

Package: eodhd (via r-universe)

September 5, 2024

Type Package

Title Official 'eodhd' API R Financial Library

Version 1.0.4

Description Official 'eodhd' API R Library. It helps to get and work with financial data, historical data and etc. API reference available at <<https://eodhd.com/financial-apis/>>.

License MIT + file LICENSE

Encoding UTF-8

URL <https://github.com/EodHistoricalData/EODHD-APIs-R-Financial-Library>

BugReports

<https://github.com/EodHistoricalData/EODHD-APIs-R-Financial-Library/issues>

Suggests testthat

Imports httr

RoxygenNote 7.2.3

NeedsCompilation no

Author EODHD Maintainer [cph, aut, cre]

Maintainer EODHD Maintainer <support@eodhistoricaldata.com>

Date/Publication 2024-02-05 12:50:09 UTC

Additional_repositories <https://cranhaven.r-universe.dev>

Repository <https://cranhaven.r-universe.dev>

RemoteUrl <https://github.com/cranhaven/cranhaven.r-universe.dev>

RemoteRef package/eodhd

RemoteSha a4dcdad89f7a907c9a26ad70a83301b6838753ba

Contents

| | |
|---|-----------|
| financial_news | 2 |
| get_bonds_fundamentals_data | 3 |
| get_bulk_eod_splits_dividends_data | 4 |
| get_details_trading_hours_stock_market_holidays | 5 |
| get_earning_trends_data | 6 |
| get_economic_events_data | 6 |
| get_eod_historical_stock_market_data | 7 |
| get_fundamentals_data | 8 |
| get_historical_dividends_data | 9 |
| get_historical_market_capitalization_data | 10 |
| get_historical_splits_data | 10 |
| get_insider_transactions_data | 11 |
| get_intraday_historical_data | 12 |
| get_list_of_exchanges | 13 |
| get_live_stock_prices | 14 |
| get_macro_indicators_data | 14 |
| get_options_data | 15 |
| get_sentiment_data | 16 |
| get_stock_market_tick_data | 17 |
| get_technical_indicator_data | 18 |
| get_upcoming_earnings_data | 19 |
| get_upcoming_IPOs_data | 20 |
| get_upcoming_splits_data | 21 |
| rest_get_method | 21 |
| stock_market_screener | 22 |
| symbol_change_history | 23 |
| Index | 24 |

| | |
|----------------|-----------------------|
| financial_news | <i>Financial news</i> |
|----------------|-----------------------|

Description

This function retrieves financial news data from the API.

Usage

```
financial_news(
  api_token,
  s = NULL,
  t = NULL,
  from_date = NULL,
  to_date = NULL,
  limit = NULL,
  offset = NULL
)
```

Arguments

| | |
|-----------|---|
| api_token | The API token for authentication. |
| s | (required if t empty) - The ticker code to get news for. |
| t | (required if s empty) - The tag to get news on a given topic. |
| from_date | date from with format Y-m-d |
| to_date | date from with format Y-m-d |
| limit | The number of results should be returned with the query |
| offset | The offset of the data. |

Value

A list containing the financial news.

Examples

```
api_token <- "demo"
s <- "balance sheet"
t <- NULL
from_date <- "2017-09-10"
to_date <- "2017-09-12"
limit <- "50"
offset <- "100"
result <- financial_news(api_token, s, t, from_date, to_date, limit, offset)
```

get_bonds_fundamentals_data

Get bonds fundamentals data

Description

This function retrieves bond fundamentals data from the API.

Usage

```
get_bonds_fundamentals_data(api_token, isin)
```

Arguments

| | |
|-----------|--|
| api_token | The API token for authentication. |
| isin | The ISIN (International Securities Identification Number) of the bond. |

Value

A list containing the bond fundamentals data.

Examples

```
api_token <- "demo"  
isin <- "DE000CB83CF0"  
result <- get_bonds_fundamentals_data(api_token, isin)
```

```
get_bulk_eod_splits_dividends_data  
  Get bulk fundamentals data
```

Description

This function retrieves bulk fundamentals data from the API.

Usage

```
get_bulk_eod_splits_dividends_data(  
  api_token,  
  country = "US",  
  type = NULL,  
  date = NULL,  
  symbols = NULL,  
  filter = NULL  
)
```

Arguments

| | |
|-----------|--|
| api_token | The API token for authentication. |
| country | <- 'US' |
| type | <- can be empty, splits or dividends |
| date | By default, the data for last trading day will be downloaded, but if you need any specific date, add 'date' parameter to the URL |
| symbols | To download last day data for several symbols, for example, for MSFT and AAPL, you can add the 'symbols' parameter |
| filter | If you need more data, like company name, you can use '&filter=extended' |

Value

A list containing bulk fundamentals data.

Examples

```
api_token <- "demo"
country <- 'US'
type <- 'splits'
date <- "2010-09-21"
symbols <- "MSFT"
filter <- "extended"
result <- get_bulk_eod_splits_dividends_data(api_token, country, type, date, symbols, filter)
```

get_details_trading_hours_stock_market_holidays

Get details trading hours, stock market holidays and symbol change history

Description

This function retrieves details trading hours, stock market holidays and symbol change history from the API.

Usage

```
get_details_trading_hours_stock_market_holidays(
  api_token,
  code,
  from_date = NULL,
  to_date = NULL
)
```

Arguments

| | |
|-----------|---|
| api_token | The API token for authentication. |
| code | Use the exchange code from the API endpoint |
| from_date | the format is 'YYYY-MM-DD' |
| to_date | the format is 'YYYY-MM-DD' |

Value

A list containing the details trading hours, stock market holidays and symbol change history.

Examples

```
api_token <- "demo"
code <- "US"
from_date <- "2017-09-10"
to_date <- "2017-09-12"

result <- get_details_trading_hours_stock_market_holidays(api_token, code, from_date, to_date)
```

```
get_earning_trends_data
```

Get earning trends data

Description

This function retrieves earning trends data from the API.

Usage

```
get_earning_trends_data(api_token, symbols)
```

Arguments

| | |
|-----------|--|
| api_token | The API token for authentication. |
| symbols | You can request specific symbols to get historical and upcoming data |

Value

A list containing earning trends.

Examples

```
api_token <- "demo"  
symbols <- "AAPL.US"  
result <- get_earning_trends_data(api_token, symbols)
```

```
get_economic_events_data
```

Get economic events data

Description

This function retrieves economic events data from the API.

Usage

```
get_economic_events_data(  
  api_token,  
  date_from = NULL,  
  date_to = NULL,  
  country = NULL,  
  comparison = NULL,  
  offset = NULL,  
  limit = NULL  
)
```

Arguments

| | |
|------------|---|
| api_token | The API token for authentication. |
| date_from | date from with format Y-m-d |
| date_to | date from with format Y-m-d |
| country | The country code is in ISO 3166 format, has 2 symbols |
| comparison | Possible values: mom, qoq, yoy |
| offset | Possible values from 0 to 1000 |
| limit | Possible values from 0 to 1000 |

Value

A list containing economic data events.

Examples

```
api_token <- "demo"
date_from <- "2017-09-10"
date_to <- "2017-09-12"
country <- "US"
comparison <- "qoq"
offset <- "0"
limit <- "50"
result <- get_economic_events_data(api_token, date_from, date_to, country,
comparison, offset, limit)
```

```
get_eod_historical_stock_market_data
  Get historical data
```

Description

This function retrieves historical data from the API.

Usage

```
get_eod_historical_stock_market_data(
  api_token,
  symbol,
  from_date = NULL,
  to_date = NULL,
  period = "d",
  order = "a"
)
```

Arguments

| | |
|-----------|--|
| api_token | The API token for authentication. |
| symbol | consists of two parts: SYMBOL_NAME.EXCHANGE_ID, then you can use, for example, AAPL.MX for Mexican Stock Exchange. or AAPL.US for NASDAQ |
| from_date | Format: YYYY-MM-DD. The start date for earnings data, if not provided, today will be used. |
| to_date | Format: YYYY-MM-DD. The end date for earnings data, if not provided, today + 7 days will be used. |
| period | use 'd' for daily, 'w' for weekly, 'm' for monthly prices. By default, daily prices will be shown. |
| order | use 'a' for ascending dates (from old to new), 'd' for descending dates (from new to old). |

Value

A list containing the historical data.

Examples

```
api_token <- "demo"
from_date <- "2017-09-10"
to_date <- "2017-09-12"
symbol <- "AAPL.MX"
period <- "d"
order <- "a"
result <- get_eod_historical_stock_market_data(api_token, symbol, from_date, to_date, period, order)
```

get_fundamentals_data *Get fundamentals data data*

Description

This function retrieves fundamentals data from the API.

Usage

```
get_fundamentals_data(api_token, ticker)
```

Arguments

| | |
|-----------|--|
| api_token | The API token for authentication. |
| ticker | consists of two parts: [SYMBOL_NAME].[EXCHANGE_ID] |

Value

A list containing the fundamental data.

Examples

```
api_token <- "demo"  
ticker <- "AAPL.US"  
result <- get_fundamentals_data(api_token, ticker)
```

get_historical_dividends_data
Get historical dividends data

Description

This function retrieves historical dividends data from the API.

Usage

```
get_historical_dividends_data(  
  api_token,  
  ticker,  
  date_from = NULL,  
  date_to = NULL  
)
```

Arguments

- `api_token` The API token for authentication.
- `ticker` consists of two parts: [SYMBOL_NAME].[EXCHANGE_ID].
- `date_from` date from with format Y-m-d
- `date_to` date from with format Y-m-d

Value

A list containing historical dividends data.

Examples

```
api_token <- "demo"  
ticker <- "AAPL.US"  
date_from <- "2017-09-10"  
date_to <- "2017-09-12"  
result <- get_historical_dividends_data(api_token, ticker, date_from, date_to)
```

```
get_historical_market_capitalization_data
```

Get stock market tick data

Description

This function retrieves stock market tick data from the API.

Usage

```
get_historical_market_capitalization_data(  
  api_token,  
  ticker,  
  from_date = NULL,  
  to_date = NULL  
)
```

Arguments

| | |
|-----------|--|
| api_token | The API token for authentication. |
| ticker | is the ticker code and it consists of two parts: SYMBOL_NAME.EXCHANGE_ID |
| from_date | Format: YYYY-MM-DD. |
| to_date | Format: YYYY-MM-DD. |

Value

A list containing the stock market tick data.

Examples

```
api_token <- "demo"  
from_date <- "2017-09-10"  
to_date <- "2017-09-12"  
ticker <- "AAPL"  
result <- get_historical_market_capitalization_data(api_token, ticker, from_date, to_date)
```

```
get_historical_splits_data
```

Get historical splits data

Description

This function retrieves historical splits data from the API.

Usage

```
get_historical_splits_data(api_token, ticker, date_from = NULL, date_to = NULL)
```

Arguments

| | |
|-----------|---|
| api_token | The API token for authentication. |
| ticker | consists of two parts: [SYMBOL_NAME],[EXCHANGE_ID]. |
| date_from | date from with format Y-m-d |
| date_to | date from with format Y-m-d |

Value

A list containing historical splits data.

Examples

```
api_token <- "demo"
ticker <- "AAPL.US"
date_from <- "2017-09-10"
date_to <- "2017-09-12"
result <- get_historical_splits_data(api_token, ticker, date_from, date_to)
```

```
get_insider_transactions_data
```

Get insider transactions data

Description

This function retrieves the insider transactions from the API.

Usage

```
get_insider_transactions_data(  
  api_token,  
  date_from = NULL,  
  date_to = NULL,  
  code = NULL,  
  limit = NULL  
)
```

Arguments

| | |
|-----------|---|
| api_token | The API token for authentication. |
| date_from | date from with format Y-m-d |
| date_to | date from with format Y-m-d |
| code | to get the data only for Apple Inc (AAPL), use AAPL.US or AAPL ticker code. |
| limit | the limit for entries per result, from 1 to 1000. |

Value

A list containing the insider transactions data.

Examples

```
api_token <- "demo"
date_from <- "2017-09-10"
date_to <- "2017-09-12"
code <- "AAPL.US"
limit <- "150"
result <- get_insider_transactions_data(api_token, date_from, date_to, code, limit)
```

```
get_intraday_historical_data
      Get historical intraday data
```

Description

This function retrieves historical intraday data from the API. IMPORTANT: data for all exchanges is provided in the UTC timezone, with Unix timestamps.

Usage

```
get_intraday_historical_data(
  api_token,
  symbol,
  from_unix_time = NULL,
  to_unix_time = NULL,
  interval = "5m"
)
```

Arguments

| | |
|----------------|---|
| api_token | The API token for authentication. |
| symbol | consists of two parts: SYMBOL_NAME.EXCHANGE_ID, then you can use, for example, AAPL.MX for Mexican Stock Exchange. or AAPL.US for NASDAQ |
| from_unix_time | Parameters should be passed in UNIX time with UTC timezone, for example, these values are correct: "from=1627896900&to=1630575300" and correspond to ' 2021-08-02 09:35:00 ' and ' 2021-09-02 09:35:00 '. |
| to_unix_time | Parameters should be passed in UNIX time with UTC timezone. |
| interval | the possible intervals: '5m' for 5-minutes, '1h' for 1 hour, and '1m' for 1-minute intervals. |

Value

A list containing the historical data.

Examples

```
api_token <- "demo"  
from_unix_time <- "1627896900"  
to_unix_time <- "1630575300"  
symbol <- "AAPL.MX"  
interval <- "5m"  
result <- get_intraday_historical_data(api_token, symbol, from_unix_time, to_unix_time, interval)
```

get_list_of_exchanges *Get list of exchanges*

Description

This function retrieves list of exchanges from the API.

Usage

```
get_list_of_exchanges(api_token)
```

Arguments

`api_token` The API token for authentication.

Value

A list containing the list of exchanges.

Examples

```
api_token <- "demo"  
result <- get_list_of_exchanges(api_token)
```

get_live_stock_prices *Get live stock prices*

Description

This function retrieves live stock prices from the API.

Usage

```
get_live_stock_prices(api_token, ticker, s = NULL)
```

Arguments

| | |
|-----------|---|
| api_token | The API token for authentication. |
| ticker | consists of two parts: [SYMBOL_NAME].[EXCHANGE_ID] |
| s | add "s=" parameter to your function and you will be able to get data for multiple tickers at one request, all tickers should be separated with a comma. |

Value

A list containing a live stock prices.

Examples

```
api_token <- "demo"  
ticker <- "AAPL.US"  
s <- "VTI, EUR.FOREX"  
result <- get_live_stock_prices(api_token, ticker, s)
```

get_macro_indicators_data
Get macro indicators data

Description

This function retrieves macro indicators data from the API.

Usage

```
get_macro_indicators_data(api_token, country, indicator = NULL)
```

Arguments

| | |
|-----------|--|
| api_token | The API token for authentication. |
| country | Defines the country for which the indicator will be shown. The country should be defined in the Alpha-3 ISO format. |
| indicator | Defines which macroeconomics data indicator will be shown. All possible indicators will be available on: https://eodhistoricaldata.com/financial-apis/macroeconomics-data-and-macro-indicators-api/ |

Value

A list containing the macro indicators data.

Examples

```
api_token <- "demo"
country <- "USA"
indicator <- "gdp_current_usd"
result <- get_macro_indicators_data(api_token, country, indicator)
```

get_options_data *Get options data*

Description

This function retrieves options data from the API.

Usage

```
get_options_data(
  api_token,
  symbol,
  date_to = NULL,
  date_from = NULL,
  trade_date_to = NULL,
  trade_date_from = NULL,
  contract_name = NULL
)
```

Arguments

| | |
|---------------|---|
| api_token | The API token for authentication. |
| symbol | Required - Could be any supported symbol. No default value. |
| date_to | date from with format Y-m-d |
| date_from | date from with format Y-m-d |
| trade_date_to | date from with format Y-m-d |

trade_date_from date from with format Y-m-d
contract_name Name of a particular contract.

Value

A list containing options data.

Examples

```
api_token <- "demo"  
symbol <- "AAPL.US"  
date_to <- "2017-09-10"  
date_from <- "2017-09-12"  
trade_date_to <- "2010-05-15"  
trade_date_from <- "2010-05-16"  
contract_name <- "AAPL230818C00055000"  
result <- get_options_data(api_token, date_to, date_from, trade_date_to,  
trade_date_from, contract_name)
```

get_sentiment_data *Get sentiment data.*

Description

This function retrieves sentiment data from the API.

Usage

```
get_sentiment_data(api_token, s, from_date = NULL, to_date = NULL)
```

Arguments

| | |
|-----------|---|
| api_token | The API token for authentication. |
| s | Parameter to your URL and you will be able to get data for multiple tickers at one request, all tickers should be separated with a comma. |
| from_date | Format: YYYY-MM-DD. |
| to_date | Format: YYYY-MM-DD. |

Value

A list containing the sentiment data.

Examples

```
api_token <- "demo"
from_date <- "2017-09-10"
to_date <- "2017-09-12"
s <- "AAPL.MX"
result <- get_sentiment_data(api_token, s, from_date, to_date)
```

get_stock_market_tick_data

Get stock market tick data

Description

This function retrieves stock market tick data from the API.

Usage

```
get_stock_market_tick_data(
  api_token,
  symbol,
  from_timestamp,
  to_timestamp,
  limit = 1
)
```

Arguments

| | |
|----------------|--|
| api_token | The API token for authentication. |
| symbol | consists of two parts: SYMBOL_NAME.EXCHANGE_ID. This API works only for US exchanges for the moment, then you can use 'AAPL' or 'AAPL.US' to get the data as well for other US tickers. |
| from_timestamp | use these parameters to filter data by datetime. Parameters should be passed in UNIX time with UTC timezone, for example, these values are correct: "from=1627896900&to=16305753" and correspond to ' 2021-08-02 09:35:00 ' and ' 2021-09-02 09:35:00 '. |
| to_timestamp | use these parameters to filter data by datetime. |
| limit | use 'a' for ascending dates (from old to new), 'd' for descending dates (from new to old). |

Value

A list containing the stock market tick data.

Examples

```

api_token <- "demo"
from_timestamp <- "1694354400"
to_timestamp <- "1694455200"
symbol <- "AAPL"
limit <- 1
result <- get_stock_market_tick_data(api_token, symbol, from_timestamp, to_timestamp, limit)

```

```

get_technical_indicator_data
    Get technical indicator data

```

Description

This function retrieves technical indicator data from the API.

Usage

```

get_technical_indicator_data(
  api_token,
  ticker,
  func,
  period = 50,
  date_from = NULL,
  date_to = NULL,
  order = "a",
  splitadjusted_only = "0"
)

```

Arguments

| | |
|--------------------|---|
| api_token | The API token for authentication. |
| ticker | consists of two parts: [SYMBOL_NAME].[EXCHANGE_ID] |
| func | the function that will be applied to data series to get technical indicator data. |
| period | the number of data points used to calculate each moving average value. |
| date_from | date from with format Y-m-d. |
| date_to | date from with format Y-m-d. |
| order | use 'a' for ascending dates (from old to new) and 'd' for descending dates (from new to old). By default, dates are shown in ascending order. |
| splitadjusted_only | By default, we calculate data for some functions by closes adjusted with splits and dividends. If you need to calculate the data by closes adjusted only with splits, set this parameter to '1'. Works with the following functions: sma, ema, wma, volatility, rsi, slope, and macd. |

Value

A list containing the technical indicator data.

Examples

```
api_token <- "demo"
ticker <- "AAPL.US"
func <- "sma"
period <- "100"
date_from <- "2017-09-10"
date_to <- "2017-09-12"
order <- "d"
splitadjusted_only <- "1"
result <- get_technical_indicator_data(api_token, ticker, func, period,
date_from, date_to, order, splitadjusted_only)
```

```
get_upcoming_earnings_data
      Get upcoming earnings data
```

Description

This function retrieves upcoming earnings data from the API.

Usage

```
get_upcoming_earnings_data(
  api_token,
  from_date = NULL,
  to_date = NULL,
  symbols = NULL
)
```

Arguments

| | |
|-----------|---|
| api_token | The API token for authentication. |
| from_date | Format: YYYY-MM-DD. The start date for earnings data, if not provided, today will be used. |
| to_date | Format: YYYY-MM-DD. The end date for earnings data, if not provided, today + 7 days will be used. |
| symbols | You can request specific symbols to get historical and upcoming data. |

Value

A list containing the upcoming earnings data.

Examples

```
api_token <- "demo"  
from_date <- "2017-09-10"  
to_date <- "2017-09-12"  
symbols <- "MS"  
result <- get_upcoming_earnings_data(api_token, from_date, to_date, symbols)
```

`get_upcoming_IPOs_data`

Get upcoming IPOs data

Description

This function retrieves upcoming IPOs data from the API.

Usage

```
get_upcoming_IPOs_data(api_token, from_date = NULL, to_date = NULL)
```

Arguments

| | |
|------------------------|---|
| <code>api_token</code> | The API token for authentication. |
| <code>from_date</code> | Format: YYYY-MM-DD. The start date for IPOs data, if not provided, today will be used. |
| <code>to_date</code> | Format: YYYY-MM-DD. The end date for IPOs data, if not provided, today + 7 days will be used. |

Value

A list containing the upcoming IPOs data.

Examples

```
api_token <- "demo"  
from_date <- "2017-09-10"  
to_date <- "2017-09-12"  
result <- get_upcoming_IPOs_data(api_token, from_date, to_date)
```

get_upcoming_splits_data
Get upcoming splits data

Description

This function retrieves upcoming splits data from the API.

Usage

```
get_upcoming_splits_data(api_token, from_date = NULL, to_date = NULL)
```

Arguments

| | |
|-----------|---|
| api_token | The API token for authentication. |
| from_date | Format: YYYY-MM-DD. The start date for splits data, if not provided, today will be used. |
| to_date | Format: YYYY-MM-DD. The end date for splits data, if not provided, today + 7 days will be used. |

Value

A list containing the upcoming splits data.

Examples

```
api_token <- "demo"  
from_date <- "2017-09-10"  
to_date <- "2017-09-12"  
result <- get_upcoming_splits_data(api_token, from_date, to_date)
```

rest_get_method *Generic REST GET method*

Description

This function performs a generic REST GET request to the specified API endpoint. It takes an API key, endpoint, URI, and querystring as input and returns the response data in JSON format.

Usage

```
rest_get_method(api_key, endpoint = "", uri = "", querystring = list())
```

Arguments

| | |
|-------------|--|
| api_key | A character string representing the API key for authentication. |
| endpoint | A character string representing the API endpoint to request. |
| uri | A character string representing the URI for the request. |
| querystring | A list of key-value pairs representing the query parameters for the request. |

Value

The response data in JSON format.

Examples

```
api_key <- "demo"
endpoint <- "bond-fundamentals"
uri <- "DE000CB83CF0"
querystring <- list()
result <- rest_get_method(api_key, endpoint, uri, querystring)
print(result)
```

stock_market_screener *Stock market screener*

Description

This function retrieves stock market screener from the API.

Usage

```
stock_market_screener(  
  api_token,  
  sort = NULL,  
  filters = NULL,  
  limit = NULL,  
  signals = NULL,  
  offset = NULL  
)
```

Arguments

| | |
|-----------|--|
| api_token | The API token for authentication. |
| sort | Sorts all fields with type 'Number' in ascending/descending order. |
| filters | Filters out tickers by different fields. |
| limit | The number of results should be returned with the query. |
| signals | Filter out tickers by signals, the calculated fields. |
| offset | The offset of the data. |

Value

A list containing the stock market screener.

Examples

```
api_token <- "demo"
sort <- "market_capitalization.desc"
filters <- NULL
limit <- "1"
signals <- "bookvalue_neg"
offset <- "50"
result <- stock_market_screener(api_token, sort, filters, limit, signals, offset)
```

symbol_change_history *Get symbol change history*

Description

This function retrieves symbol change history from the API.

Usage

```
symbol_change_history(api_token, from_date = NULL, to_date = NULL)
```

Arguments

| | |
|-----------|--|
| api_token | The API token for authentication. |
| from_date | The start date for symbol change history (optional). |
| to_date | The end date for symbol change history (optional). |

Value

A list containing symbol change history data.

Examples

```
api_token <- "demo"
from_date <- "2023-01-01"
to_date <- "2023-12-31"
result <- symbol_change_history(api_token, from_date, to_date)
```

Index

[financial_news](#), [2](#)

[get_bonds_fundamentals_data](#), [3](#)
[get_bulk_eod_splits_dividends_data](#), [4](#)
[get_details_trading_hours_stock_market_holidays](#),
[5](#)
[get_earning_trends_data](#), [6](#)
[get_economic_events_data](#), [6](#)
[get_eod_historical_stock_market_data](#),
[7](#)
[get_fundamentals_data](#), [8](#)
[get_historical_dividends_data](#), [9](#)
[get_historical_market_capitalization_data](#),
[10](#)
[get_historical_splits_data](#), [10](#)
[get_insider_transactions_data](#), [11](#)
[get_intraday_historical_data](#), [12](#)
[get_list_of_exchanges](#), [13](#)
[get_live_stock_prices](#), [14](#)
[get_macro_indicators_data](#), [14](#)
[get_options_data](#), [15](#)
[get_sentiment_data](#), [16](#)
[get_stock_market_tick_data](#), [17](#)
[get_technical_indicator_data](#), [18](#)
[get_upcoming_earnings_data](#), [19](#)
[get_upcoming_IPOs_data](#), [20](#)
[get_upcoming_splits_data](#), [21](#)

[rest_get_method](#), [21](#)

[stock_market_screener](#), [22](#)
[symbol_change_history](#), [23](#)