

Package ‘eodhdR2’

September 3, 2024

Type Package

Title Official R API for Fetching Data from 'EODHD'

Version 0.5.0

Maintainer Marcelo S. Perlin <marceloperlin@gmail.com>

Description Second and backward-incompatible version of R package 'eodhd' <<https://eodhd.com/>>, extended with a cache and quota system, also offering functions for cleaning and aggregating the financial data.

License MIT + file LICENSE

URL <https://github.com/EodHistoricalData/R-Library-for-financial-data-2024>

BugReports

<https://github.com/EodHistoricalData/R-Library-for-financial-data-2024/issues>

Encoding UTF-8

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, cli, dplyr, fs, glue, httr, jsonlite, lubridate, purrr, readr, tidyr, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Marcelo S. Perlin [aut, cre, ctr],
Unicorn Data Services [cph]

Repository CRAN

Date/Publication 2024-09-03 15:10:02 UTC

Contents

| | |
|----------------------------|---|
| get_demo_token | 2 |
| get_dividends | 2 |
| get_exchanges | 3 |
| get_fundamentals | 4 |
| get_prices | 5 |
| get_splits | 6 |

| | |
|----------------------------|---|
| get_tickers | 7 |
| parse_financials | 8 |
| set_token | 9 |

| | |
|--------------|-----------|
| Index | 10 |
|--------------|-----------|

| | |
|----------------|--|
| get_demo_token | <i>Returns token for demonstration</i> |
|----------------|--|

Description

Returns token for demonstration

Usage

```
get_demo_token()
```

Value

A string with token

Examples

```
get_demo_token()
```

| | |
|---------------|---|
| get_dividends | <i>Retrieves dividend data from the api</i> |
|---------------|---|

Description

This function will query the dividend end point <https://eodhd.com/financial-apis/api-splits-dividends> and return:

- dates (declaration, record, payment)
- value of dividend (adjusted and unadjusted)
- currency of dividend

Usage

```
get_dividends(
    ticker = "AAPL",
    exchange = "US",
    cache_folder = get_default_cache(),
    check_quota = TRUE
)
```

Arguments

| | |
|--------------|---|
| ticker | A company ticker (e.g. AAPL). You can find all tickers for a particular exchange with <code>get_tickers()</code> . |
| exchange | A exchange symbol (e.g. US). You can find all tickers for a particular exchange with <code>get_tickers()</code> . Be aware that, for US companies, the exchange symbols is simply "US" |
| cache_folder | A local directory to store cache files. By default, all functions use a temporary path, meaning that the caching system is session persistent (it will remove all files when you exit your R session). If you want a persistent caching system, simply point argument <code>cache_folder</code> to a local directory in your filesystem. Be aware, however, that a persistent cache will not refresh your data for new api queries. |
| check_quota | A flag (TRUE/FALSE) for whether to check the current quota status from the api. This option implies a small cost of execution time. If you need speed, just set it to FALSE. |

Value

A dataframe with dividend information

Examples

```
## Not run:
set_token(get_demo_token())
df_div <- get_dividends(ticker = "AAPL", exchange = "US")
df_div

## End(Not run)
```

| | |
|---------------|--|
| get_exchanges | <i>Retrieves the list of available exchanges</i> |
|---------------|--|

Description

Retrieves the list of available exchanges

Usage

```
get_exchanges(cache_folder = get_default_cache())
```

Arguments

| | |
|--------------|---|
| cache_folder | A local directory to store cache files. By default, all functions use a temporary path, meaning that the caching system is session persistent (it will remove all files when you exit your R session). If you want a persistent caching system, simply point argument <code>cache_folder</code> to a local directory in your filesystem. Be aware, however, that a persistent cache will not refresh your data for new api queries. |
|--------------|---|

Value

a dataframe with information about available exchanges

Examples

```
# you need a valid token (not test) for this to work
## Not run:
set_token("YOUR_VALID_TOKEN")
df_exc <- get_exchanges()

## End(Not run)
```

| | |
|------------------|---|
| get_fundamentals | <i>Retrieves and parses fundamental and financial data from eodhd api</i> |
|------------------|---|

Description

This function will download raw data from the fundamental end point of eodhd <https://eodhd.com/financial-apis/stock-etfs-fundamental-data-feeds> and return a list. The raw data includes:

- General information for the company (code, ISIN, currency, ..)
- Financial highlights
- Valuation
- Raw financial data (see [parse_financials\(\)](#) for parsing this data)
- and many more (see example for more details regarding the output)

Usage

```
get_fundamentals(
  ticker = "AAPL",
  exchange = "US",
  cache_folder = get_default_cache(),
  check_quota = TRUE
)
```

Arguments

| | |
|----------|---|
| ticker | A company ticker (e.g. AAPL). You can find all tickers for a particular exchange with get_tickers() . |
| exchange | A exchange symbol (e.g. US). You can find all tickers for a particular exchange with get_tickers() . Be aware that, for US companies, the exchange symbols is simply "US" |

- cache_folder A local directory to store cache files. By default, all functions use a temporary path, meaning that the caching system is session persistent (it will remove all files when you exit your R session). If you want a persistent caching system, simply point argument cache_folder to a local directory in your filesystem. Be aware, however, that a persistent cache will not refresh your data for new api queries.
- check_quota A flag (TRUE/FALSE) for whether to check the current quota status from the api. This option implies a small cost of execution time. If you need speed, just set it to FALSE.

Value

a list with several fundamental information

Examples

```
## Not run:
set_token(get_demo_token())
l_out <- get_fundamentals(ticker = "AAPL", exchange = "US")
names(l_out)

## End(Not run)
```

| | |
|------------|---|
| get_prices | <i>Retrieves adjusted and unadjusted stock prices</i> |
|------------|---|

Description

This function will query the price end point of eodhd and return daily stock price from a set of ticker and exchange. It also includes the daily stock return (percentage variation).

Usage

```
get_prices(
  ticker = "AAPL",
  exchange = "US",
  cache_folder = get_default_cache(),
  check_quota = TRUE
)
```

Arguments

- ticker A company ticker (e.g. AAPL). You can find all tickers for a particular exchange with [get_tickers\(\)](#).
- exchange A exchange symbol (e.g. US). You can find all tickers for a particular exchange with [get_tickers\(\)](#). Be aware that, for US companies, the exchange symbols is simply "US"

| | |
|--------------|--|
| cache_folder | A local directory to store cache files. By default, all functions use a temporary path, meaning that the caching system is session persistent (it will remove all files when you exit your R session). If you want a persistent caching system, simply point argument cache_folder to a local directory in your filesystem. Be aware, however, that a persistent cache will not refresh your data for new api queries. |
| check_quota | A flag (TRUE/FALSE) for whether to check the current quota status from the api. This option implies a small cost of execution time. If you need speed, just set it to FALSE. |

Value

A dataframe with prices

Examples

```
## Not run:
set_token(get_demo_token())
df_prices <- get_prices(ticker = "AAPL", exchange = "US")

## End(Not run)
```

| | |
|------------|---|
| get_splits | <i>Retrieves splits data from eodhd</i> |
|------------|---|

Description

This function will query the splits end point of eodhd and return all split information for a given stock/exchange.

Usage

```
get_splits(
  ticker = "AAPL",
  exchange = "US",
  cache_folder = get_default_cache(),
  check_quota = TRUE
)
```

Arguments

| | |
|----------|---|
| ticker | A company ticker (e.g. AAPL). You can find all tickers for a particular exchange with get_tickers() . |
| exchange | A exchange symbol (e.g. US). You can find all tickers for a particular exchange with get_tickers() . Be aware that, for US companies, the exchange symbols is simply "US" |

| | |
|--------------|--|
| cache_folder | A local directory to store cache files. By default, all functions use a temporary path, meaning that the caching system is session persistent (it will remove all files when you exit your R session). If you want a persistent caching system, simply point argument cache_folder to a local directory in your filesystem. Be aware, however, that a persistent cache will not refresh your data for new api queries. |
| check_quota | A flag (TRUE/FALSE) for whether to check the current quota status from the api. This option implies a small cost of execution time. If you need speed, just set it to FALSE. |

Value

A dataframe with split information

Examples

```
## Not run:  
# requires a subscription (paid) token  
df_split <- get_splits(ticker = "AAPL", exchange = "US")  
  
## End(Not run)
```

| | |
|-------------|--|
| get_tickers | <i>Retrieves a list of tickers for a particular exchange</i> |
|-------------|--|

Description

Retrieves a list of tickers for a particular exchange

Usage

```
get_tickers(exchange = "US", cache_folder = get_default_cache())
```

Arguments

| | |
|--------------|--|
| exchange | A exchange symbol (e.g. US). You can find all tickers for a particular exchange with get_tickers() . Be aware that, for US companies, the exchange symbols is simply "US" |
| cache_folder | A local directory to store cache files. By default, all functions use a temporary path, meaning that the caching system is session persistent (it will remove all files when you exit your R session). If you want a persistent caching system, simply point argument cache_folder to a local directory in your filesystem. Be aware, however, that a persistent cache will not refresh your data for new api queries. |

Value

A dataframe with a list of tickers

Examples

```
## Not run: # requires a subscription (paid) token
df_tickers <- get_tickers("US")

## End(Not run)
```

| | |
|------------------|---|
| parse_financials | <i>Parses financial data from the API</i> |
|------------------|---|

Description

This function will organize the raw financial data from `get_fundamentals()`, aggregating all information into a single dataframe, including quarterly and yearly data from the Balance_sheet, Cash-flow statement and Income statement. Whenever no financial data is found in `l_out`, the function returns an empty dataframe.

Usage

```
parse_financials(l_out, type_table = "long")
```

Arguments

| | |
|-------------------------|--|
| <code>l_out</code> | A list with raw data (output from <code>get_fundamentals()</code>) |
| <code>type_table</code> | Format of table in output ("wide" or "long"). A "wide" table is a typical Excel column-oriented table where each column is a data/year. A long type of table is row-oriented, where each point of new information is a row of the table. The data is the same, it just changes the orientation of rows/columns. The default value is a "long" table. |

Value

A dataframe with organized financial data in the wide or long format

Examples

```
## Not run:
set_token(get_demo_token())
l_out <- get_fundamentals(ticker = "AAPL", exchange = "US")

df_fin <- parse_financials(l_out, "long")
df_fin

## End(Not run)
```

| | |
|-----------|---|
| set_token | <i>Sets up authentication for eodhd</i> |
|-----------|---|

Description

Uses the token from <https://eodhd.com/cp/dashboard> to authenticate your R session. You can find your own eodhd token from the website. Alternatively, a demo token is also available for testing purposes, with a limited supply of data.

Usage

```
set_token(token = get_demo_token())
```

Arguments

| | |
|-------|---|
| token | the token from eodhd. The default value is a demo token "demo", which allows for partial access to the data. See get_demo_token() for using a demo token. |
|-------|---|

Value

Nothing

Examples

```
## Not run:  
set_token()  
  
## End(Not run)
```

Index

`get_demo_token`, 2
`get_demo_token()`, 9
`get_dividends`, 2
`get_exchanges`, 3
`get_fundamentals`, 4
`get_fundamentals()`, 8
`get_prices`, 5
`get_splits`, 6
`get_tickers`, 7
`get_tickers()`, 3–7

`parse_financials`, 8
`parse_financials()`, 4

`set_token`, 9