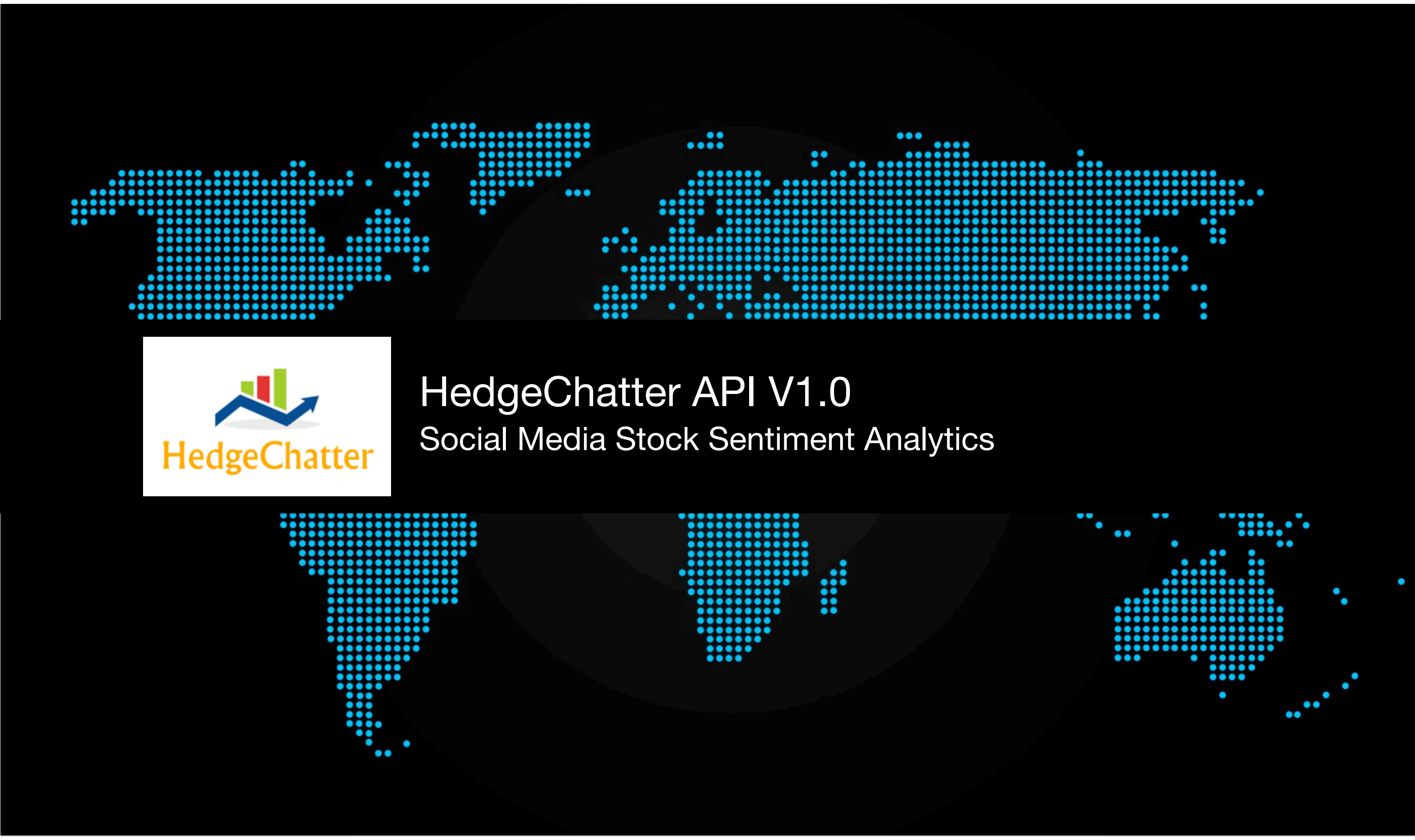
HedgeChatter

API Specs v1.0, 2015



HedgeChatter API V1.0

Social Media Stock Sentiment Analytics

API Types:

The API data comes as three products which are tailored for different types of subscribers. A subscriber can opt to receive one or more of these products.

The three APIs include:

1) Dashboard APIs
2) Sentiment APIs
3) Alert APIs

AUTHENTICATION:

All APIs use OAuth 2.0 for authentication. The subscriber is given a client token and a secret token through the HedgeChatter site and can use these in the API requests to authenticate. See the 'Authentication API' section at the end of this document for more details.

FORMAT:

The JSON format is used for all APIs. A post method is used to make a request with the body of the post containing a valid JSON string. The response from the API is also in JSON format.

TOKEN:

All requests must have an “**action**” field that determines the API being invoked. Also all requests must have an “**access_token**” field which is obtained using the authentication API. This field is not mentioned in the 'Request' section of the APIs since it is common to all of them.

RESPONSE:

All responses have an “**error**” field that is set to 0 or 1. A “**message**” field that is set to “success” or the error message. A “**result**” field that contains an object with the requested data. The API response details below show only the fields in the “result” and not the common fields that are in all responses.

Dashboard APIs

This API is intended for HedgeChatter partners who want to display data similar to that on the HedgeChatter site, but on their own site for their customers, a licensing agreement will need to be in place to recreate the HedgeChatter dashboard.

Get Stock Dashboard

Request

action = “getStockDashboard”

symbol - Stock ticker symbol

data = [value1, value2, ...]
An array of values specifying what data to return. Allowed values are: **chatter, price, volume, news, sentiment, manipulation**

start = YYYYMMDD

Specifies the date for the start of data. The minimum date allowed is 20120601. Optional. If not specified than June 1st, 2012 is used or the stock IPO date which ever is later.

end = YYYYMMDD

Specifies the date for the end of data; inclusive. Optional. If not specified the most current trading day of the stock is used.

Response
Only the requested data objects are returned.

chatter = [{object}, {object}, ...]
timestamp - Unix timestamp in UTC
value - Number of chatter message

price = [{object}, {object}, ...]
timestamp - Unix timestamp in UTC
value - Stock adjusted closing price on day of timestamp

volume = [{object}, {object}, ...]
timestamp - Unix timestamp in UTC
volume - Stock trading volume

news = [{object}, {object}, ...]
    timestamp - Unix timestamp in UTC of when the news item was saved
    title - Title of the news article
    link - Link to the news article

sentiment = [{object}, {object}, ...]
    timestamp - Unix timestamp in UTC
    sbuy - Percent of users expressing strong buy buy - Percent of users expressing buy
    hold - Percent of users expressing hold
    sell - Percent of users expressing sell
    ssell - Percent of users expressing strong sell

manipulation = [{object}, {object}, ...]
    timestamp - Unix timestamp in UTC
    good - Percent of messages with good content
    bad - Percent of messages with questionable content

Get Stock Dashboard Update

Request

action = “getStockDashboardUpdate”

data = [value1, value2, ...]
An array of values specifying what data to return. Allowed values are: **chatter, price, volume, news, sentiment, manipulation**

date = YYYYMMDD
Specifies the date for the returned data. The minimum date allowed is 20120601. Optional. If not specified than the most current date is returned.

Response

data = [{object}, {object}, ...]
The array is sorted by symbol in ascending order.

symbol = Symbol of the corresponding data
Only the requested data objects are returned.

chatter = {object}
timestamp - Unix timestamp in UTC
value - Number of chatter message

price = {object}
timestamp - Unix timestamp in UTC
value - Stock adjusted closing price on day of timestamp

volume = {object}
timestamp - Unix timestamp in UTC volume - Stock trading volume

news = {object}
timestamp - Unix timestamp in UTC of when the news item was saved title - Title of the news article
link - Link to the news article

sentiment = {object}
- timestamp - Unix timestamp in UTC
- sbuy - Percent of users expressing strong buy buy - Percent of users expressing buy
- hold - Percent of users expressing hold
- sell - Percent of users expressing sell
- ssell - Percent of users expressing strong sell

manipulation = {object}
- timestamp - Unix timestamp in UTC
- good - Percent of messages with good content
- bad - Percent of messages with questionable content

Get Market Dashboard

Request

action = "getMarketDashboard"

data = [value1, value2, ...]

An array of values specifying what data to return. Allowed values are: **chatterVolume, sentimentVolume, negativeSentiment, positiveSentiment, sentimentGainers, sentimentLosers, newsVolume, newsSpike, twitterVolume, twitterSpike, accurateAlerts**

top = Number of items to get; applies to all lists. Max possible is 100.

date = YYYYMMDD

The date of the data. Optional. If not provided the most current date is used. The earliest date for this API is 20131001 or October 1st, 2013.

Response

Only the requested data objects are returned. The arrays are sorted with lower rank numbers first.

chatterVolume = [{object}, {object}, ...]
- symbol - Stock ticker symbol.
- rank - Rank in the list (lower is better)
- volume - Amount of chatter

sentimentVolume = [{object}, {object}, ...]
- symbol - Stock ticker symbol.
- rank - Rank in the list (lower is better)
- volume - Amount of chatter expressing sentiment

positiveSentiment = [{object}, {object}, ...]
- symbol - Stock ticker symbol.
- rank - Rank in the list (lower is better)
- positive - Amount of positive chatter
- negative - Amount of negative chatter

negativeSentiment = [{object}, {object}, ...]
- symbol - Stock ticker symbol.
- rank - Rank in the list (lower is better)
- positive - Amount of positive chatter
- negative - Amount of negative chatter

sentimentGainers = [{object}, {object}, ...]
    symbol - Stock ticker symbol.
    rank - Rank in the list (lower is better)
    volume - Amount of positive chatter change

sentimentLosers = [{object}, {object}, ...]
    symbol - Stock ticker symbol.
    rank - Rank in the list (lower is better)
    volume - Amount of negative chatter change

newsVolume = [{object}, {object}, ...]
    symbol - Stock ticker symbol.
    rank - Rank in the list (lower is better)
    volume - Amount of news coverage

newsSpike = [{object}, {object}, ...]
    symbol - Stock ticker symbol.
    rank - Rank in the list (lower is better)
    gain - Percent gain in news coverage

twitterVolume = [{object}, {object}, ...]
    symbol - Stock ticker symbol.
    rank - Rank in the list (lower is better)
    volume - Amount of Twitter chatter

twitterSpike = [{object}, {object}, ...]
    symbol - Stock ticker symbol
    rank - Rank in the list (lower is better)
    gain - Percent gain in twitter messages

accurateAlerts = [{object}, {object}, ...]
    symbol - Stock ticker symbol
    rank - Rank in the list (lower is better)
    alerts - Number of alerts
    accuracy - Percent of alerts correctly predicted by "positive sentiment reputation" alerts
    high - maximum percent gain within 60 days after the alert
    low - maximum percent loss within 60 days after the alert

Sentiment APIs

This API is intended for hedge funds and other financial firms that want to combine sentiment data along with traditional data to generate their own trading signals.

Get Sentiment

Request

action = “getSentiment”

symbols - [symbol1, symbol2, ...]

If not provided then data is return for all symbols available on the start date.

data = [value1, value2, ...]

An array of values specifying what data to return. If not specified then all fields are returned. Allowed values are:
num - number of chatter messages
nsent – number of chatter messages with sentiment
tsent – weighted total of expressed sentiment
unum - unique number of chatter messages
unsent – unique number of chatter messages with sentiment
utsent - unique messages weighted total of express sentiment
sbuy - number of strong buy
buy - number of buy **hold** - number of hold
sell - number of sell
ssell - number of strong sell
usbuy - unique number of strong buy
ubuy - unique number of buy
uhold - unique number of hold
usell - unique number of sell
ussell - unique number of strong sell

start = YYYYMMDD

Specifies the date for the start of data. Required. If value is less than 10,000 the start date is this many days before the current date. The minimum date allowed is 20120601.

end = YYYYMMDD

Specifies the date for the end of data; inclusive. Optional. If not specified the most current calender day is be used.

Response

sentiment = [{object}, {object}, ...]

The array is sorted by symbol and then by date both in ascending order.

symbol - Symbol of the corresponding data

date - Date of the corresponding data

field - value

Each of the fields and values requested in the "**data**" array of the request.

Alert APIs

This API is intended for any customer that wants to build an automated trading system using the alerts generated by the HedgeChatter proprietary sentiment analysis based trading algorithms.

Get Alerts

Request

action = "getAlerts"

symbols - [symbol1, symbol2, ...]

If not provided then data is return for all symbols.

data = [value1, value2, ...]

An array of values specifying what data to return. If not specified then all fields are returned. Allowed values are:
tsa – timestamp of when the alert triggered
timestamp – timestamp of the price data
type – type of alert
direction - expected direction of price movement
price – adjusted close price of the stock on day of the alert
liq – liquidity on the date of the alert
mcap – market cap on the day of the alert

start = YYYYMMDD

Specifies the date for the start of data. Required if "**timestamp**" is not provided. If value is less than 10,000 the start date is this many days before the current date. The minimum date allowed is 20121001.

end = YYYYMMDD

Specifies the date for the end of data; inclusive. Optional. If not specified the most current calender day is be used.

timestamp = Unix timestamp in UCT

Specifies the earliest timestamp for the returned alerts. Optional.

type = [type1, type2, ...]

An array of values specifying what alert types to return. Optional. If not specified than all alert types are returned. Allowed values are: **chatter volume, sentiment volume, positive sentiment, negative sentiment, positive sentiment reputation.**

Response alerts = [{object}, {object}, ...]

The array is sorted by symbol and then by date both in ascending order.

symbol - Symbol of the corresponding data

date - Date of the corresponding data

field - value

Each of the fields and values requested in the "**data**" array of the request.

Authentication API

These APIs are used to obtain or delete a token which must be used in subsequent API requests. Also an API is provided to change the client secret.

The base URL for this APIs is:
Please contact info@HedgeChatter.com for the API URL

All responses have an "**error**" field that is set to 0 or 1. A "**message**" field that is set to "success" or the error message. If the response returns additional data it is in a "**result**" field that contains an object with the requested data. The API response details below show only the fields in the "result" and not the common fields that are in all responses.

Get Token

If the credentials are valid, an access token is returned which can be used for subsequent API requests. If there is an existing token that has not expired then it is returned instead of creating a new one.

Request
- action = "getToken"
- grant_type = "client_credentials"
- client_id - the id assigned to the client
- client_secret - the secret assigned to the client

Response
- access_token - the token to be used in future requests
- expires_in - number of seconds after which the token will expire if not used
- token_type = "bearer"
- scope - null; not used

Delete Token

If the credentials are valid, all tokens for the client are deleted. This method should be used to end a session. Although it is not required and any issued token will expire in time.

Request
- action = "deleteToken"
- client_id - the id assigned to the client
- client_secret - the secret assigned to the client

Response
- no result returned; check the 'error' field for success

Update Secret

This API is used to change the client secret. If the credentials are valid the value specified by 'new_secret' is accepted and become the 'client_secret' for subsequent requests.

Request

action = “updateSecret”
client_id - the id assigned to the client
client_secret - the secret assigned to the client
new_secret - the new secret value

Response

no result returned; check the 'error' field for success

HedgeChatter - Contact

For additional information regarding our platforms or API’s, please contact us at:

- 1-678-744-9720
- or by visiting www.HedgeChatter.com
- or by emailing us at info@hedgechatter.com