

# Package: simfinapi (via r-universe)

September 14, 2024

**Type** Package

**Title** Accessing 'SimFin' Data

**Version** 1.0.0

**Description** Through simfinapi, you can intuitively access the 'SimFin' Web-API (<<https://www.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://app.simfin.com/login>>.

**License** GPL-3

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

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

**Depends** R (>= 3.5)

**Imports** checkmate (>= 2.0.0), data.table (>= 1.12.8), httr2, lifecycle, memoise (>= 1.1.0), RcppSimdJson (>= 0.1.1), utils

**Suggests** knitr, rmarkdown, spelling, testthat

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Config/testthat/parallel** true

**Config/testthat/start-first** check\_inputs

**Encoding** UTF-8

**Language** en-US

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**Repository** <https://matthiasgomolka.r-universe.dev>

**RemoteUrl** <https://github.com/matthiasgomolka/simfinapi>

**RemoteRef** HEAD

**RemoteSha** 1f5a7dde88690098e8d88a254c463372171553c0

## Contents

param_doc . . . . .	2
sfa_load_common_shares_outstanding . . . . .	3
sfa_load_companies . . . . .	4
sfa_load_shareprices . . . . .	5
sfa_load_statements . . . . .	6
sfa_load_weighted_shares_outstanding . . . . .	8
sfa_set_api_key . . . . .	9
sfa_set_cache_dir . . . . .	10

<b>Index</b>	<b>11</b>
--------------	-----------

---

param_doc	<i>Parameter documentation</i>
-----------	--------------------------------

---

### Description

Parameter documentation

### Arguments

api_key	<b>character</b> Your 'SimFin' API key. It's recommended to set the API key globally using <a href="#">sfa_set_api_key</a> .
cache_dir	<b>character</b> Your cache directory. It's recommended to set the cache directory globally using <a href="#">sfa_set_cache_dir</a> .
ticker	<b>integer</b> Ticker of the companies of interest.
id	<b>integer</b> 'SimFin' IDs of the companies of interest. Any id will be internally translated to the respective ticker. This reduces the number of queries in case you query the same company via <i>ticker</i> <i>and</i> <i>id</i> .
asreported	<b>logical</b> If TRUE, retrieves the as-reported (not restated) data.
ttm	<b>logical</b> If TRUE, retrieves trailing twelve month periods.
start	<b>Date</b> 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 <b>AFTER</b> the specified date.
end	<b>Date</b> 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 <b>BEFORE</b> the specified date.
period	<b>character</b> 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. <ul style="list-style-type: none"> <li>'q1': First fiscal quarter.</li> <li>'q2': Second fiscal quarter.</li> </ul>

	<ul style="list-style-type: none"> <li>• 'q3': Third fiscal quarter.</li> <li>• 'q4': Fourth fiscal quarter.</li> <li>• 'fy': Full fiscal year.</li> <li>• 'h1': First 6 months of fiscal year.</li> <li>• 'h2': Last 6 months of fiscal year.</li> <li>• '9m': First nine months of fiscal year.</li> <li>• '6m': Any fiscal 6 month period (first + second half years; reserved for SimFin+ users).</li> <li>• 'quarters': All quarters (q1 + q2 + q3 + q4; reserved for SimFin+ users).</li> </ul>
fyear	<b>integer</b> 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	<b>logical</b> 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: <ul style="list-style-type: none"> <li>• Market-Cap</li> <li>• Price to Earnings Ratio (quarterly)</li> <li>• Price to Earnings Ratio (ttm)</li> <li>• Price to Sales Ratio (quarterly)</li> <li>• Price to Sales Ratio (ttm)</li> <li>• Price to Book Value (ttm)</li> <li>• Price to Free Cash Flow (quarterly)</li> <li>• Price to Free Cash Flow (ttm)</li> <li>• Enterprise Value (ttm)</li> <li>• EV/EBITDA (ttm)</li> <li>• EV/Sales (ttm)</li> <li>• EV/FCF (ttm)</li> <li>• Book to Market Value (ttm)</li> <li>• Operating Income/EV (ttm).</li> </ul>

### 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\_load\_common\_shares\_outstanding

*Common Shares Outstanding*

---

### Description

Get the number of common shares outstanding for a company.

**Usage**

```
sfa_load_common_shares_outstanding(
  id = NULL,
  ticker = NULL,
  start = NULL,
  end = NULL,
  api_key = getOption("sfa_api_key"),
  cache_dir = getOption("sfa_cache_dir")
)
```

**Arguments**

id	<b>integer</b> 'SimFin' IDs of the companies of interest. Any id will be internally translated to the respective ticker. This reduces the number of queries in case you query the same company via <i>ticker</i> <i>and</i> <i>id</i> .
ticker	<b>integer</b> Ticker of the companies of interest.
start	<b>Date</b> 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 <b>AFTER</b> the specified date.
end	<b>Date</b> 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 <b>BEFORE</b> the specified date.
api_key	<b>character</b> Your 'SimFin' API key. It's recommended to set the API key globally using <code>sfa_set_api_key</code> .
cache_dir	<b>character</b> Your cache directory. It's recommended to set the cache directory globally using <code>sfa_set_cache_dir</code> .

**Value**

**data.table** containing the common shares outstanding.

---

sfa\_load\_companies      *List Companies*

---

**Description**

Get a list of all companies in the SimFin database. See <https://simfin.readme.io/reference/list-1> and <https://simfin.readme.io/reference/general-1> for more information.

**Usage**

```
sfa_load_companies(
  api_key = getOption("sfa_api_key"),
  cache_dir = getOption("sfa_cache_dir"),
  details = FALSE
)
```

**Arguments**

api_key	<b>character</b> Your 'SimFin' API key. It's recommended to set the API key globally using <a href="#">sfa_set_api_key</a> .
cache_dir	<b>character</b> Your cache directory. It's recommended to set the cache directory globally using <a href="#">sfa_set_cache_dir</a> .
details	<b>logical(1)</b> : If TRUE, return a more detailed data.table with additional columns. Default: FALSE.

**Value**

data.table::data.table() containing basic company information.

---

sfa\_load\_shareprices *Load share prices*

---

**Description**

Retrieve share price data and ratios.

**Usage**

```
sfa_load_shareprices(
  ticker = NULL,
  id = NULL,
  start = NULL,
  end = NULL,
  ratios = FALSE,
  asreported = FALSE,
  api_key = getOption("sfa_api_key"),
  cache_dir = getOption("sfa_cache_dir")
)
```

**Arguments**

ticker	<b>integer</b> Ticker of the companies of interest.
id	<b>integer</b> 'SimFin' IDs of the companies of interest. Any id will be internally translated to the respective ticker. This reduces the number of queries in case you query the same company via <i>ticker</i> and <i>id</i> .

start	<b>Date</b> 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 <b>AFTER</b> the specified date.
end	<b>Date</b> 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 <b>BEFORE</b> the specified date.
ratios	<b>logical</b> If TRUE, additional ratios and derived metrics are included.
asreported	<b>logical</b> If TRUE, retrieves the as-reported (not restated) data.
api_key	<b>character</b> Your 'SimFin' API key. It's recommended to set the API key globally using <a href="#">sfa_set_api_key</a> .
cache_dir	<b>character</b> Your cache directory. It's recommended to set the cache directory globally using <a href="#">sfa_set_cache_dir</a> .

### Value

[data.table](#) with share price data.

---

sfa\_load\_statements    *Get financial statements*

---

### Description

Fundamentals and derived figures can be retrieved here.

### Usage

```
sfa_load_statements(  
  ticker = NULL,  
  id = NULL,  
  statements,  
  period = "fy",  
  fyear = NULL,  
  start = NULL,  
  end = NULL,  
  ttm = FALSE,  
  asreported = FALSE,  
  api_key = getOption("sfa_api_key"),  
  cache_dir = getOption("sfa_cache_dir")  
)
```

## Arguments

ticker	<a href="#">integer</a> Ticker of the companies of interest.
id	<a href="#">integer</a> 'SimFin' IDs of the companies of interest. Any id will be internally translated to the respective ticker. This reduces the number of queries in case you query the same company via <i>ticker</i> and <i>id</i> .
statements	<a href="#">character</a> vector of statements, available values: pl (Profit & Loss), bs (Balance Sheet), cf (Cash Flow), derived (Derived Ratios and Indicators).
period	<a href="#">character</a> 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. <ul style="list-style-type: none"><li>• 'q1': First fiscal quarter.</li><li>• 'q2': Second fiscal quarter.</li><li>• 'q3': Third fiscal quarter.</li><li>• 'q4': Fourth fiscal quarter.</li><li>• 'fy': Full fiscal year.</li><li>• 'h1': First 6 months of fiscal year.</li><li>• 'h2': Last 6 months of fiscal year.</li><li>• '9m': First nine months of fiscal year.</li><li>• '6m': Any fiscal 6 month period (first + second half years; reserved for SimFin+ users).</li><li>• 'quarters': All quarters (q1 + q2 + q3 + q4; reserved for SimFin+ users).</li></ul>
fyear	<a href="#">integer</a> 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	<a href="#">Date</a> 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	<a href="#">Date</a> 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	<a href="#">logical</a> If TRUE, retrieves trailing twelve month periods.
asreported	<a href="#">logical</a> If TRUE, retrieves the as-reported (not restated) data.
api_key	<a href="#">character</a> Your 'SimFin' API key. It's recommended to set the API key globally using <a href="#">sfa_set_api_key</a> .
cache_dir	<a href="#">character</a> Your cache directory. It's recommended to set the cache directory globally using <a href="#">sfa_set_cache_dir</a> .

## 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\_load\_weighted\_shares\_outstanding  
*Weighted Shares Outstanding*

---

## Description

Get the number of weighted shares outstanding for a company.

## Usage

```
sfa_load_weighted_shares_outstanding(  
  id = NULL,  
  ticker = NULL,  
  fyear = NULL,  
  period = NULL,  
  start = NULL,  
  end = NULL,  
  ttm = NULL,  
  api_key = getOption("sfa_api_key"),  
  cache_dir = getOption("sfa_cache_dir")  
)
```

## Arguments

id	<b>integer</b> 'SimFin' IDs of the companies of interest. Any id will be internally translated to the respective ticker. This reduces the number of queries in case you query the same company via <i>ticker</i> <i>and</i> <i>id</i> .
ticker	<b>integer</b> Ticker of the companies of interest.
fyear	<b>integer</b> 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.
period	<b>character</b> 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. <ul style="list-style-type: none"><li>• 'q1': First fiscal quarter.</li><li>• 'q2': Second fiscal quarter.</li><li>• 'q3': Third fiscal quarter.</li><li>• 'q4': Fourth fiscal quarter.</li><li>• 'fy': Full fiscal year.</li></ul>



- '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).

start	<b>Date</b> 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 <b>AFTER</b> the specified date.
end	<b>Date</b> 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 <b>BEFORE</b> the specified date.
ttm	<b>logical</b> If TRUE, retrieves trailing twelve month periods.
api_key	<b>character</b> Your 'SimFin' API key. It's recommended to set the API key globally using <a href="#">sfa_set_api_key</a> .
cache_dir	<b>character</b> Your cache directory. It's recommended to set the cache directory globally using <a href="#">sfa_set_cache_dir</a> .

**Value**

[data.table](#) containing the common shares outstanding.

---

<code>sfa_set_api_key</code>	<i>Set your 'SimFin' API key globally</i>
------------------------------	---

---

**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.

**Value**

**character** The API key (invisibly).

**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.

**Value**

**character** The cache directory (invisibly).

# Index

character, [2](#), [4-10](#)

data.table, [4](#), [6](#), [7](#), [9](#)

Date, [2](#), [4](#), [6](#), [7](#), [9](#)

integer, [2-5](#), [7](#), [8](#)

logical, [2](#), [3](#), [6](#), [7](#), [9](#), [10](#)

param\_doc, [2](#)

sfa\_load\_common\_shares\_outstanding, [3](#)

sfa\_load\_companies, [4](#)

sfa\_load\_shareprices, [5](#)

sfa\_load\_statements, [6](#)

sfa\_load\_weighted\_shares\_outstanding,  
[8](#)

sfa\_set\_api\_key, [2](#), [4-7](#), [9](#), [9](#)

sfa\_set\_cache\_dir, [2](#), [4-7](#), [9](#), [10](#)