the article.

`title`* string

The title of the news article.

`status` string

The status of this request's response.

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "count": 1,
  "next_url": "https://api.polygon.io:443/v2/reference/news?cursor=eyJsaW1pdCI6MSwic29ydC
```

```
    "request_id": "831afdb0b8078549fed053476984947a",
    "results": [
      {
        "amp_url": "https://amp.benzinga.com/amp/content/20784086",
        "article_url": "https://www.benzinga.com/markets/cryptocurrency/21/04/20784086/cath
        "author": "Rachit  Vats",
        "description": "<p>Cathie Wood-led Ark Investment Management on Friday snapped up a
        "id": "nJsSJJdwViHZcw5367rZi7_qkXLfMzacXBfpv-vD9UA",
        "image_url": "https://cdn2.benzinga.com/files/imagecache/og_image_social_share_120(
        "keywords": [
          "Sector ETFs",
          "Penny Stocks",
          "Cryptocurrency",
          "Small Cap",
          "Markets",
          "Trading Ideas",
          "ETFs"
        ],
        "published_utc": "2021-04-26T02:33:17Z",
        "publisher": {
          "favicon_url": "https://s3.polygon.io/public/public/assets/news/favicons/benzinga
          "homepage_url": "https://www.benzinga.com/",
          "logo_url": "https://s3.polygon.io/public/public/assets/news/logos/benzinga.svg",
          "name": "Benzinga"
        },
        "tickers": [
          "DOCU",
          "DDD",
          "NIU",
          "ARKF",
          "NVDA",
          "SKLZ",
          "PCAR",
          "MASS",
          "PSTI",
          "SPFR",
          "TREE",
          "PHR",
          "IRDM",
          "BEAM",
          "ARKW",
          "ARKK",
          "ARKG",
          "PSTG",
          "SQ",
          "IONS",
          "SYRS"
        ],
        "title": "Cathie Wood Adds More Coinbase, Skillz, Trims Square"
      }
    ],
```

```
  "status": "OK"
}
```


Ticker Types

GET /v3/reference/tickers/types

List all ticker types that Polygon.io has.

Parameters

asset_class	stocks

Filter by asset class.

locale	

Filter by locale.

```
https://api.polygon.io/v3/reference/tickers/types?asset_class=stocks&apiKey=1RQBx
JjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

JSON Run Query

Response Attributes

`count` integer

The total number of results for this request.

`request_id`* string

A request ID assigned by the server.

`results` array

`asset_class`* enum [stocks, options, crypto, fx, indices]

An identifier for a group of similar financial instruments.

`code`* string

A code used by Polygon.io to refer to this ticker type.

`description`* string

A short description of this ticker type.

`locale`* enum [us, global]

An identifier for a geographical location.

`status`* string

The status of this request's response.

Was this helpful?
Help us improve

Yes No

Response Object


```
{
  "count": 1,
  "request_id": "31d59dda-80e5-4721-8496-d0d32a654afe",
  "results": [
    {
      "asset_class": "stocks",
      "code": "CS",
      "description": "Common Stock",
      "locale": "us"
    }
  ],
  "status": "OK"
}
```


Market Holidays

GET /v1/marketstatus/upcoming

Get upcoming market holidays and their open/close times.

```
https://api.polygon.io/v1/marketstatus/upcoming?apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

Run Query

Response Attributes

`response` array

`close` string

The market close time on the holiday (if it's not closed).

`date` string

The date of the holiday.

`exchange` string

Which market the record is for.

`name` string

The name of the holiday.

`open` string

The market open time on the holiday (if it's not closed).

`status` string

The status of the market on the holiday.

Was this helpful?

Help us improve

Yes No

Response Object


```
[
  {
    "date": "2020-11-26",
    "exchange": "NYSE",
    "name": "Thanksgiving",
    "status": "closed"
  },
  {
```

```
    "date": "2020-11-26",
    "exchange": "NASDAQ",
    "name": "Thanksgiving",
    "status": "closed"
  },
  {
    "date": "2020-11-26",
    "exchange": "OTC",
    "name": "Thanksgiving",
    "status": "closed"
  },
  {
    "close": "2020-11-27T18:00:00.000Z",
    "date": "2020-11-27",
    "exchange": "NASDAQ",
    "name": "Thanksgiving",
    "open": "2020-11-27T14:30:00.000Z",
    "status": "early-close"
  },
  {
    "close": "2020-11-27T18:00:00.000Z",
    "date": "2020-11-27",
    "exchange": "NYSE",
    "name": "Thanksgiving",
    "open": "2020-11-27T14:30:00.000Z",
    "status": "early-close"
  }
]
```


Market Status

GET /v1/marketstatus/now

Get the current trading status of the exchanges and overall financial markets.

```
https://api.polygon.io/v1/marketstatus/now?apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSj
j
```

Copy

Run Query

Response Attributes

`afterHours` boolean

Whether or not the market is in post-market hours.

`currencies` object

`crypto` string

The status of the crypto market.

`fx` string

The status of the forex market.

`earlyHours` boolean

Whether or not the market is in pre-market hours.

`exchanges` object

`nasdaq` string

The status of the Nasdaq market.

`nyse` string

The status of the NYSE market.

`otc` string

The status of the OTC market.

`indicesGroups` object

`cccy` string

The status of Cboe Streaming Market Indices Cryptocurrency ("CCCY") indices trading hours.

`dow_jones` string

The status of Dow Jones indices trading hours

`ftse_russell` string

The status of Financial Times Stock Exchange Group ("FTSE") Russell indices trading hours.

`msci` string

The status of Morgan Stanley Capital International ("MSCI") indices trading hours.

`mstar` string

The status of Morningstar ("MSTAR") indices trading hours.

`mstarc`

The status of Morningstar Customer ("MSTARC") indices trading hours.

`nasdaq` string

The status of National Association of Securities Dealers Automated Quotations ("Nasdaq") indices trading hours.

`s_and_p` string

The status of Standard & Poors's ("S&P") indices trading hours.

`societe_generale` string

The status of Societe Generale indices trading hours.

`market` string

The status of the market as a whole.

`serverTime` string

The current time of the server, returned as a date-time in RFC3339 format.

Was this helpful?

Help us improve

Yes No

Response Object


```json
{
  "afterHours": true,
  "currencies": {
    "crypto": "open",
    "fx": "open"
  },
  "earlyHours": false,
  "exchanges": {
    "nasdaq": "extended-hours",
    "nyse": "extended-hours",
    "otc": "closed"
  },
  "market": "extended-hours",
  "serverTime": "2020-11-10T17:37:37-05:00"
}
```

Stock Splits v3

GET /v3/reference/splits

Get a list of historical stock splits, including the ticker symbol, the execution date, and the factors of the split ratio.

Parameters

`ticker`

Return the stock splits that contain this ticker.

Additional filter parameters

`execution_date`

Query by execution date with the format YYYY-MM-DD.

Additional filter parameters

`reverse_split`

Query for reverse stock splits. A split ratio where split_from is greater than split_to represents a reverse split. By default this filter is not used.

`order`

Order results based on the `sort` field.

`limit`

Limit the number of results returned, default is 10 and max is 1000.

`sort`

Sort field used for ordering.

```
https://api.polygon.io/v3/reference/splits?apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSj
j
```

Copy

JSON

Run Query

Response Attributes

`next_url` string

If present, this value can be used to fetch the next page of data.

`request_id` string

`results` array

`execution_date` string

The execution date of the stock split. On this date the stock split was applied.

`split_from`* number

The second number in the split ratio.
For example: In a 2-for-1 split, split_from would be 1.

`split_to`* number

The first number in the split ratio.
For example: In a 2-for-1 split, split_to would be 2.

`ticker` string

The ticker symbol of the stock split.

`status` string

The status of this request's response.

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "next_url": "https://api.polygon.io/v3/splits/AAPL?cursor=YWN0aXZlPXRydWUmZGF0ZT0yMDIxL
  "results": [
    {
      "execution_date": "2020-08-31",
      "split_from": 1,
      "split_to": 4,
      "ticker": "AAPL"
    },
    {
      "execution_date": "2005-02-28",
      "split_from": 1,
      "split_to": 2,
```

```
      "ticker": "AAPL"
    }
  ],
  "status": "OK"
}
```


Dividends v3

GET /v3/reference/dividends

Get a list of historical cash dividends, including the ticker symbol, declaration date, ex-dividend date, record date, pay date, frequency, and amount.

Parameters

`ticker`

Return the dividends that contain this ticker.

Additional filter parameters

`ex_dividend_date`

Query by ex-dividend date with the format YYYY-MM-DD.

Additional filter parameters

`record_date`

Query by record date with the format YYYY-MM-DD.

Additional filter parameters

`declaration_date`

Query by declaration date with the format YYYY-MM-DD.

Additional filter parameters

pay_date

Query by pay date with the format YYYY-MM-DD.

Additional filter parameters

frequency

Query by the number of times per year the dividend is paid out. Possible values are 0 (one-time), 1 (annually), 2 (bi-annually), 4 (quarterly), and 12 (monthly).

cash_amount

Query by the cash amount of the dividend.

Additional filter parameters

dividend_type

Query by the type of dividend. Dividends that have been paid and/or are expected to be paid on consistent schedules are denoted as CD. Special Cash dividends that have been paid that are infrequent or unusual, and/or can not be expected to occur in the future are denoted as SC.

order

Order results based on the `sort` field.

limit

Limit the number of results returned, default is 10 and max is 1000.

sort

Sort field used for ordering.

```
https://api.polygon.io/v3/reference/dividends?apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

Run Query

Response Attributes

`next_url` string

If present, this value can be used to fetch the next page of data.

`request_id*` string

`results` array

`cash_amount*` number

The cash amount of the dividend per share owned.

`currency` string

The currency in which the dividend is paid.

`declaration_date` string

The date that the dividend was announced.

`dividend_type*` enum [CD, SC, LT, ST]

The type of dividend. Dividends that have been paid and/or are expected to be paid on consistent schedules are denoted as CD. Special Cash dividends that have been paid that are infrequent or unusual, and/or can not be expected to occur in the future are denoted as SC. Long-Term and Short-Term capital gain distributions are denoted as LT and ST, respectively.

`ex_dividend_date*` string

The date that the stock first trades without the dividend, determined by the exchange.

`frequency*` integer

The number of times per year the dividend is paid out. Possible values are 0 (one-time), 1 (annually), 2 (bi-annually), 4 (quarterly), and 12 (monthly).

`pay_date` string

The date that the dividend is paid out.

`record_date` string

The date that the stock must be held to receive the dividend, set by the company.

`ticker*` string

The ticker symbol of the dividend.

`status` string

Was this helpful?

Help us improve

Yes No

Response Object

```
{
  "next_url": "https://api.polygon.io/v3/reference/dividends/AAPL?cursor=YWN0aXZlPXRydWUm
  "results": [
    {
      "cash_amount": 0.22,
      "declaration_date": "2021-10-28",
      "dividend_type": "CD",
      "ex_dividend_date": "2021-11-05",
      "frequency": 4,
      "pay_date": "2021-11-11",
      "record_date": "2021-11-08",
      "ticker": "AAPL"
    },
    {
      "cash_amount": 0.22,
      "declaration_date": "2021-07-27",
      "dividend_type": "CD",
      "ex_dividend_date": "2021-08-06",
      "frequency": 4,
      "pay_date": "2021-08-12",
      "record_date": "2021-08-09",
      "ticker": "AAPL"
    }
  ],
  "status": "OK"
}
```


Stock Financials vX

GET /vX/reference/financials

Get historical financial data for a stock ticker. The financials data is extracted from XBRL from company SEC filings using the methodology outlined here.

This API is experimental.

Parameters

ticker	

Query by company ticker.

`cik`

Query by central index key (CIK) Number

`company_name`

Query by company name.

Additional filter parameters

`sic`

Query by standard industrial classification (SIC)

`filing_date`

Query by the date when the filing with financials data was filed in YYYY-MM-DD format.

Best used when querying over date ranges to find financials based on filings that happen in a time period.

Examples:

To get financials based on filings that have happened after January 1, 2009 use the query param filing_date.gte=2009-01-01

To get financials based on filings that happened in the year 2009 use the query params filing_date.gte=2009-01-01&filing_date.lt=2010-01-01

Additional filter parameters

`period_of_report_date`

The period of report for the filing with financials data in YYYY-MM-DD format.

Additional filter parameters

`timeframe`

Query by timeframe. Annual financials originate from 10-K filings, and quarterly financials originate from 10-Q filings. Note: Most companies do not file quarterly reports for Q4 and instead include those financials in their annual report, so some companies my not return quarterly financials for Q4

`include_sources`

Whether or not to include the `xpath` and `formula` attributes for each financial data point. See the `xpath` and `formula` response attributes for more info. False by default.

`order`

Order results based on the `sort` field.

`limit`

Limit the number of results returned, default is 10 and max is 100.

sort

Sort field used for ordering.

```
https://api.polygon.io/vX/reference/financials?apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

Run Query

Response Attributes

count* integer

The total number of results for this request.

---

next_url string

If present, this value can be used to fetch the next page of data.

---

request_id* string

A request id assigned by the server.

---

results* array

acceptance_datetime

The datetime (EST timezone) the filing was accepted by EDGAR in YYYYMMDDHHMMSS format.

---

cik* string

The CIK number for the company.

---

company_name* string

The company name.

---

end_date string

The end date of the period that these financials cover in YYYYMMDD format.

---

filing_date

The date that the SEC filing which these financials were derived from was made available. Note that this is not necessarily the date when this information became public, as some companies may publish a press release before filing with the SEC.

---

financials* object

`balance_sheet` object

Balance sheet. The keys in this object can be any of the fields listed in the Balance Sheet section of the financials API glossary of terms.

`*` object

An individual financial data point.

`derived_from` array [string]

The list of report IDs (or errata) which were used to derive this data point. This value is only returned for data points taken directly from XBRL when the `include_sources` query parameter is `true` and if source is SourceInterReportDerived.

`formula` string

The name of the formula used to derive this data point from other financial data points. Information about the formulas can be found here. This value is only returned for data points that are not explicitly expressed within the XBRL source file when the `include_sources` query parameter is `true` and if source is SourceIntraReportImpute.

`label*` string

A human readable label for the financial data point.

`order*` integer

An indicator of what order within the statement that you would find this data point.

`source`

The source where this data point came from. This will be one of: SourceDirectReport, SourceIntraReportImpute or SourceInterReportDerived.

`unit*` string

The unit of the financial data point.

`value*` number

The value of the financial data point.

`xpath` string

The XPath 1.0 query that identifies the fact from within the XBRL source file. This value is only returned for data points taken directly from XBRL when the `include_sources` query parameter is `true` and if source is SourceDirectReport.

`cash_flow_statement` object

Cash flow statement. The keys in this object can be any of the fields listed in the Cash Flow Statement section of the financials API glossary of terms. See the attributes of the objects within `balance_sheet` for more details.

`comprehensive_income` object

Comprehensive income. The keys in this object can be any of the fields listed in the Comprehensive Income section of the financials API glossary of terms. See the attributes of the objects within `balance_sheet` for more details.

`income_statement` object

Income statement. The keys in this object can be any of the fields listed in the Income Statement section of the financials API glossary of terms. See the attributes of the objects within `balance_sheet` for more details.

`fiscal_period*` string

Fiscal period of the report according to the company (Q1, Q2, Q3, Q4, or FY).

`fiscal_year` string

Fiscal year of the report according to the company.

`source_filing_file_url`

The URL of the specific XBRL instance document within the SEC filing that these financials were derived from.

`source_filing_url` string

The URL of the SEC filing that these financials were derived from.

`start_date` string

The start date of the period that these financials cover in YYYYMMDD format.

`tickers` array [string]

The list of ticker symbols for the company.

`timeframe*` string

The timeframe of the report (quarterly, annual or ttm).

`status*` string

The status of this request's response.

Was this helpful?

Help us improve

Yes No

Response Object

```
{
  "count": 1,
  "next_url": "https://api.polygon.io/vX/reference/financials?",
```

```
  "request_id": "55eb92ed43b25568ab0cce159830ea34",
  "results": [
    {
      "cik": "0001650729",
      "company_name": "SiteOne Landscape Supply, Inc.",
      "end_date": "2022-04-03",
      "filing_date": "2022-05-04",
      "financials": {
        "balance_sheet": {
          "assets": {
            "label": "Assets",
            "order": 100,
            "unit": "USD",
            "value": 2407400000
          },
          "current_assets": {
            "label": "Current Assets",
            "order": 200,
            "unit": "USD",
            "value": 1385900000
          },
          "current_liabilities": {
            "label": "Current Liabilities",
            "order": 700,
            "unit": "USD",
            "value": 597500000
          },
          "equity": {
            "label": "Equity",
            "order": 1400,
            "unit": "USD",
            "value": 1099200000
          },
          "equity_attributable_to_noncontrolling_interest": {
            "label": "Equity Attributable To Noncontrolling Interest",
            "order": 1500,
            "unit": "USD",
            "value": 0
          },
          "equity_attributable_to_parent": {
            "label": "Equity Attributable To Parent",
            "order": 1600,
            "unit": "USD",
            "value": 1099200000
          },
          "liabilities": {
            "label": "Liabilities",
            "order": 600,
            "unit": "USD",
            "value": 1308200000
          },
          "liabilities_and_equity": {
```

```
        "label": "Liabilities And Equity",
        "order": 1900,
        "unit": "USD",
        "value": 2407400000
      },
      "noncurrent_assets": {
        "label": "Noncurrent Assets",
        "order": 300,
        "unit": "USD",
        "value": 1021500000
      },
      "noncurrent_liabilities": {
        "label": "Noncurrent Liabilities",
        "order": 800,
        "unit": "USD",
        "value": 710700000
      }
    },
    "cash_flow_statement": {
      "exchange_gains_losses": {
        "label": "Exchange Gains/Losses",
        "order": 1000,
        "unit": "USD",
        "value": 100000
      },
      "net_cash_flow": {
        "label": "Net Cash Flow",
        "order": 1100,
        "unit": "USD",
        "value": -8600000
      },
      "net_cash_flow_continuing": {
        "label": "Net Cash Flow, Continuing",
        "order": 1200,
        "unit": "USD",
        "value": -8700000
      },
      "net_cash_flow_from_financing_activities": {
        "label": "Net Cash Flow From Financing Activities",
        "order": 700,
        "unit": "USD",
        "value": 150600000
      },
      "net_cash_flow_from_financing_activities_continuing": {
        "label": "Net Cash Flow From Financing Activities, Continuing",
        "order": 800,
        "unit": "USD",
        "value": 150600000
      },
      "net_cash_flow_from_investing_activities": {
        "label": "Net Cash Flow From Investing Activities",
        "order": 400,
```

```
      "unit": "USD",
      "value": -41000000
    },
    "net_cash_flow_from_investing_activities_continuing": {
      "label": "Net Cash Flow From Investing Activities, Continuing",
      "order": 500,
      "unit": "USD",
      "value": -41000000
    },
    "net_cash_flow_from_operating_activities": {
      "label": "Net Cash Flow From Operating Activities",
      "order": 100,
      "unit": "USD",
      "value": -118300000
    },
    "net_cash_flow_from_operating_activities_continuing": {
      "label": "Net Cash Flow From Operating Activities, Continuing",
      "order": 200,
      "unit": "USD",
      "value": -118300000
    }
  },
  "comprehensive_income": {
    "comprehensive_income_loss": {
      "label": "Comprehensive Income/Loss",
      "order": 100,
      "unit": "USD",
      "value": 40500000
    },
    "comprehensive_income_loss_attributable_to_noncontrolling_interest": {
      "label": "Comprehensive Income/Loss Attributable To Noncontrolling Interest",
      "order": 200,
      "unit": "USD",
      "value": 0
    },
    "comprehensive_income_loss_attributable_to_parent": {
      "label": "Comprehensive Income/Loss Attributable To Parent",
      "order": 300,
      "unit": "USD",
      "value": 40500000
    },
    "other_comprehensive_income_loss": {
      "label": "Other Comprehensive Income/Loss",
      "order": 400,
      "unit": "USD",
      "value": 40500000
    },
    "other_comprehensive_income_loss_attributable_to_parent": {
      "label": "Other Comprehensive Income/Loss Attributable To Parent",
      "order": 600,
      "unit": "USD",
      "value": 8200000
```

```
    }
  },
  "income_statement": {
    "basic_earnings_per_share": {
      "label": "Basic Earnings Per Share",
      "order": 4200,
      "unit": "USD / shares",
      "value": 0.72
    },
    "benefits_costs_expenses": {
      "label": "Benefits Costs and Expenses",
      "order": 200,
      "unit": "USD",
      "value": 768400000
    },
    "cost_of_revenue": {
      "label": "Cost Of Revenue",
      "order": 300,
      "unit": "USD",
      "value": 536100000
    },
    "costs_and_expenses": {
      "label": "Costs And Expenses",
      "order": 600,
      "unit": "USD",
      "value": 768400000
    },
    "diluted_earnings_per_share": {
      "label": "Diluted Earnings Per Share",
      "order": 4300,
      "unit": "USD / shares",
      "value": 0.7
    },
    "gross_profit": {
      "label": "Gross Profit",
      "order": 800,
      "unit": "USD",
      "value": 269200000
    },
    "income_loss_from_continuing_operations_after_tax": {
      "label": "Income/Loss From Continuing Operations After Tax",
      "order": 1400,
      "unit": "USD",
      "value": 32300000
    },
    "income_loss_from_continuing_operations_before_tax": {
      "label": "Income/Loss From Continuing Operations Before Tax",
      "order": 1500,
      "unit": "USD",
      "value": 36900000
    },
    "income_tax_expense_benefit": {
```

```
    "label": "Income Tax Expense/Benefit",
    "order": 2200,
    "unit": "USD",
    "value": 4600000
  },
  "interest_expense_operating": {
    "label": "Interest Expense, Operating",
    "order": 2700,
    "unit": "USD",
    "value": 4300000
  },
  "net_income_loss": {
    "label": "Net Income/Loss",
    "order": 3200,
    "unit": "USD",
    "value": 32300000
  },
  "net_income_loss_attributable_to_noncontrolling_interest": {
    "label": "Net Income/Loss Attributable To Noncontrolling Interest",
    "order": 3300,
    "unit": "USD",
    "value": 0
  },
  "net_income_loss_attributable_to_parent": {
    "label": "Net Income/Loss Attributable To Parent",
    "order": 3500,
    "unit": "USD",
    "value": 32300000
  },
  "net_income_loss_available_to_common_stockholders_basic": {
    "label": "Net Income/Loss Available To Common Stockholders, Basic",
    "order": 3700,
    "unit": "USD",
    "value": 32300000
  },
  "operating_expenses": {
    "label": "Operating Expenses",
    "order": 1000,
    "unit": "USD",
    "value": 228000000
  },
  "operating_income_loss": {
    "label": "Operating Income/Loss",
    "order": 1100,
    "unit": "USD",
    "value": 41200000
  },
  "participating_securities_distributed_and_undistributed_earnings_loss_basic": {
    "label": "Participating Securities, Distributed And Undistributed Earnings/Lc
    "order": 3800,
    "unit": "USD",
    "value": 0
```

```
          },
          "preferred_stock_dividends_and_other_adjustments": {
            "label": "Preferred Stock Dividends And Other Adjustments",
            "order": 3900,
            "unit": "USD",
            "value": 0
          },
          "revenues": {
            "label": "Revenues",
            "order": 100,
            "unit": "USD",
            "value": 805300000
          }
        }
      },
      "fiscal_period": "Q1",
      "fiscal_year": "2022",
      "source_filing_file_url": "https://api.polygon.io/v1/reference/sec/filings/00016507
      "source_filing_url": "https://api.polygon.io/v1/reference/sec/filings/0001650729-22
      "start_date": "2022-01-03"
    }
  ],
  "status": "OK"
}
```


Conditions

GET /v3/reference/conditions

List all conditions that Polygon.io uses.

Parameters

asset_class	stocks

Filter for conditions within a given asset class.

data_type	

Filter by data type.

id	

Filter for conditions with a given ID.

sip

Filter by SIP. If the condition contains a mapping for that SIP, the condition will be returned.

order

Order results based on the `sort` field.

limit

Limit the number of results returned, default is 10 and max is 1000.

sort

Sort field used for ordering.

```
https://api.polygon.io/v3/reference/conditions?asset_class=stocks&apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

JSON

Run Query

Response Attributes

`count`* integer

The total number of results for this request.

`next_url` string

If present, this value can be used to fetch the next page of data.

`request_id`* string

A request ID assigned by the server.

`results`* array

An array of conditions that match your query.

> `abbreviation` string
>
> A commonly-used abbreviation for this condition.
>
> `asset_class`* enum [stocks, options, crypto, fx]
>
> An identifier for a group of similar financial instruments.
>
> `data_types`* array [string]
>
> Data types that this condition applies to.

`description` string

A short description of the semantics of this condition.

`exchange` integer

If present, mapping this condition from a Polygon.io code to a SIP symbol depends on this attribute. In other words, data with this condition attached comes exclusively from the given exchange.

`id*` integer

An identifier used by Polygon.io for this condition. Unique per data type.

`legacy` boolean

If true, this condition is from an old version of the SIPs' specs and no longer is used. Other conditions may or may not reuse the same symbol as this one.

`name*` string

The name of this condition.

`sip_mapping*` object

A mapping to a symbol for each SIP that has this condition.

- `CTA` string
- `OPRA` string
- `UTP` string

`type*` enum [sale_condition, quote_condition, sip_generated_flag, financial_status_indicator, short_sale_restriction_indicator, settlement_condition, market_condition, trade_thru_exempt]

An identifier for a collection of related conditions.

`update_rules` object

A list of aggregation rules.

- `consolidated*` object

  Describes aggregation rules on a consolidated (all exchanges) basis.

  - `updates_high_low*` boolean

    Whether or not trades with this condition update the high/low.

  - `updates_open_close*` boolean

    Whether or not trades with this condition update the open/close.

`updates_volume`* boolean

Whether or not trades with this condition update the volume.

`market_center`* object

Describes aggregation rules on a per-market-center basis.

`updates_high_low`* boolean

Whether or not trades with this condition update the high/low.

`updates_open_close`* boolean

Whether or not trades with this condition update the open/close.

`updates_volume`* boolean

Whether or not trades with this condition update the volume.

`status`* string

The status of this request's response.

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "count": 1,
  "request_id": "31d59dda-80e5-4721-8496-d0d32a654afe",
  "results": [
    {
      "asset_class": "stocks",
      "data_types": [
        "trade"
      ],
      "id": 2,
      "name": "Average Price Trade",
      "sip_mapping": {
        "CTA": "B",
        "UTP": "W"
      },
      "type": "condition",
      "update_rules": {
        "consolidated": {
          "updates_high_low": false,
          "updates_open_close": false,
          "updates_volume": true
        },
        "market_center": {
```

```
            "updates_high_low": false,
            "updates_open_close": false,
            "updates_volume": true
          }
        }
      }
    ],
    "status": "OK"
  }
```


Exchanges

GET /v3/reference/exchanges

List all exchanges that Polygon.io knows about.

Parameters

asset_class | stocks

Filter by asset class.

locale |

Filter by locale.

```
https://api.polygon.io/v3/reference/exchanges?asset_class=stocks&apiKey=1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj
```

Copy

JSON

Run Query

Response Attributes

`count` integer

The total number of results for this request.

`request_id`* string

A request ID assigned by the server.

`results` array

`acronym` string

A commonly used abbreviation for this exchange.

`asset_class`* enum [stocks, options, crypto, fx]

An identifier for a group of similar financial instruments.

`id`* integer

A unique identifier used by Polygon.io for this exchange.

`locale`* enum [us, global]

An identifier for a geographical location.

`mic` string

The Market Identifer Code of this exchange (see ISO 10383).

`name`* string

Name of this exchange.

`operating_mic` string

The MIC of the entity that operates this exchange.

`participant_id` string

The ID used by SIP's to represent this exchange.

`type`* enum [exchange, TRF, SIP]

Represents the type of exchange.

`url` string

A link to this exchange's website, if one exists.

`status`* string

The status of this request's response.

Was this helpful?

Help us improve

Yes No

Response Object

```
{
  "count": 1,
  "request_id": "31d59dda-80e5-4721-8496-d0d32a654afe",
  "results": [
    {
      "acronym": "AMEX",
      "asset_class": "stocks",
      "id": 1,
      "locale": "us",
      "mic": "XASE",
      "name": "NYSE American, LLC",
      "operating_mic": "XNYS",
      "participant_id": "A",
      "type": "exchange",
      "url": "https://www.nyse.com/markets/nyse-american"
    }
  ],
  "status": "OK"
}
```


Stocks WebSocket Documentation

Requires a "Stocks Basic" subscription

Upgrade

The Polygon.io Stocks WebSocket API provides streaming access to the latest stock market data from all US stock exchanges. You can specify which channels you want to consume by sending instructions in the form of actions. Our WebSockets emit events to notify you when an event has occurred in a channel you've subscribed to.

Our WebSocket APIs are based on entitlements that control which WebSocket Clusters you can connect to and which kinds of data you can access. Examples in these docs include your API key, which only you can see, and are personalized based on your entitlements.

Step 1: Connect

Your current plan doesn't include access to the Stocks WebSocket Cluster. Upgrade to get access.

Connecting to a cluster:

Delayed: `wscat -c wss://delayed.polygon.io/stocks`

Copy

Real-time: `wscat -c wss://socket.polygon.io/stocks`

Copy

On connection you will receive the following message:

```
[{
        "ev":"status",
        "status":"connected",
        "message": "Connected Successfully"
}]
```

Step 2: Authenticate

You must authenticate before you can make any other requests.

```
{"action":"auth","params":"1RQBxJjR6uKqzsfPUpWtrHCSsFMOwSjj"}
```

Copy

On successful authentication you will receive the following message:

```
[{
        "ev":"status",
        "status":"auth_success",
        "message": "authenticated"
}]
```

Step 3: Subscribe

Once authenticated, you can request a stream. You can request multiple streams in the same request.

```
{"action":"subscribe","params":"AM.LPL"}
```

Copy

You can also request multiple streams from the same cluster.

```
{"action":"subscribe","params":"AM.LPL,AM.MSFT"}
```

Copy

Usage

Things happen very quickly in the world of finance, which means a Polygon.io WebSocket client must be able to handle many incoming messages per second. Due to the nature of the WebSocket protocol, if a client is slow to consume messages from the server, Polygon.io's server must buffer messages and send them only as fast as the client can consume them. To help prevent the message buffer from getting too long, Polygon.io may send more than one JSON object in a single WebSocket message. We accomplish this by wrapping all messages in a JSON array, and adding more objects to the array if the message buffer is getting longer. For example, consider a WebSocket message with a single trade event in it:

```
[
    {"ev":"T","sym":"MSFT","i":"50578","x":4,"p":215.9721,"s":100,"t":1611082428813,"z":3}
]
```

If your client is consuming a bit slow, or 2+ events happened in very short succession, you may receive a single WebSocket message with more than one event inside it, like this:

```
[
    {"ev":"T","sym":"MSFT","i":"50578","x":4,"p":215.9721,"s":100,"t":1611082428813,"z":3},
    {"ev":"T","sym":"MSFT","i":"12856","x":4,"p":215.989,"s":1,"c":[37],"t":1611082428814,"z":3
]
```

Note that if a client is consuming messages too slowly for too long, Polygon.io's server-side buffer may get too large. If that happens, Polygon.io will terminate the WebSocket connection. You can check your account dashboard to see if a connection was terminated as a slow consumer. If this happens to you consistently, consider subscribing to fewer symbols or channels.

Your Plan

Upgrade to get access.

Upgrade

Client Libraries

Python
client-python

Go
client-go

Javascript
client-js

PHP
client-php

Kotlin
client-jvm

Aggregates (Per Minute)

WS
Delayed: wss://delayed.polygon.io/stocks
Real-Time: wss://socket.polygon.io/stocks

Stream real-time minute aggregates for a given stock ticker symbol.

Requires a "Stocks Starter" subscription Upgrade

Parameters

ticker*	*

Specify a stock ticker or use * to subscribe to all stock tickers. You can also use a comma separated list to subscribe to multiple stock tickers. You can retrieve available stock tickers from our Stock Tickers API.

```
{"action":"subscribe", "params":"AM.*"}
```

Copy

Response Attributes

`ev` enum [AM]

The event type.

`sym` string

The ticker symbol for the given stock.

`v` integer

The tick volume.

`av` integer

Today's accumulated volume.

`op` number

Today's official opening price.

`vw` number

The tick's volume weighted average price.

`o` number

The opening tick price for this aggregate window.

`c` number

The closing tick price for this aggregate window.

`h` number

The highest tick price for this aggregate window.

`l` number

The lowest tick price for this aggregate window.

`a` number

Today's volume weighted average price.

`z` integer

The average trade size for this aggregate window.

`s` integer

The timestamp of the starting tick for this aggregate window in Unix Milliseconds.

`e` integer

The timestamp of the ending tick for this aggregate window in Unix Milliseconds.

`otc` boolean

Whether or not this aggregate is for an OTC ticker. This field will be left off if false.

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "ev": "AM",
  "sym": "GTE",
  "v": 4110,
  "av": 9470157,
  "op": 0.4372,
  "vw": 0.4488,
  "o": 0.4488,
  "c": 0.4486,
  "h": 0.4489,
  "l": 0.4486,
  "a": 0.4352,
  "z": 685,
  "s": 1610144640000,
  "e": 1610144700000
}
```


Aggregates (Per Second)

WS Delayed: wss://delayed.polygon.io/stocks
Real-Time: wss://socket.polygon.io/stocks

Stream real-time second aggregates for a given stock ticker symbol.

Requires a "Stocks Starter" subscription Upgrade

Parameters

ticker*	*

Specify a stock ticker or use * to subscribe to all stock tickers. You can also use a comma separated list to subscribe to multiple stock tickers. You can retrieve available stock tickers from our Stock Tickers API.

```
{"action":"subscribe", "params":"A.*"}
```

Copy

Response Attributes

`ev` enum [A]

The event type.

`sym` string

The ticker symbol for the given stock.

`v` integer

The tick volume.

`av` integer

Today's accumulated volume.

`op` number

Today's official opening price.

`vw` number

The tick's volume weighted average price.

`o` number

The opening tick price for this aggregate window.

`c` number

The closing tick price for this aggregate window.

`h` number

The highest tick price for this aggregate window.

`l` number

The lowest tick price for this aggregate window.

`a` number

Today's volume weighted average price.

`z` integer

The average trade size for this aggregate window.

`s` integer

The timestamp of the starting tick for this aggregate window in Unix Milliseconds.

`e` integer

The timestamp of the ending tick for this aggregate window in Unix Milliseconds.

`otc` boolean

Whether or not this aggregate is for an OTC ticker. This field will be left off if false.

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "ev": "A",
  "sym": "SPCE",
  "v": 200,
  "av": 8642007,
  "op": 25.66,
  "vw": 25.3981,
  "o": 25.39,
  "c": 25.39,
  "h": 25.39,
  "l": 25.39,
  "a": 25.3714,
  "z": 50,
  "s": 1610144868000,
  "e": 1610144869000
}
```


Trades

WS

Delayed: wss://delayed.polygon.io/stocks

Real-Time: wss://socket.polygon.io/stocks

Stream real-time trades for a given stock ticker symbol.

Requires a "Stocks Developer" subscription

Upgrade

Parameters

ticker*	*

Specify a stock ticker or use * to subscribe to all stock tickers. You can also use a comma separated list to subscribe to multiple stock tickers. You can retrieve available stock tickers from our Stock Tickers API.

```
{"action":"subscribe", "params":"T.*"}
```

Copy

Response Attributes

`ev` enum [T]

The event type.

`sym` string

The ticker symbol for the given stock.

`x` integer

The exchange ID. See Exchanges for Polygon.io's mapping of exchange IDs.

`i` string

The trade ID.

`z` integer

The tape. (1 = NYSE, 2 = AMEX, 3 = Nasdaq).

`p` number

The price.

`s` integer

The trade size.

`c` array [integer]

The trade conditions. See Conditions and Indicators" for Polygon.io's trade conditions glossary.

`t` integer

The Timestamp in Unix MS.

`q` integer

The sequence number represents the sequence in which message events happened. These are increasing and unique per ticker symbol, but will not always be sequential (e.g., 1, 2, 6, 9, 10, 11).

`trfi` integer

The ID for the Trade Reporting Facility where the trade took place.

`trft` integer

The TRF (Trade Reporting Facility) Timestamp in Unix MS. This is the timestamp of when the trade reporting facility received this trade.

Was this helpful?

Help us improve

Yes No

Response Object

```
{
  "ev": "T",
  "sym": "MSFT",
  "x": 4,
  "i": "12345",
  "z": 3,
  "p": 114.125,
  "s": 100,
  "c": [
    0,
    12
  ],
  "t": 1536036818784,
  "q": 3681328
}
```

Quotes

WS Delayed: wss://delayed.polygon.io/stocks

Real-Time: wss://socket.polygon.io/stocks

Stream real-time quotes for a given stock ticker symbol.

Requires a "Stocks Advanced" subscription

Upgrade

Parameters

ticker*	*

Specify a stock ticker or use * to subscribe to all stock tickers. You can also use a comma separated list to subscribe to multiple stock tickers. You can retrieve available stock tickers from our Stock Tickers API.

```
{"action":"subscribe", "params":"Q.*"}
```

Copy

Response Attributes

ev enum [Q]

The event type.

---

sym string

The ticker symbol for the given stock.

---

bx integer

The bid exchange ID.

---

bp number

The bid price.

---

bs integer

The bid size. This represents the number of round lot orders at the given bid price. The normal round lot size is 100 shares. A bid size of 2 means there are 200 shares for purchase at the given bid price.

---

ax integer

The ask exchange ID.

---

ap number

The ask price.

`as` integer

The ask size. This represents the number of round lot orders at the given ask price. The normal round lot size is 100 shares. An ask size of 2 means there are 200 shares available to purchase at the given ask price.

`c` integer

The condition.

`i` array [integer]

The indicators. For more information, see our glossary of Conditions and Indicators.

`t` integer

The Timestamp in Unix MS.

`q` integer

The sequence number represents the sequence in which quote events happened. These are increasing and unique per ticker symbol, but will not always be sequential (e.g., 1, 2, 6, 9, 10, 11). Values reset after each trading session/day.

`z` integer

The tape. (1 = NYSE, 2 = AMEX, 3 = Nasdaq).

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "ev": "Q",
  "sym": "MSFT",
  "bx": 4,
  "bp": 114.125,
  "bs": 100,
  "ax": 7,
  "ap": 114.128,
  "as": 160,
  "c": 0,
  "i": [
    604
  ],
  "t": 1536036818784,
  "q": 50385480,
  "z": 3
}
```

Fair Market Value

WS Business: wss://business.polygon.io/stocks

Real-time fair market value for a given stock ticker symbol.

Requires a "Stocks Enterprise" subscription Upgrade

Parameters

ticker*	*

Specify a stock ticker or use * to subscribe to all stock tickers. You can also use a comma separated list to subscribe to multiple stock tickers. You can retrieve available stock tickers from our Stock Tickers API.

```
{"action":"subscribe", "params":"FMV.*"}
```

Copy

Response Attributes

`ev` enum [FMV]

The event type.

`fmv`

Fair market value is only available on Business plans. It is our proprietary algorithm to generate a real-time, accurate, fair market value of a tradable security. For more information, contact us.

`sym`

The ticker symbol for the given security.

`t`

The nanosecond timestamp.

Was this helpful?

Help us improve

Yes No

Response Object


```
{
  "ev": "FMV",
  "val": 189.22,
  "sym": "AAPL",
  "t": 1678220098130
}
```


Products

Stocks

Options

Indices

Forex

Crypto

Docs

Stocks API

Options API

Indices API

Forex API

Crypto API

Launchpad Docs

Resources

Client Libraries

Roadmap

Knowledge Base

System Status

Company

About

Blog

Careers

Contact Us

Pricing

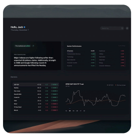
View Pricing

March 11, 2024

Integration: everviz

March 6, 2024

Historical File Downloads Included In All Plans – No Really



December 11, 2023

Case Study: Fey

June 20, 2023

Track Movements Across Markets with our New Universal Snapshot API

Terms of Service

Privacy Policy