Trade Hybrid APis

Coinranking- https://developers.coinranking.com/api/documentation/stats/stats

https://api.coinranking.com/v2/stats

X-RapidAPI-Host - coinranking1.p.rapidapi.com
App

default-application_6766308

X-RapidAPI-Key

39b9c246b0msh8981e7993ba7354p1804d6jsn4711338b7ff9

Request URL

rapidapi.com

```
const http = require('https');

const options = {
        method: 'GET',
        hostname: 'coinranking1.p.rapidapi.com',
        port: null,
        path: '/stats?referenceCurrencyUuid=yhjMzLPhuIDl',
        headers: {
                'x-rapidapi-key': '39b9c246b0msh8981e7993ba7354p1804d6jsn4711338b7ff9',
                'x-rapidapi-host': 'coinranking1.p.rapidapi.com'
        }
};

const req = http.request(options, function (res) {
        const chunks = [];

        res.on('data', function (chunk) {
                chunks.push(chunk);
        });

        res.on('end', function () {
                const body = Buffer.concat(chunks);
                console.log(body.toString());
```

```
        });
});

req.end();
```

Investing-Cryptocurrency api

App

default-application_6766308

X-RapidAPI-Key

39b9c246b0msh8981e7993ba7354p1804d6jsn4711338b7ff9

rapidapi.com

Query Params

time_utc_offset

(optional)

Number

UTC value in seconds, for example : utc+8 -> 8 * 60 * 60 = 28800

lang_ID

(optional)

Number

The value of all_langs/lang_ID returned in .../get-meta-data endpoint

X-RapidAPI-Host

investing-cryptocurrency-markets.p.rapidapi.com

```
const http = require('https');

const options = {
        method: 'GET',
        hostname: 'investing-cryptocurrency-markets.p.rapidapi.com',
        port: null,
        path: '/coins/list-pairs?time_utc_offset=28800&lang_ID=1',
        headers: {
                'x-rapidapi-key': '39b9c246b0msh8981e7993ba7354p1804d6jsn4711338b7ff9',
                'x-rapidapi-host': 'investing-cryptocurrency-markets.p.rapidapi.com'
        }
};
```

```javascript
const req = http.request(options, function (res) {
        const chunks = [];

        res.on('data', function (chunk) {
                chunks.push(chunk);
        });

        res.on('end', function () {
                const body = Buffer.concat(chunks);
                console.log(body.toString());
        });
});

req.end();
```

Binance Api

App

default-application_6766308

X-RapidAPI-Key

39b9c246b0msh8981e7993ba7354p1804d6jsn4711338b7ff9

Request URL

rapidapi.com

```javascript
const http = require('https');

const options = {
        method: 'GET',
        hostname: 'binance43.p.rapidapi.com',
        port: null,
        path: '/ticker/24hr',
        headers: {
                'x-rapidapi-key': '39b9c246b0msh8981e7993ba7354p1804d6jsn4711338b7ff9',
                'x-rapidapi-host': 'binance43.p.rapidapi.com'
        }
};

const req = http.request(options, function (res) {
```

```
        const chunks = [];

        res.on('data', function (chunk) {
                chunks.push(chunk);
        });

        res.on('end', function () {
                const body = Buffer.concat(chunks);
                console.log(body.toString());
        });
});

req.end();
```

CNBC api
Query Params

symbol

*

String

Symbol of stock, index, exchange, etc...

```
const http = require('https');

const options = {
        method: 'GET',
        hostname: 'cnbc.p.rapidapi.com',
        port: null,
        path: '/symbols/translate?symbol=TSLA',
        headers: {
                'x-rapidapi-key': '39b9c246b0msh8981e7993ba7354p1804d6jsn4711338b7ff9',
                'x-rapidapi-host': 'cnbc.p.rapidapi.com'
        }
};

const req = http.request(options, function (res) {
```

```javascript
        const chunks = [];

        res.on('data', function (chunk) {
                chunks.push(chunk);
        });

        res.on('end', function () {
                const body = Buffer.concat(chunks);
                console.log(body.toString());
        });
});

req.end();
```

Fidelity Investments api

```javascript
const http = require('https');

const options = {
        method: 'GET',
        hostname: 'fidelity-investments.p.rapidapi.com',
        port: null,
        path: '/auto-complete?query=apple',
        headers: {
                'x-rapidapi-key': '39b9c246b0msh8981e7993ba7354p1804d6jsn4711338b7ff9',
                'x-rapidapi-host': 'fidelity-investments.p.rapidapi.com'
        }
};

const req = http.request(options, function (res) {
        const chunks = [];

        res.on('data', function (chunk) {
                chunks.push(chunk);
        });

        res.on('end', function () {
```

```
                const body = Buffer.concat(chunks);
                console.log(body.toString());
        });
});


req.end();
```

Real time data news api

Query Params

query
*
String
Search query for which to get news.

limit
(optional)
Number
Maximum number of news articles to return.

Default: 500

Allowed values: 1-500

time_published
(optional)
anytime
Enum
Find news articles published in a specific time range.

Default: anytime

source
(optional)
String
Domain of the source from which to return news articles.

Example: cnn.com

country
(optional)

String

Country code. See all available country codes.

Default: `US`

lang

(optional)

String

The language to use for the results, specified as a 2-letter language code - see ISO 639-1 alpha-2.

Default: `en`

```javascript
const http = require('https');

const options = {
	method: 'GET',
	hostname: 'real-time-news-data.p.rapidapi.com',
	port: null,
	path: '/search?query=Elon%20Musk&limit=500&time_published=anytime&country=US&lang=en',
	headers: {
		'x-rapidapi-key': '39b9c246b0msh8981e7993ba7354p1804d6jsn4711338b7ff9',
		'x-rapidapi-host': 'real-time-news-data.p.rapidapi.com'
	}
};

const req = http.request(options, function (res) {
	const chunks = [];

	res.on('data', function (chunk) {
		chunks.push(chunk);
	});

	res.on('end', function () {
		const body = Buffer.concat(chunks);
		console.log(body.toString());
	});
});

req.end();
```

Quotient api

Query Params

symbol
*
String
The symbol of the asset to look for, e.g., `AAPL` (Apple Inc), `MSFT` (Microsoft).

type
*
String
The type of option contract. (`Call` or `Put`).

expiration
(optional)
String
The expiration date of the option (supported format is : YYYY-mm-dd), e.g., `2021-12-14`.

strike
(optional)
Number
The strike price of the option, e.g., `122.00`.

min_expiry
(optional)
String
Alternatively, a lower bound for the expiration date (supported format is : YYYY-mm-dd), e.g.,
`2021-05-21`. If given, expiry range [`min_expiry`, `max_expiry`] takes priority over `expiration`
parameter.

max_expiry
(optional)
String
Alternatively, an upper bound for the expiration date (supported format is : YYYY-mm-dd), e.g.,
`2021-12-14`. If given, expiry range [`min_expiry`, `max_expiry`] takes priority over `expiration`
parameter.

min_strike
(optional)
Number
Alternatively, a lower bound for strike price, e.g., `50`. If given, strike range [`min_strike`,
`max_strike`] takes priority over `strike` parameter.

max_strike
(optional)
Number

Alternatively, an upper bound for strike price, e.g., 90. If given, strike range [min_strike, max_strike] takes priority over strike parameter.

```javascript
const http = require('https');

const options = {
	method: 'GET',
	hostname: 'quotient.p.rapidapi.com',
	port: null,
	path: '/options/prices?symbol=AAPL&type=Call&expiration=2024-12-14&strike=300&min_expiry=2024-05-21&max_expiry=2025-12-14&min_strike=50&max_strike=200',
	headers: {
		'x-rapidapi-key': '39b9c246b0msh8981e7993ba7354p1804d6jsn4711338b7ff9',
		'x-rapidapi-host': 'quotient.p.rapidapi.com'
	}
};

const req = http.request(options, function (res) {
	const chunks = [];

	res.on('data', function (chunk) {
		chunks.push(chunk);
	});

	res.on('end', function () {
		const body = Buffer.concat(chunks);
		console.log(body.toString());
	});
});

req.end();
```

Alpha vantage api
Query Params

function

*

String

The time series of your choice.

symbol

*

String

The name of the equity of your choice.

outputsize

(optional)

String

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

datatype

(optional)

String

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

```javascript
const http = require('https');

const options = {
	method: 'GET',
	hostname: 'alpha-vantage.p.rapidapi.com',
	port: null,
	path:
'/query?function=TIME_SERIES_DAILY&symbol=MSFT&outputsize=compact&datatype=json',
	headers: {
		'x-rapidapi-key': '39b9c246b0msh8981e7993ba7354p1804d6jsn4711338b7ff9',
		'x-rapidapi-host': 'alpha-vantage.p.rapidapi.com'
	}
};

const req = http.request(options, function (res) {
	const chunks = [];

	res.on('data', function (chunk) {
		chunks.push(chunk);
	});
```

```
        res.on('end', function () {
                const body = Buffer.concat(chunks);
                console.log(body.toString());
        });
});

req.end();
```

BB Finance api

Query Params

id
*

String

The id of stickers, use market/auto-complete API to get correct values

template

(optional)

String

One of the following COMMODITY|CURRENCY|INDEX|INDEXFUTURE|RATE|STOCK

```
const http = require('https');

const options = {
        method: 'GET',
        hostname: 'bb-finance.p.rapidapi.com',
        port: null,
        path: '/stock/get-statistics?id=aapl%3Aus&template=STOCK',
        headers: {
                'x-rapidapi-key': '39b9c246b0msh8981e7993ba7354p1804d6jsn4711338b7ff9',
                'x-rapidapi-host': 'bb-finance.p.rapidapi.com'
        }
};

const req = http.request(options, function (res) {
        const chunks = [];

        res.on('data', function (chunk) {
                chunks.push(chunk);
```

```
        });

        res.on('end', function () {
                const body = Buffer.concat(chunks);
                console.log(body.toString());
        });
});

req.end();
```

Trading view api

Query Params

exchange

*

String

One of the following :

US|ADX|AMEX|ATHEX|EURONEXT_BRU|ASX|BAHRAIN|BCBA|BCS|BELEX|BET|BER|BIST|BME|
BMFBOVESPA|BMV|BSE|BSSE|BVC|BVL|BVB|BVCV|BX|CSE|DFM|DUS|OMXCOP|OMXTSE|OM
XHEX|EGX|EURONEXT|EURONEXT_PAR|EURONEXT_AMS|EURONEXT_LIS|FWB|GPW|HAN|H
KEX|HNX|HOSE|IDX|JSE|LSE|LSIN|MIL|MOEX|MYX|MUN|NAG|NASDAQ|NEO|NEWCONNECT|N
GM|NSE|NSENG|NYSE|NZX|KRX|OTC|OMXICE|OMXRSE|OMXSTO|OMXVSE|LUXSE|OSL|PSE|
QSE|SGX|SIX|SWB|SZSE|SSE|SET|TADAWUL|TASE|TPEX|TSE|TSX|TSXV|TWSE|UPCOM|XETR

name

(optional)

String

One of the following :

volume_gainers|percent_change_gainers|percent_change_losers|percent_range_gainers|percent_ra
nge_losers|gap_gainers|gap_losers|percent_gap_gainers|percent_gap_losers

locale

(optional)

String

The language code

```
const http = require('https');

const options = {
        method: 'GET',
```

```
                hostname: 'trading-view.p.rapidapi.com',
                port: null,
                path: '/market/get-movers?exchange=US&name=volume_gainers&locale=en',
                headers: {
                        'x-rapidapi-key': '39b9c246b0msh8981e7993ba7354p1804d6jsn4711338b7ff9',
                        'x-rapidapi-host': 'trading-view.p.rapidapi.com'
                }
};

const req = http.request(options, function (res) {
        const chunks = [];

        res.on('data', function (chunk) {
                chunks.push(chunk);
        });

        res.on('end', function () {
                const body = Buffer.concat(chunks);
                console.log(body.toString());
        });
});

req.end();
```

Twitter Api

Query Params

listId
*
String
cursor
(optional)

```
const http = require('https');

const options = {
        method: 'GET',
        hostname: 'twitter-api47.p.rapidapi.com',
        port: null,
```

```javascript
        path: '/v2/latest-list-tweets',
        headers: {
                'x-rapidapi-key': '39b9c246b0msh8981e7993ba7354p1804d6jsn4711338b7ff9',
                'x-rapidapi-host': 'twitter-api47.p.rapidapi.com'
        }
};

const req = http.request(options, function (res) {
        const chunks = [];

        res.on('data', function (chunk) {
                chunks.push(chunk);
        });

        res.on('end', function () {
                const body = Buffer.concat(chunks);
                console.log(body.toString());
        });
});

req.end();
```