Package 'simfinapi'

September 11, 2021

Type Package

Title Accessing 'SimFin' Data

Version 0.2.0

Description Through simfinapi, you can intuitively access the 'SimFin'
Web-API (<https://simfin.com/>) to make 'SimFin' data easily available
in R. To obtain an 'SimFin' API key (and thus to use this package),
you need to register at <https://simfin.com/login>.

License GPL-3

URL <https://github.com/matthiasgomolka/simfinapi>

BugReports <https://github.com/matthiasgomolka/simfinapi/issues>

Depends R (>= 3.5)

Imports bit64, checkmate (>= 2.0.0), data.table (>= 1.12.8),
future.apply (>= 1.4.0), httr, memoise (>= 1.1.0), progressr,
RcppSimdJson (>= 0.1.1), utils

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

Config/testthat/edition 3

Config/testthat/parallel true

Config/testthat/start-first check_inputs

Encoding UTF-8

RoxygenNote 7.1.2

NeedsCompilation no

Author Matthias Gomolka [aut, cre]

Maintainer Matthias Gomolka <matthias.gomolka@posteo.de>

Repository CRAN

Date/Publication 2021-09-11 13:40:02 UTC

R **topics documented:**

---

param_doc *Parameter documentation*

Description

Parameter documentation

Arguments

api_key	[character](#) Your 'SimFin' API key. It's recommended to set the API key globally using [sfa_set_api_key](#).
cache_dir	[character](#) Your cache directory. It's recommended to set the cache directory globally using [sfa_set_cache_dir](#).
sfplus	[logical](#) SetTRUE if you have a SimFin+ account. It's recommended to set sfplus globally using [sfa_set_sfplus](#).
ticker	[integer](#) Ticker of the companies of interest.
simfin_id	[integer](#) 'SimFin' IDs of the companies of interest. Any simfin_id will be internally translated to the respective ticker. This reduces the number of queries in case you query the same company via ticker *and* simfin_id.
start	[Date](#) Filter for the report dates (reserved for SimFin+ users). With this filter you can filter the statements by the date on which the reported period ended ('Report Date'). By specifying a value here, only statements will be retrieved with report dates ending AFTER the specified date.
end	[Date](#) Filter for the report dates (reserved for SimFin+ users). With this filter you can filter the statements by the date on which the reported period ended ('Report Date'). By specifying a value here, only statements will be retrieved with report dates ending BEFORE the specified date.
period	[character](#) Filter for periods. As a non-SimFin+ user, you have to provide exactly one period. As SimFin+ user, this filter can be omitted to retrieve all statements available for the company.

- "q1": First fiscal quarter.

- "q2": Second fiscal quarter.

- "q3": Third fiscal quarter.

- "q4": Fourth fiscal quarter.

- "fy": Full fiscal year.

- "h1": First 6 months of fiscal year.

- "h2": Last 6 months of fiscal year.

- "9m": First nine months of fiscal year.

- "6m": Any fiscal 6 month period (first + second half years; reserved for SimFin+ users).

- "quarters": All quarters (q1 + q2 + q3 + q4; reserved for SimFin+ users).

fyear   integer Filter for fiscal year. As a non-SimFin+ user, you have to provide exactly one fiscal year. As SimFin+ user, this filter can be omitted to retrieve all data available for the company.

ratios   logical With TRUE, you can display some price related ratios along with the share price data (reserved for SimFin+ users). The ratios that will be displayed are:

- Market-Cap

- Price to Earnings Ratio (quarterly)

- Price to Earnings Ratio (ttm)

- Price to Sales Ratio (quarterly)

- Price to Sales Ratio (ttm)

- Price to Book Value (ttm)

- Price to Free Cash Flow (quarterly)

- Price to Free Cash Flow (ttm)

- Enterprise Value (ttm)

- EV/EBITDA (ttm)

- EV/Sales (ttm)

- EV/FCF (ttm)

- Book to Market Value (ttm)

- Operating Income/EV (ttm).

Parallel processing

This function supports parallel processing via future.apply. If your machine has several cores (most have), you can make the API calls in parallel. To do so, define a future::plan() before calling the function.

---

set_clean_names *Clean names*

Description

Cleans column names so that they comply to R naming conventions. Keeps the original names in the label attribute.

Usage

```
set_clean_names(DT)
```

Arguments

DT   A [data.table](#).

Value

A [data.table](#) with cleaned names and labels.

---

sfa_get_entities *Get a table of all available 'SimFin' ID's with ticker and name.*

Description

Get a table of all available 'SimFin' ID's with ticker and name.

Usage

```
sfa_get_entities(
  api_key = getOption("sfa_api_key"),
  cache_dir = getOption("sfa_cache_dir")
)
```

Arguments

api_key   [character](#) Your 'SimFin' API key. It's recommended to set the API key globally using [sfa_set_api_key](#).

cache_dir   [character](#) Your cache directory. It's recommended to set the cache directory globally using [sfa_set_cache_dir](#).

---

sfa_get_info *Get basic company information*

Description

Get basic company information

Usage

```
sfa_get_info(
  ticker = NULL,
  simfin_id = NULL,
  api_key = getOption("sfa_api_key"),
  cache_dir = getOption("sfa_cache_dir"),
  sfplus = getOption("sfa_sfplus", default = FALSE)
)
```

Arguments

ticker	[integer](#) Ticker of the companies of interest.
simfin_id	[integer](#) 'SimFin' IDs of the companies of interest. Any simfin_id will be internally translated to the respective ticker. This reduces the number of queries in case you query the same company via ticker *and* simfin_id.
api_key	[character](#) Your 'SimFin' API key. It's recommended to set the API key globally using [sfa_set_api_key](#).
cache_dir	[character](#) Your cache directory. It's recommended to set the cache directory globally using [sfa_set_cache_dir](#).
sfplus	[logical](#) SetTRUE if you have a SimFin+ account. It's recommended to set sfplus globally using [sfa_set_sfplus](#).

---

sfa_get_prices *Get price data*

Description

Share price data and ratios can be retrieved here. All share prices are adjusted for stock splits. If you are interested in more details, take a look at this page: https://simfin.com/data/help/main?topic=apiv2-prices

Usage

```
sfa_get_prices(
  ticker = NULL,
  simfin_id = NULL,
  ratios = NULL,
  start = NULL,
  end = NULL,
  api_key = getOption("sfa_api_key"),
  cache_dir = getOption("sfa_cache_dir"),
  sfplus = getOption("sfa_sfplus", default = FALSE)
)
```

Arguments

ticker	[integer](#) Ticker of the companies of interest.
simfin_id	[integer](#) 'SimFin' IDs of the companies of interest. Any simfin_id will be internally translated to the respective ticker. This reduces the number of queries in case you query the same company via ticker *and* simfin_id.
ratios	[logical](#) With TRUE, you can display some price related ratios along with the share price data (reserved for SimFin+ users). The ratios that will be displayed are:

- Market-Cap
- Price to Earnings Ratio (quarterly)
- Price to Earnings Ratio (ttm)
- Price to Sales Ratio (quarterly)
- Price to Sales Ratio (ttm)
- Price to Book Value (ttm)
- Price to Free Cash Flow (quarterly)
- Price to Free Cash Flow (ttm)
- Enterprise Value (ttm)
- EV/EBITDA (ttm)
- EV/Sales (ttm)
- EV/FCF (ttm)
- Book to Market Value (ttm)
- Operating Income/EV (ttm).

start	[Date](#) Filter for the report dates (reserved for SimFin+ users). With this filter you can filter the statements by the date on which the reported period ended ('Report Date'). By specifying a value here, only statements will be retrieved with report dates ending AFTER the specified date.
end	[Date](#) Filter for the report dates (reserved for SimFin+ users). With this filter you can filter the statements by the date on which the reported period ended ('Report Date'). By specifying a value here, only statements will be retrieved with report dates ending BEFORE the specified date.
api_key	[character](#) Your 'SimFin' API key. It's recommended to set the API key globally using [sfa_set_api_key](#).

cache_dir	[character](#) Your cache directory. It's recommended to set the cache directory globally using [sfa_set_cache_dir](#).
sfplus	[logical](#) SetTRUE if you have a SimFin+ account. It's recommended to set sfplus globally using [sfa_set_sfplus](#).

Parallel processing

This function supports parallel processing via `future.apply`. If your machine has several cores (most have), you can make the API calls in parallel. To do so, define a `future::plan()` before calling the function.

---

sfa_get_ref   *Download reference data*

Description

Download reference data on industries and markets.

Usage

```
sfa_get_ref(ref_data, api_key = getOption("sfa_api_key"))
```

Arguments

ref_data	[character](#) Either "industries" or "markets".
api_key	[character](#) Your 'SimFin' API key. It's recommended to set the API key globally using [sfa_set_api_key](#).

Value

[data.table](#) containing reference data on "industries" or "markets", depending on ref_data.

---

sfa_get_shares   *Shares Outstanding*

Description

Common shares outstanding (point-in-time) and weighted average basic/diluted shares outstanding for all periods can be retrieved here. These shares are the aggregate figures for the entire company. If you are interested in more details, take a look at this page: https://simfin.com/data/help/main?topic=apiv2-shares

Usage

```
sfa_get_shares(
  ticker = NULL,
  simfin_id = NULL,
  type,
  period = "fy",
  fyear = data.table::year(Sys.Date()) - 1L,
  start = NULL,
  end = NULL,
  api_key = getOption("sfa_api_key"),
  cache_dir = getOption("sfa_cache_dir"),
  sfplus = getOption("sfa_sfplus", default = FALSE)
)
```

Arguments

ticker	integer Ticker of the companies of interest.
simfin_id	integer 'SimFin' IDs of the companies of interest. Any simfin_id will be internally translated to the respective ticker. This reduces the number of queries in case you query the same company via ticker *and* simfin_id.
type	character Type of shares outstanding to be retrieved. • "common": Common shares outstanding. • "wa-basic": Weighted average basic shares outstanding for a period. • "wa-diluted": Weighted average diluted shares outstanding for a period.
period	character Filter for periods. As a non-SimFin+ user, you have to provide exactly one period. As SimFin+ user, this filter can be omitted to retrieve all statements available for the company. • "q1": First fiscal quarter. • "q2": Second fiscal quarter. • "q3": Third fiscal quarter. • "q4": Fourth fiscal quarter. • "fy": Full fiscal year. • "h1": First 6 months of fiscal year. • "h2": Last 6 months of fiscal year. • "9m": First nine months of fiscal year. • "6m": Any fiscal 6 month period (first + second half years; reserved for SimFin+ users). • "quarters": All quarters (q1 + q2 + q3 + q4; reserved for SimFin+ users).
fyear	integer Filter for fiscal year. As a non-SimFin+ user, you have to provide exactly one fiscal year. As SimFin+ user, this filter can be omitted to retrieve all data available for the company.
start	Date Filter for the report dates (reserved for SimFin+ users). With this filter you can filter the statements by the date on which the reported period ended ('Report Date'). By specifying a value here, only statements will be retrieved with report dates ending AFTER the specified date.

end	[Date] Filter for the report dates (reserved for SimFin+ users). With this filter you can filter the statements by the date on which the reported period ended ('Report Date'). By specifying a value here, only statements will be retrieved with report dates ending BEFORE the specified date.
api_key	[character] Your 'SimFin' API key. It's recommended to set the API key globally using [sfa_set_api_key].
cache_dir	[character] Your cache directory. It's recommended to set the cache directory globally using [sfa_set_cache_dir].
sfplus	[logical] SetTRUE if you have a SimFin+ account. It's recommended to set sfplus globally using [sfa_set_sfplus].

Fiscal year

Only works with type = "wa-basic" and type = "wa-diluted".

Parallel processing

This function supports parallel processing via future.apply. If your machine has several cores (most have), you can make the API calls in parallel. To do so, define a future::plan() before calling the function.

---

sfa_get_statement *Get basic company information*

Description

Fundamentals and derived figures can be retrieved here.

Usage

```
sfa_get_statement(
  ticker = NULL,
  simfin_id = NULL,
  statement,
  period = "fy",
  fyear = NULL,
  start = NULL,
  end = NULL,
  ttm = FALSE,
  shares = FALSE,
  api_key = getOption("sfa_api_key"),
  cache_dir = getOption("sfa_cache_dir"),
  sfplus = getOption("sfa_sfplus", default = FALSE)
)
```

Arguments

ticker	[integer] Ticker of the companies of interest.
simfin_id	[integer] 'SimFin' IDs of the companies of interest. Any simfin_id will be internally translated to the respective ticker. This reduces the number of queries in case you query the same company via ticker *and* simfin_id.
statement	[character] Statement to be retrieved. One of

- "pl": Profit & Loss statement
- "bs": Balance Sheet
- "cf": Cash Flow statement
- "derived": Derived figures & fundamental ratios
- "all": Retrieves all 3 statements + ratios. Please note that this option is reserved for SimFin+ users.

period	[character] Filter for periods. As a non-SimFin+ user, you have to provide exactly one period. As SimFin+ user, this filter can be omitted to retrieve all statements available for the company.

- "q1": First fiscal quarter.
- "q2": Second fiscal quarter.
- "q3": Third fiscal quarter.
- "q4": Fourth fiscal quarter.
- "fy": Full fiscal year.
- "h1": First 6 months of fiscal year.
- "h2": Last 6 months of fiscal year.
- "9m": First nine months of fiscal year.
- "6m": Any fiscal 6 month period (first + second half years; reserved for SimFin+ users).
- "quarters": All quarters (q1 + q2 + q3 + q4; reserved for SimFin+ users).

fyear	[integer] Filter for fiscal year. As a non-SimFin+ user, you have to provide exactly one fiscal year. As SimFin+ user, this filter can be omitted to retrieve all data available for the company.
start	[Date] Filter for the report dates (reserved for SimFin+ users). With this filter you can filter the statements by the date on which the reported period ended ('Report Date'). By specifying a value here, only statements will be retrieved with report dates ending AFTER the specified date.
end	[Date] Filter for the report dates (reserved for SimFin+ users). With this filter you can filter the statements by the date on which the reported period ended ('Report Date'). By specifying a value here, only statements will be retrieved with report dates ending BEFORE the specified date.
ttm	[logical] If TRUE, you can return the trailing twelve months statements for all periods, meaning at every available point in time the sum of the last 4 available quarterly figures.
shares	[logical] If TRUE, you can display the weighted average basic & diluted shares outstanding for each period along with the fundamentals. Reserved for SimFin+ users (as non-SimFin+ user, you can still use the shares outstanding endpoints).

api_key	character Your 'SimFin' API key. It's recommended to set the API key globally using sfa_set_api_key.
cache_dir	character Your cache directory. It's recommended to set the cache directory globally using sfa_set_cache_dir.
sfplus	logical SetTRUE if you have a SimFin+ account. It's recommended to set sfplus globally using sfa_set_sfplus.

Value

data.table containing the statement(s) data.

Parallel processing

This function supports parallel processing via future.apply. If your machine has several cores (most have), you can make the API calls in parallel. To do so, define a future::plan() before calling the function.

---

sfa_set_api_key   *Set your 'SimFin' API key globally*

Description

It is highly recommended to set the API key globally as it makes specifying the api_key argument of other sfa_* functions obsolete.

There are two ways to set your API key globally:

1. Provide the API key directly as a string (api_key = "YourApiKey").

2. Create a system-wide environment variable containing you API key and refer to that (env_var = "YourEnvVar"). How to create a system-wide environment variable depends on your operating system.

The second option is recommended because your R scripts won't contain your API key and it is safe to keep your scripts in an open repository like GitHub.

Usage

```
sfa_set_api_key(api_key, env_var)
```

Arguments

api_key	character You API key. Ignored if you specify env_var as well.
env_var	character Name of an environment variable holding you API key, e.g. SIMFIN_API_KEY. Leave empty (NULL, default) if you want to set your API key directly.

Examples

```
## Not run:
# set API key directly
sfa_set_api_key(api_key = "YourApiKey")

# set API key via environment variable
# (this assumes you already created an environment variable called
# 'SIMFIN_API_KEY' which contains you API key)
sfa_set_api_key(env_var = "SIMFIN_API_KEY")

## End(Not run)
```

---

sfa_set_cache_dir *Set cache directory globally*

Description

It is highly recommended to set the cache directory globally. This has two advantages:

1. Results from the 'SimFin' API calls are cached over the end of the session. This is especially interesting if you don't have a SimFin+ account and the number of API calls is limited to 2,000 per day.
2. It makes specifying the cache_dir argument of other sfa_* functions obsolete.

Usage

```
sfa_set_cache_dir(path, create = FALSE)
```

Arguments

path   [character](#) The directory where you want to cache the responses from the 'SimFin' API calls.

create   [logical](#) Set TRUE if you want to create path automatically if it does not yet exist.

---

sfa_set_sfplus *Specify the type of you SimFin account*

Description

If you have a SimFin+ account, it is highly recommended to specify this globally as it makes specifying the sfplus argument of other sfa_* functions obsolete.

You don't need this function if you don't have a SimFin+ account.

Usage

```
sfa_set_sfplus(sfplus = TRUE)
```

Arguments

sfplus   [logical](#) Defaults to `TRUE` to specify that you have a SimFin+ account.

Details

There is no good reason to use `sfa_set_sfplus(FALSE)` as all functions assume this by default.

Examples

```
## Not run:
# Tell simfinapi that you have a SimFin+ account
sfa_set_sfplus()

## End(Not run)
```

Index