

Package ‘yfinancer’

May 8, 2026

Title 'Yahoo Finance' API Wrapper

Version 0.1.3

Description Download financial market data, company information, financial statements, options data, and more from the unofficial 'Yahoo Finance' API.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

URL <https://github.com/gacolitti/yfinancer>

BugReports <https://github.com/gacolitti/yfinancer/issues>

Suggests devtools, mockery, testthat, usethis, withr

Config/testthat/edition 3

Imports htr2, jsonlite, purrr, dplyr, lubridate, rlang, stringr, tidy, glue, utils

Depends R (>= 4.1.0)

LazyData true

NeedsCompilation no

Author Giovanni Colitti [aut, cre, cph]

Maintainer Giovanni Colitti <g.a.colitti@gmail.com>

Repository CRAN

Date/Publication 2025-04-17 21:30:02 UTC

Contents

get_balance_sheet	2
get_cashflow	4
get_financials	5
get_history	7
get_income_statement	9
get_info	10

get_tickers	13
get_tickers_balance_sheet	14
get_tickers_cashflow	15
get_tickers_financials	16
get_tickers_history	17
get_tickers_income_statement	19
get_tickers_info	20
get_user_agent	21
print.yf_ticker	21
print.yf_tickers	22
search_tickers	22
validate_tickers	24
valid_balance_keys	24
valid_cashflow_keys	25
valid_income_keys	26
valid_modules	26

Index	28
--------------	-----------

get_balance_sheet	<i>Get balance sheet for a ticker</i>
-------------------	---------------------------------------

Description

Retrieves balance sheet data from Yahoo Finance for a specified ticker symbol. Balance sheets show a company's assets, liabilities, and shareholders' equity at a specific point in time.

Usage

```
get_balance_sheet(
  ticker,
  freq = c("annual", "quarterly"),
  start = NULL,
  end = NULL,
  balance_keys = NULL,
  pretty = TRUE,
  wide = TRUE,
  proxy = NULL,
  output = c("tibble", "response", "request")
)
```

Arguments

ticker	A ticker object created with get_tickers() or a ticker symbol string
freq	Frequency of data: "annual" or "quarterly" (default "annual")
start	Start timestamp as date, datetime, or string (default EOY 2016)
end	End timestamp as date, datetime, or string (default current timestamp)

<code>balance_keys</code>	Vector of specific balance sheet keys to include (default all) See <code>valid_balance_keys</code> for available options.
<code>pretty</code>	Format column names to be more readable (default TRUE)
<code>wide</code>	Return data in wide format with dates as columns (default TRUE). If FALSE, returns data in long format with a date column.
<code>proxy</code>	Optional proxy settings for the request
<code>output</code>	Object to return. Can be "tibble", "response", or "request" (default "tibble")

Value

Either a tibble with balance sheet data, an httr2 response object, or an httr2 request object depending on the value of the output argument.

Available Balance Keys

Examples:

- TotalAssets
- TotalCapitalization
- CurrentAssets

See `valid_balance_keys` for a full list of available balance keys.

Examples

```
## Not run:
apple <- get_tickers("AAPL")

# Get annual balance sheet
balance_sheet <- get_balance_sheet(apple)

# Get quarterly balance sheet
quarterly_balance <- get_balance_sheet(apple, freq = "quarterly")

# Get specific balance sheet items
assets_liabilities <- get_balance_sheet(apple,
  balance_keys = c("TotalAssets", "TotalLiabilities")
)

# Get data for a specific time period
balance_2020_2022 <- get_balance_sheet(apple,
  start = "2020-01-01",
  end = "2022-12-31"
)

## End(Not run)
```

 get_cashflow

Get cash flow statement for a ticker

Description

Retrieves cash flow statement data from Yahoo Finance for a specified ticker symbol. Cash flow statements show how changes in balance sheet accounts and income affect cash and cash equivalents, breaking the analysis down to operating, investing, and financing activities.

Usage

```
get_cashflow(
  ticker,
  freq = c("annual", "quarterly"),
  start = NULL,
  end = NULL,
  cashflow_keys = NULL,
  pretty = TRUE,
  wide = TRUE,
  proxy = NULL,
  output = c("tibble", "response", "request")
)
```

Arguments

ticker	A ticker object created with <code>get_tickers()</code> or a ticker symbol string
freq	Frequency of data: "annual" or "quarterly" (default "annual")
start	Start timestamp as date, datetime, or string (default EOY 2016)
end	End timestamp as date, datetime, or string (default current timestamp)
cashflow_keys	Vector of specific cash flow statement keys to include (default all) See <code>valid_cashflow_keys</code> for available options.
pretty	Format column names to be more readable (default TRUE)
wide	Return data in wide format with dates as columns (default TRUE). If FALSE, returns data in long format with a date column.
proxy	Optional proxy settings for the request
output	Object to return. Can be "tibble", "response", or "request" (default "tibble")

Value

Either a tibble with cash flow statement data, an `httr2` response object, or an `httr2` request object depending on the value of the output argument.

Available Cashflow Keys

Examples:

- OperatingCashFlow
- FreeCashFlow

See `valid_cashflow_keys` for a full list of available cashflow keys.

Examples

```
## Not run:
apple <- get_tickers("AAPL")

# Get annual cash flow statement
cash_flow <- get_cashflow(apple)

# Get quarterly cash flow statement
quarterly_cash_flow <- get_cashflow(apple, freq = "quarterly")

# Get specific cash flow items
operating_cash <- get_cashflow(apple,
  cashflow_keys = c("OperatingCashFlow", "FreeCashFlow")
)

# Get data for a specific time period
cash_2020_2022 <- get_cashflow(apple,
  start = "2020-01-01",
  end = "2022-12-31"
)

## End(Not run)
```

get_financials

Get all financial statements for a ticker

Description

Retrieves all three main financial statements (income statement, balance sheet, and cash flow statement) from Yahoo Finance for a specified ticker symbol in a single call. This is a convenience function that calls the individual statement functions and returns the results as a list.

Usage

```
get_financials(
  ticker,
  freq = "annual",
  start = NULL,
  end = NULL,
  cashflow_keys = NULL,
```

```

balance_keys = NULL,
income_keys = NULL,
pretty = TRUE,
wide = TRUE,
proxy = NULL,
output = c("tibble", "response", "request")
)

```

Arguments

ticker	A ticker object created with <code>get_tickers()</code> or a ticker symbol string
freq	Frequency of data: "annual" or "quarterly" (default "annual")
start	Start timestamp as date, datetime, or string (default EOY 2016)
end	End timestamp as date, datetime, or string (default current timestamp)
cashflow_keys	Vector of specific cash flow statement keys to include (default all) See <code>valid_cashflow_keys</code> for available options.
balance_keys	Vector of specific balance sheet keys to include (default all) See <code>valid_balance_keys</code> for available options.
income_keys	Vector of specific income statement keys to include (default all) See <code>valid_income_keys</code> for available options.
pretty	Format column names to be more readable (default TRUE)
wide	Return data in wide format with dates as columns (default TRUE). If FALSE, returns data in long format with a date column.
proxy	Optional proxy settings for the request
output	Object to return. Can be "tibble", "response", or "request" (default "tibble")

Details

Note that this function makes multiple API calls to Yahoo Finance. Be aware of potential rate limiting issues when making frequent requests. If you encounter HTTP 429 (Too Many Requests) errors, consider implementing a delay between requests or using a proxy.

Value

A list containing three elements:

- `income_statement`: Income statement data
- `balance_sheet`: Balance sheet data
- `cashflow`: Cash flow statement data

If output is "request" or "response", returns a list of `httr2` request or response objects instead.

Examples

```
## Not run:
apple <- get_tickers("AAPL")

# Get all annual financial statements
financials <- get_financials(apple)

# Access individual statements from the results
income <- financials$income_statement
balance <- financials$balance_sheet
cashflow <- financials$cashflow

# Get all quarterly financial statements
quarterly_financials <- get_financials(apple, freq = "quarterly")

# Get financial statements for a specific time period
financials_2020_2022 <- get_financials(apple,
  start = "2020-01-01",
  end = "2022-12-31"
)

## End(Not run)
```

`get_history`*Get historical market data for a ticker*

Description

Retrieves historical price data from Yahoo Finance for a specified ticker symbol.

Usage

```
get_history(
  ticker,
  period = "1mo",
  interval = "1d",
  start = NULL,
  end = NULL,
  prepost = FALSE,
  auto_adjust = TRUE,
  back_adjust = TRUE,
  repair = TRUE,
  proxy = NULL,
  output = c("tibble", "response", "request")
)
```

Arguments

ticker	A ticker name or ticker object created with <code>get_tickers()</code> .
period	The period to download data for (default "1mo"). Valid values are "1d", "5d", "1mo", "3mo", "6mo", "1y", "2y", "5y", "10y", "ytd", "max". Ignored if start and end are provided.
interval	The interval between data points (default "1d"). Valid values are "1m", "2m", "5m", "15m", "30m", "60m", "90m", "1h", "1d", "5d", "1wk", "1mo", "3mo".
start	Start time for query expressed as a date, datetime, or string in YYYY-MM-DD HH:MM:SS format.
end	End time for query expressed as a date, datetime, or string in YYYY-MM-DD HH:MM:SS format.
prepost	Include pre and post market data (default FALSE)
auto_adjust	Adjust all OHLC automatically (default TRUE)
back_adjust	Adjust data to reflect splits and dividends (default TRUE)
repair	Repair missing data (default TRUE)
proxy	Optional proxy settings
output	Object to return. Can be "tibble", "response", or "request" (default "tibble")

Value

Either a tibble with historical market data, an httr2 response object, or an httr2 request object depending on the value of the output argument.

Examples

```
## Not run:
apple <- get_tickers("AAPL")
# Get 1 month of daily data
apple_history <- get_history(apple)
# Get 1 year of daily data
apple_history_1y <- get_history(apple, period = "1y")
# Get custom date range
apple_history_custom <- get_history(
  apple,
  start = "2022-01-01",
  end = "2022-12-31"
)

## End(Not run)
```

get_income_statement *Get income statement for a ticker*

Description

Retrieves income statement data from Yahoo Finance for a specified ticker symbol. Income statements show a company's revenues, expenses, and profits over a specific period.

Usage

```
get_income_statement(  
  ticker,  
  freq = c("annual", "quarterly"),  
  start = NULL,  
  end = NULL,  
  income_keys = NULL,  
  pretty = TRUE,  
  wide = TRUE,  
  proxy = NULL,  
  output = c("tibble", "response", "request")  
)
```

Arguments

ticker	A ticker object created with <code>get_tickers()</code> or a ticker symbol string
freq	Frequency of data: "annual" or "quarterly" (default "annual")
start	Start timestamp as date, datetime, or string (default EOY 2016)
end	End timestamp as date, datetime, or string (default current timestamp)
income_keys	Vector of specific income statement keys to include (default all) See <code>valid_income_keys</code> for available options.
pretty	Format column names to be more readable (default TRUE)
wide	Return data in wide format with dates as columns (default TRUE). If FALSE, returns data in long format with a date column.
proxy	Optional proxy settings for the request
output	Object to return. Can be "tibble", "response", or "request" (default "tibble")

Value

Either a tibble with income statement data, an `httr2` response object, or an `httr2` request object depending on the value of the `output` argument.

Available Income Keys

Examples:

- TotalRevenue
- GrossProfit
- OperatingIncome
- NetIncome

See `valid_income_keys` for a full list of available income keys.

Examples

```
## Not run:
apple <- get_tickers("AAPL")

# Get annual income statement
income_stmt <- get_income_statement(apple)

# Get quarterly income statement
quarterly_income <- get_income_statement(apple, freq = "quarterly")

# Get specific income statement items
revenue_income <- get_income_statement(apple,
  income_keys = c("TotalRevenue", "NetIncome")
)

# Get data for a specific time period
income_2020_2022 <- get_income_statement(apple,
  start = "2020-01-01",
  end = "2022-12-31"
)

## End(Not run)
```

get_info

Retrieve asset information from Yahoo Finance

Description

This function retrieves detailed asset information from Yahoo Finance's `quoteSummary` API. It can fetch various types of data including asset profiles, financial statements, key statistics, and more. The function supports retrieving multiple data modules in a single request.

Usage

```
get_info(
  ticker,
  modules = "summaryProfile",
  output = c("tibble", "list", "response", "request"),
  proxy = NULL
)
```

Arguments

ticker	A ticker name or ticker object created with <code>get_tickers()</code> .
modules	Character vector of module names to retrieve. Default is "summaryProfile". See section "Available Modules" for common options.
output	The type of output to return. Can be "tibble" (default), "list" (raw parsed JSON), "response" (httr2 response), or "request" (httr2 request).
proxy	Optional proxy settings for the request.

Value

Depending on the output parameter and number of modules requested:

- For a single module with `output="tibble"`: A tibble containing the module data
- For multiple modules with `output="tibble"`: A named list of tibbles, one per module
- For `output="list"`: The raw parsed JSON data
- For `output="response"`: The httr2 response object
- For `output="request"`: The httr2 request object

Available Modules

The `modules` parameter accepts any of the valid module names from Yahoo Finance API. Common modules include:

- "assetProfile": Asset overview, description, industry, sector, officers
- "summaryProfile": Brief asset profile information
- "financialData": Key financial metrics and ratios
- "defaultKeyStatistics": Important statistics like market cap, P/E ratio
- "incomeStatementHistory": Annual income statements
- "incomeStatementHistoryQuarterly": Quarterly income statements
- "balanceSheetHistory": Annual balance sheets
- "balanceSheetHistoryQuarterly": Quarterly balance sheets
- "cashflowStatementHistory": Annual cash flow statements
- "cashflowStatementHistoryQuarterly": Quarterly cash flow statements

See `valid_modules` for a complete list of available modules.

Authentication

This function requires authentication via two components:

- **A1 Cookie:** Session identifier
- **Crumb:** Security token

Authentication methods (in order of priority):

1. Environment variables: YFINANCE_CRUMB and YFINANCE_A1
2. Saved auth file: ~/.yfinance/auth
3. Auto-generated using curl-impersonate (requires installation)

Example:

```
Sys.setenv(YFINANCE_CRUMB = "your-crumb")
Sys.setenv(YFINANCE_A1 = "your-a1-cookie")
```

See <https://github.com/lwthiker/curl-impersonate> for curl-impersonate installation. Option 3 above expects curl_chrome110 to be installed and available in the system path.

Examples

```
## Not run:
# Get a single ticker
apple <- get_tickers("AAPL")

# Get summary information
# using default module "summaryProfile"
apple_summary <- get_info(apple)

# Get basic company profile
apple_profile <- get_info(apple, modules = "assetProfile")

# Get key financial metrics
apple_financials <- get_info(apple, modules = "financialData")

# Get multiple modules as a list of tibbles
apple_data <- get_info(apple,
  modules = c("incomeStatementHistory", "balanceSheetHistory", "cashflowStatementHistory")
)

# Access specific financial statements
income_statement <- apple_data$incomeStatementHistory
balance_sheet <- apple_data$balanceSheetHistory

# Get raw JSON response for custom processing
apple_raw <- get_info(apple, modules = "assetProfile", output = "response")

## End(Not run)
```

`get_tickers`*Get Ticker Objects*

Description

Creates one or more ticker objects for accessing data for ticker symbols. This function handles both single and multiple ticker symbols and validates the provided ticker symbols.

Usage

```
get_tickers(..., proxy = NULL)
```

Arguments

<code>...</code>	One or more ticker symbols as separate arguments (e.g., "AAPL", "MSFT")
<code>proxy</code>	Optional proxy settings

Value

For a single symbol: A list containing ticker data and methods with class "yf_ticker" For multiple symbols: A list containing multiple ticker objects with class "yf_tickers"

Rate Limiting

Yahoo Finance does not provide official API documentation or rate limits. Based on community observations, there are approximate limits of a few hundred requests per day from a single IP address before throttling may occur. When working with multiple tickers, consider:

- Batching requests when possible
- Adding delays between requests using `Sys.sleep()`
- Caching results for frequently accessed tickers
- Using the batch functions (e.g., `get_tickers_history()`) instead of individual calls

Examples

```
## Not run:  
# Get a single ticker  
apple <- get_tickers("AAPL")  
  
# Get historical data for a single ticker  
apple_history <- get_history(apple)  
  
# Get company information for a single ticker  
apple_info <- get_info(apple)  
  
# Get multiple tickers  
tech_tickers <- get_tickers("AAPL", "MSFT", "GOOG")
```

```
# Get information for multiple tickers
tech_info <- get_tickers_info(tech_tickers)

# Get historical data for multiple tickers
tech_history <- get_tickers_history(tech_tickers, period = "1y")

## End(Not run)
```

```
get_tickers_balance_sheet
```

Get balance sheet for multiple tickers

Description

Retrieves balance sheet data from Yahoo Finance for multiple specified ticker symbols. Balance sheets show a company's assets, liabilities, and shareholders' equity at a specific point in time.

Usage

```
get_tickers_balance_sheet(
  tickers_obj,
  freq = c("annual", "quarterly"),
  start = NULL,
  end = NULL,
  balance_keys = NULL,
  pretty = TRUE,
  wide = TRUE,
  proxy = NULL,
  output = c("tibble", "response", "request")
)
```

Arguments

tickers_obj	A tickers object created with get_tickers()
freq	Frequency of data: "annual" or "quarterly" (default "annual")
start	Start timestamp as date, datetime, or string (default EOY 2016)
end	End timestamp as date, datetime, or string (default current timestamp)
balance_keys	Vector of specific balance sheet keys to include (default all) See valid_balance_keys for available options.
pretty	Format column names to be more readable (default TRUE)
wide	Return data in wide format with dates as columns (default TRUE). If FALSE, returns data in long format with a date column.
proxy	Optional proxy settings for the request
output	Object to return. Can be "tibble", "response", or "request" (default "tibble")

Details

See `get_balance_sheet` for more details on the balance sheet.

Value

A list of tibbles with balance sheet data for each ticker

Examples

```
## Not run:
tech_tickers <- get_tickers(c("AAPL", "MSFT", "GOOG"))
tech_balance <- get_tickers_balance_sheet(tech_tickers)

## End(Not run)
```

`get_tickers_cashflow` *Get cash flow statement for multiple tickers*

Description

Retrieves cash flow statement data from Yahoo Finance for multiple specified ticker symbols. Cash flow statements show how changes in balance sheet accounts and income affect cash and cash equivalents, breaking the analysis down to operating, investing, and financing activities.

Usage

```
get_tickers_cashflow(
  tickers_obj,
  freq = c("annual", "quarterly"),
  start = NULL,
  end = NULL,
  cashflow_keys = NULL,
  pretty = TRUE,
  wide = TRUE,
  proxy = NULL,
  output = c("tibble", "response", "request")
)
```

Arguments

<code>tickers_obj</code>	A tickers object created with <code>get_tickers()</code>
<code>freq</code>	Frequency of data: "annual" or "quarterly" (default "annual")
<code>start</code>	Start timestamp as date, datetime, or string (default EOY 2016)
<code>end</code>	End timestamp as date, datetime, or string (default current timestamp)
<code>cashflow_keys</code>	Vector of specific cash flow statement keys to include (default all) See <code>valid_cashflow_keys</code> for available options.

pretty	Format column names to be more readable (default TRUE)
wide	Return data in wide format with dates as columns (default TRUE). If FALSE, returns data in long format with a date column.
proxy	Optional proxy settings for the request
output	Object to return. Can be "tibble", "response", or "request" (default "tibble")

Details

See `get_cashflow` for more details on the cash flow statement.

Value

A list of tibbles with cash flow statement data for each ticker

Examples

```
## Not run:
tech_tickers <- get_tickers(c("AAPL", "MSFT", "GOOG"))
tech_cashflow <- get_tickers_cashflow(tech_tickers)

## End(Not run)
```

`get_tickers_financials`

Get all financial statements for multiple tickers

Description

Get all financial statements for multiple tickers

Usage

```
get_tickers_financials(
  tickers_obj,
  freq = c("annual", "quarterly"),
  start = NULL,
  end = NULL,
  cashflow_keys = NULL,
  balance_keys = NULL,
  income_keys = NULL,
  pretty = TRUE,
  wide = TRUE,
  proxy = NULL,
  output = c("tibble", "response", "request")
)
```

Arguments

tickers_obj	A tickers object created with get_tickers()
freq	Frequency of data: "annual" or "quarterly" (default "annual")
start	Start timestamp as date, datetime, or string (default EOY 2016)
end	End timestamp as date, datetime, or string (default current timestamp)
cashflow_keys	Vector of specific cash flow statement keys to include (default all) See valid_cashflow_keys for available options.
balance_keys	Vector of specific balance sheet keys to include (default all) See valid_balance_keys for available options.
income_keys	Vector of specific income statement keys to include (default all) See valid_income_keys for available options.
pretty	Format column names to be more readable (default TRUE)
wide	Return data in wide format with dates as columns (default TRUE). If FALSE, returns data in long format with a date column.
proxy	Optional proxy settings for the request
output	Object to return. Can be "tibble", "response", or "request" (default "tibble")

Value

A nested list containing financial statements for each ticker

Examples

```
## Not run:
tech_tickers <- get_tickers(c("AAPL", "MSFT", "GOOG"))
tech_financials <- get_tickers_financials(tech_tickers)

## End(Not run)
```

get_tickers_history *Get historical data for multiple tickers*

Description

Retrieves historical market data from Yahoo Finance for multiple specified ticker symbols.

Usage

```
get_tickers_history(
  tickers_obj,
  period = "1mo",
  interval = "1d",
  start = NULL,
  end = NULL,
```

```

prepost = FALSE,
auto_adjust = TRUE,
back_adjust = TRUE,
repair = TRUE,
output = c("tibble", "response", "request"),
proxy = NULL
)

```

Arguments

tickers_obj	A tickers object created with get_tickers()
period	The period to download data for (default "1mo"). Valid values are "1d", "5d", "1mo", "3mo", "6mo", "1y", "2y", "5y", "10y", "ytd", "max". Ignored if start and end are provided.
interval	The interval between data points (default "1d"). Valid values are "1m", "2m", "5m", "15m", "30m", "60m", "90m", "1h", "1d", "5d", "1wk", "1mo", "3mo".
start	Start time for query expressed as a date, datetime, or string in YYYY-MM-DD HH:MM:SS format.
end	End time for query expressed as a date, datetime, or string in YYYY-MM-DD HH:MM:SS format.
prepost	Include pre and post market data (default FALSE)
auto_adjust	Adjust all OHLC automatically (default TRUE)
back_adjust	Adjust data to reflect splits and dividends (default TRUE)
repair	Repair missing data (default TRUE)
output	Object to return. Can be "tibble", "response", or "request" (default "tibble")
proxy	Optional proxy settings

Details

See get_history for more details on the historical market data.

Value

A list of tibbles with historical market data for each ticker

Examples

```

## Not run:
tech_tickers <- get_tickers(c("AAPL", "MSFT", "GOOG"))
tech_history <- get_tickers_history(tech_tickers, period = "1y")

## End(Not run)

```

`get_tickers_income_statement`*Get income statement for multiple tickers*

Description

Retrieves income statement data from Yahoo Finance for multiple specified ticker symbols. Income statements show a company's revenues, expenses, and profits over a specific period.

Usage

```
get_tickers_income_statement(  
  tickers_obj,  
  freq = c("annual", "quarterly"),  
  start = NULL,  
  end = NULL,  
  income_keys = NULL,  
  pretty = TRUE,  
  wide = TRUE,  
  proxy = NULL,  
  output = c("tibble", "response", "request")  
)
```

Arguments

<code>tickers_obj</code>	A tickers object created with <code>get_tickers()</code>
<code>freq</code>	Frequency of data: "annual" or "quarterly" (default "annual")
<code>start</code>	Start timestamp as date, datetime, or string (default EOY 2016)
<code>end</code>	End timestamp as date, datetime, or string (default current timestamp)
<code>income_keys</code>	Vector of specific income statement keys to include (default all) See <code>valid_income_keys</code> for available options.
<code>pretty</code>	Format column names to be more readable (default TRUE)
<code>wide</code>	Return data in wide format with dates as columns (default TRUE). If FALSE, returns data in long format with a date column.
<code>proxy</code>	Optional proxy settings for the request
<code>output</code>	Object to return. Can be "tibble", "response", or "request" (default "tibble")

Details

See `get_income_statement` for more details on the income statement.

Value

A list of tibbles with income statement data for each ticker

Examples

```
## Not run:
tech_tickers <- get_tickers(c("AAPL", "MSFT", "GOOG"))
tech_income <- get_tickers_income_statement(tech_tickers)

## End(Not run)
```

get_tickers_info

Get information for multiple tickers

Description

Retrieves company information from Yahoo Finance for multiple specified ticker symbols.

Usage

```
get_tickers_info(
  tickers_obj,
  modules = "summaryProfile",
  output = c("tibble", "response", "request"),
  proxy = NULL
)
```

Arguments

tickers_obj	A tickers object created with get_tickers()
modules	Character vector of module names to retrieve. Default is "summaryProfile". See section "Available Modules" for common options.
output	The type of output to return. Can be "tibble" (default), "list" (raw parsed JSON), "response" (httr2 response), or "request" (httr2 request).
proxy	Optional proxy settings for the request.

Details

See get_info for more details on the company information.

Value

A list of information for each ticker

Examples

```
## Not run:
tech_tickers <- get_tickers(c("AAPL", "MSFT", "GOOG"))
tech_info <- get_tickers_info(tech_tickers)

## End(Not run)
```

get_user_agent	<i>Get a random user agent</i>
----------------	--------------------------------

Description

Get a random user agent

Usage

```
get_user_agent()
```

Value

A character string of a random user agent

print.yf_ticker	<i>Format ticker object print output</i>
-----------------	--

Description

Format ticker object print output

Usage

```
## S3 method for class 'yf_ticker'  
print(x, ...)
```

Arguments

x	The ticker object
...	Additional arguments passed to print

Value

The ticker object (invisibly)

`print.yf_tickers` *Format tickers object print output*

Description

Format tickers object print output

Usage

```
## S3 method for class 'yf_tickers'  
print(x, ...)
```

Arguments

`x` The tickers object
`...` Additional arguments passed to print

Value

The tickers object (invisibly)

`search_tickers` *Search for ticker symbols, companies, and news from Yahoo Finance*

Description

This function allows you to search for ticker symbols, companies, ETFs, etc. using the Yahoo Finance search API. It can also return related news articles. The search functionality is particularly useful for discovering ticker symbols when you only know the company name or part of it.

Usage

```
search_tickers(  
  query,  
  limit = 10,  
  quotes_only = TRUE,  
  fuzzy_query = FALSE,  
  lists_count = 0,  
  enable_research = FALSE,  
  proxy = NULL,  
  output = c("tibble", "response", "request", "all")  
)
```

Arguments

query	Search query string
limit	Maximum number of results to return (default 10)
quotes_only	Return only quotes/tickers, not news (default TRUE)
fuzzy_query	Enable fuzzy search for typos (default FALSE)
lists_count	Number of lists to retrieve (default 0)
enable_research	Include research reports (default FALSE)
proxy	Optional proxy settings
output	Object to return. Can be "tibble", "response", "request", or "all" (default "tibble")

Value

Depending on output parameter:

- "tibble": quotes data (or a list with quotes and news if quotes_only is FALSE)
- "response": raw response from the API
- "request": the request object
- "all": a named list containing all results: quotes, news, lists, research, and raw response

Rate Limiting

Yahoo Finance does not provide official API documentation or rate limits. Based on community observations, search queries may be more heavily rate-limited than other endpoints. To avoid rate limiting:

- Limit the frequency of search requests
- Cache search results when possible
- Consider adding delays between requests (e.g., using `Sys.sleep()`)
- Use more specific search queries to reduce the number of needed requests

Examples

```
## Not run:
# Search for Apple
apple_results <- search_tickers("Apple")

# Search for tech companies with more results
tech_results <- search_tickers("tech", limit = 20)

# Get both quotes and news articles
apple_with_news <- search_tickers("Apple", quotes_only = FALSE)

# Get all data including lists and research reports
all_apple_data <- search_tickers(
```

```

    "Apple",
    quotes_only = FALSE,
    lists_count = 5,
    enable_research = TRUE,
    output = "all"
  )

## End(Not run)

```

validate_tickers	<i>Validate Yahoo Finance Ticker Symbols</i>
------------------	--

Description

This function validates whether given ticker symbols are valid by making an API request to Yahoo Finance's validation endpoint.

Usage

```
validate_tickers(symbols = NULL)
```

Arguments

`symbols` A character vector of ticker symbols to validate

Value

A tibble with validation results for each symbol

Examples

```

## Not run:
validate_tickers(c("AAPL", "MSFT", "GOOG"))

## End(Not run)

```

valid_balance_keys	<i>Data file containing possible balance sheet keys from Yahoo Finance API</i>
--------------------	--

Description

A dataset containing the possible keys that can be found in balance sheet data retrieved from Yahoo Finance's API. These keys are used to identify and extract specific balance sheet line items.

Usage

```
valid_balance_keys
```

Format

A character vector containing 147 possible balance sheet keys

Source

Yahoo Finance API

Examples

```
# View all possible balance sheet keys
data(valid_balance_keys)
head(valid_balance_keys)

# Check if a specific key exists
"TotalAssets" %in% valid_balance_keys
```

valid_cashflow_keys	<i>Data file containing possible cash flow statement keys from Yahoo Finance API</i>
---------------------	--

Description

A dataset containing the possible keys that can be found in cash flow statement data retrieved from Yahoo Finance's API. These keys are used to identify and extract specific cash flow line items.

Usage

```
valid_cashflow_keys
```

Format

A character vector containing 131 possible cash flow statement keys

Source

Yahoo Finance API

Examples

```
# View all possible cash flow keys
data(valid_cashflow_keys)
head(valid_cashflow_keys)

# Check if a specific key exists
"OperatingCashFlow" %in% valid_cashflow_keys
```

valid_income_keys	<i>Data file containing possible income statement keys from Yahoo Finance API</i>
-------------------	---

Description

A dataset containing the possible keys that can be found in income statement data retrieved from Yahoo Finance's API. These keys are used to identify and extract specific income statement line items.

Usage

```
valid_income_keys
```

Format

A character vector containing 103 possible income statement keys

Source

Yahoo Finance API

Examples

```
# View all possible income statement keys
data(valid_income_keys)
head(valid_income_keys)

# Check if a specific key exists
"NetIncome" %in% valid_income_keys
```

valid_modules	<i>Data file containing possible Yahoo Finance API modules</i>
---------------	--

Description

A dataset containing the possible modules that can be requested from the Yahoo Finance API. Each module represents a specific type of financial data that can be retrieved for a ticker.

Usage

```
valid_modules
```

Format

A character vector containing 33 possible API modules with descriptions

Details

The modules include:

- `assetProfile` - Summary profile and company officers
- `balanceSheetHistory` - Annual balance sheet data
- `balanceSheetHistoryQuarterly` - Quarterly balance sheet data
- `calendarEvents` - Future earnings dates
- `cashFlowStatementHistory` - Annual cash flow statement data
- `cashFlowStatementHistoryQuarterly` - Quarterly cash flow statement data
- `defaultKeyStatistics` - Key performance indicators (PE, enterprise value, EPS, etc.)
- `earnings` - Earnings history
- `earningsHistory` - Historical earnings data
- `earningsTrend` - Earnings trend data
- `esgScores` - Environmental, social, and governance scores
- `financialData` - Financial KPIs (revenue, margins, cash flow, etc.)
- `institutionOwnership` - Institutional ownership data
- `insiderHolders` - Insider holdings data
- `insiderTransactions` - Insider transaction data
- and more...

Source

Yahoo Finance API

Examples

```
# View all possible modules
data(valid_modules)
head(valid_modules)

# Check if a specific module exists
"financialData" %in% valid_modules
```

Index

* datasets

- [valid_balance_keys](#), 24
- [valid_cashflow_keys](#), 25
- [valid_income_keys](#), 26
- [valid_modules](#), 26

- [get_balance_sheet](#), 2
- [get_cashflow](#), 4
- [get_financials](#), 5
- [get_history](#), 7
- [get_income_statement](#), 9
- [get_info](#), 10
- [get_tickers](#), 13
- [get_tickers_balance_sheet](#), 14
- [get_tickers_cashflow](#), 15
- [get_tickers_financials](#), 16
- [get_tickers_history](#), 17
- [get_tickers_income_statement](#), 19
- [get_tickers_info](#), 20
- [get_user_agent](#), 21

- [print.yf_ticker](#), 21
- [print.yf_tickers](#), 22

- [search_tickers](#), 22

- [valid_balance_keys](#), 24
- [valid_cashflow_keys](#), 25
- [valid_income_keys](#), 26
- [valid_modules](#), 26
- [validate_tickers](#), 24