

Package: yfscreen (via r-universe)

June 7, 2026

Type Package

Title Yahoo Finance 'screener' API

Version 0.1.3

Description Simple and efficient access to Yahoo Finance's 'screener' API <<https://finance.yahoo.com/research-hub/screener/>> for querying and retrieving financial data. The 'yfscreen' package abstracts the complexities of interacting with Yahoo Finance APIs, such as session management, crumb and cookie handling, query construction, pagination, and JSON payload generation. This abstraction allows users to focus on filtering and retrieving data rather than managing API details. Use cases include screening across a range of security types such as equities, mutual funds, ETFs, indices, and futures. The package supports advanced query capabilities, including logical operators, nested filters, and customizable payloads. It handles pagination automatically, fetching results in batches of up to 250 entries per request for efficient retrieval of large datasets. Filters can be defined dynamically to support a wide range of screening needs. The implementation uses standard HTTP libraries to handle API interactions efficiently and is available in both R and 'Python' for accessibility to a broad audience.

License GPL (>= 2)

URL <https://github.com/jasonjfoster/screen>

BugReports <https://github.com/jasonjfoster/screen/issues>

Depends R (>= 3.5.0)

Imports curl, jsonlite

Suggests covr, testthat

Config/roxygen2/old_usage TRUE

Config/roxygen2/version 8.0.0

Encoding UTF-8

LazyData true

Config/pak/sysreqs libssl-dev

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

Date/Publication 2026-06-06 23:22:46 UTC

RemoteUrl <https://github.com/jasonjfoster/screen>

RemoteRef HEAD

RemoteSha dafc45f0f856ca0ea88628711811f38666d6f6a

RemoteSubdir r

Contents

yfscreen-package	2
create_payload	3
create_query	4
data_categoryname	5
data_errors	5
data_exchange	6
data_filters	6
data_fundfamilyname	6
data_industry	7
data_peer_group	7
data_region	7
data_sector	8
get_data	8
get_session	9

Index **10**

yfscreen-package *Yahoo Finance 'screener' API*

Description

Simple and efficient access to Yahoo Finance's 'screener' API <<https://finance.yahoo.com/research-hub/screener/>> for querying and retrieving financial data. The 'yfscreen' package abstracts the complexities of interacting with Yahoo Finance APIs, such as session management, crumb and cookie handling, query construction, pagination, and JSON payload generation. This abstraction allows users to focus on filtering and retrieving data rather than managing API details. Use cases include screening across a range of security types such as equities, mutual funds, ETFs, indices, and futures. The package supports advanced query capabilities, including logical operators, nested filters, and customizable payloads. It handles pagination automatically, fetching results in batches of up to 250 entries per request for efficient retrieval of large datasets. Filters can be defined dynamically to support a wide range of screening needs. The implementation uses standard HTTP libraries to handle API interactions efficiently and is available in both R and 'Python' for accessibility to a broad audience.

Details

yfscreen is a package that provides simple and efficient access to Yahoo Finance's screener functionality for querying and retrieving financial data.

The core functionality of the yfscreen package abstracts the complexities of interacting with Yahoo Finance APIs, such as session management, crumb and cookie handling, query construction, pagination, and JSON payload generation. This abstraction allows users to focus on filtering and retrieving data rather than managing API details. Use cases include screening across a range of asset classes:

- **Equities:** coverage spans 50 regions to enable the identification of top-performing stocks based on specified criteria
- **Mutual funds:** funds can be screened using metrics such as historical performance, performance ratings, and other factors
- **ETFs:** a wide range of ETFs can be filtered by criteria including expense ratio, historical performance, and additional attributes
- **Indices:** stock market indices are available and often categorized by sector, industry, or the overall market
- **Futures:** futures contracts can be screened by exchange, price percent changes, and regional specifications

The implementation leverages standard HTTP libraries to handle API interactions efficiently and provides support for both R and Python to ensure accessibility for a broad audience.

Author(s)

Jason Foster [aut, cre]

create_payload

Create a Payload for the Yahoo Finance API

Description

A function to create a payload to query the Yahoo Finance API with customizable parameters.

Usage

```
create_payload(sec_type = "equity", query = NULL, size = 25,  
              offset = 0, sort_field = NULL, sort_type = NULL,  
              top_operator = "and")
```

Arguments

sec_type	string. Type of security to search (i.e., "equity", "mutualfund", "etf", "index", "future").
query	list. Structured query to filter results created by the <code>create_query</code> function.
size	integer. Number of results to return.
offset	integer. Starting position of the results.
sort_field	string. Field to sort the results.
sort_type	string. Type of sort to apply (i.e., "asc", "desc").
top_operator	string. Logical operator for the top-level of the query (i.e., "and", "or").

Value

A list representing the payload to be sent to the Yahoo Finance API with the specified parameters.

Examples

```
filters <- list(
  list("eq", list("region", "us")),
  list("btwn", list("intradaymarketcap", 2000000000, 10000000000)),
  list("btwn", list("intradaymarketcap", 10000000000, 100000000000)),
  list("gt", list("intradaymarketcap", 100000000000)),
  list("gt", list("dayvolume", 5000000))
)

query <- create_query(filters)

payload <- create_payload("equity", query)
```

create_query

Create a Structured Query for the Yahoo Finance API

Description

A function to create a structured query with logical operations and nested conditions formatted for the Yahoo Finance API.

Usage

```
create_query(filters = list("eq", list("region", "us")),
  top_operator = "and")
```

Arguments

filters	list. Each element is a sublist that defines a filtering condition with the following structure: comparison string. Comparison operator (i.e., "gt", "lt", "eq", "btwn"). field list. Field name (e.g., "region") and its associated value(s).
top_operator	string. Top-level logical operator to combine all filters (i.e., "and", "or").

Value

A nested list representing the structured query with logical operations and nested conditions formatted for the Yahoo Finance API.

Examples

```
filters <- list(
  list("eq", list("region", "us")),
  list("btwn", list("intradaymarketcap", 2000000000, 10000000000)),
  list("btwn", list("intradaymarketcap", 10000000000, 100000000000)),
  list("gt", list("intradaymarketcap", 100000000000)),
  list("gt", list("dayvolume", 5000000))
)

query <- create_query(filters)
```

data_categoryname	<i>Category Name Data for the Yahoo Finance API</i>
-------------------	---

Description

A data frame with the available category name data for the Yahoo Finance API.

Usage

```
data_categoryname
```

Format

A data frame.

data_errors	<i>Errors Data for the Yahoo Finance API</i>
-------------	--

Description

A data frame with the available errors data for the Yahoo Finance API.

Usage

```
data_errors
```

Format

A data frame.

data_exchange	<i>Exchange Data for the Yahoo Finance API</i>
---------------	--

Description

A data frame with the available exchange data for the Yahoo Finance API.

Usage

data_exchange

Format

A data frame.

data_filters	<i>Filters Data for the Yahoo Finance API</i>
--------------	---

Description

A data frame with the available filters data for the Yahoo Finance API.

Usage

data_filters

Format

A data frame.

data_fundfamilyname	<i>Fund Family Name Data for the Yahoo Finance API</i>
---------------------	--

Description

A data frame with the available fund family name data for the Yahoo Finance API.

Usage

data_fundfamilyname

Format

A data frame.

data_industry	<i>Industry Data for the Yahoo Finance API</i>
---------------	--

Description

A data frame with the available industry data for the Yahoo Finance API.

Usage

```
data_industry
```

Format

A data frame.

data_peer_group	<i>Peer Group Data for the Yahoo Finance API</i>
-----------------	--

Description

A data frame with the available peer group data for the Yahoo Finance API.

Usage

```
data_peer_group
```

Format

A data frame.

data_region	<i>Region Data for the Yahoo Finance API</i>
-------------	--

Description

A data frame with the available region data for the Yahoo Finance API.

Usage

```
data_region
```

Format

A data frame.

data_sector	<i>Sector Data for the Yahoo Finance API</i>
-------------	--

Description

A data frame with the available sector data for the Yahoo Finance API.

Usage

```
data_sector
```

Format

A data frame.

get_data	<i>Get Data from the Yahoo Finance API</i>
----------	--

Description

A function to get data from the Yahoo Finance API using the specified payload.

Usage

```
get_data(payload = NULL)
```

Arguments

payload list. Payload that contains search criteria created using the [create_query](#) and [create_payload](#) functions.

Value

A data frame that contains data from the Yahoo Finance API for the specified search criteria.

Examples

```
filters <- list(
  list("eq", list("region", "us")),
  list("btwn", list("intradaymarketcap", 2000000000, 10000000000)),
  list("btwn", list("intradaymarketcap", 10000000000, 100000000000)),
  list("gt", list("intradaymarketcap", 100000000000)),
  list("gt", list("dayvolume", 5000000))
)

query <- create_query(filters)
```

```
payload <- create_payload("equity", query)

## Not run:
data <- get_data(payload)

## End(Not run)
```

get_session

Get the Crumb, Cookies, and Handle for Yahoo Finance API

Description

A function to get the crumb, cookies, and handle required to authenticate and interact with the Yahoo Finance API.

Usage

```
get_session()
```

Value

A list containing the following elements:

handle	A curl handle object for subsequent requests.
crumb	A string representing the crumb value for authentication.
cookies	A data frame of cookies for the request.

Examples

```
session <- get_session()
```

Index

* datasets

- data_categoryname, [5](#)
- data_errors, [5](#)
- data_exchange, [6](#)
- data_filters, [6](#)
- data_fundfamilyname, [6](#)
- data_industry, [7](#)
- data_peer_group, [7](#)
- data_region, [7](#)
- data_sector, [8](#)

create_payload, [3](#), [8](#)

create_query, [4](#), [4](#), [8](#)

- data_categoryname, [5](#)
- data_errors, [5](#)
- data_exchange, [6](#)
- data_filters, [6](#)
- data_fundfamilyname, [6](#)
- data_industry, [7](#)
- data_peer_group, [7](#)
- data_region, [7](#)
- data_sector, [8](#)

get_data, [8](#)

get_session, [9](#)

yfscreen (yfscreen-package), [2](#)

yfscreen-package, [2](#)