

# Package: RobinHood (via r-universe)

November 4, 2024

**Type** Package

**Title** Interface for the RobinHood.com No Commission Investing Platform

**Version** 1.8.0

**Author** Joseph Blubaugh

**Maintainer** Joseph Blubaugh <jestonblu@gmail.com>

**Description** Execute API calls to the RobinHood <<https://robinhood.com>> investing platform. Functionality includes accessing account data and current holdings, retrieving investment statistics and quotes, placing and canceling orders, getting market trading hours, searching investments by popular tag, and interacting with watch lists.

**Imports** jsonlite, lubridate, profvis, magrittr, dplyr, httr, uuid, methods

**License** GPL-3

**URL** <https://github.com/JestonBlu/RobinHood>,  
<https://jestonblu.github.io/RobinHood/>

**RoxygenNote** 7.2.3

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

**RemoteUrl** <https://github.com/jestonblu/robinhood>

**RemoteRef** HEAD

**RemoteSha** 303c79c7c29c756ac08bafa691e7999ed8c08aed

## Contents

api_accounts	3
api_accounts_crypto	3
api_ach	4
api_contracts	4
api_currency_pairs	5
api_endpoints	5
api_fundamentals	6

api_historicals . . . . .	6
api_historicals_crypto . . . . .	7
api_historicals_options . . . . .	7
api_instruments . . . . .	8
api_instruments_options . . . . .	8
api_login . . . . .	9
api_logout . . . . .	9
api_marketdata . . . . .	10
api_markets . . . . .	10
api_orders . . . . .	11
api_orders_crypto . . . . .	12
api_orders_options . . . . .	13
api_portfolios . . . . .	14
api_portfolios_crypto . . . . .	14
api_positions . . . . .	14
api_positions_crypto . . . . .	15
api_positions_options . . . . .	15
api_quote . . . . .	16
api_quote_crypto . . . . .	16
api_ratings . . . . .	17
api_tag . . . . .	17
api_tickers . . . . .	17
api_user . . . . .	18
api_watchlist . . . . .	18
cancel_ach_transfer . . . . .	19
cancel_order . . . . .	19
cancel_order_crypto . . . . .	20
cancel_order_options . . . . .	21
check_rh . . . . .	21
get_accounts . . . . .	22
get_ach . . . . .	22
get_contracts . . . . .	23
get_fundamentals . . . . .	23
get_historicals . . . . .	24
get_historicals_crypto . . . . .	25
get_historicals_options . . . . .	26
get_market_hours . . . . .	27
get_order_history . . . . .	27
get_order_history_crypto . . . . .	28
get_order_history_options . . . . .	28
get_order_status . . . . .	29
get_order_status_crypto . . . . .	30
get_order_status_options . . . . .	31
get_portfolios . . . . .	31
get_positions . . . . .	32
get_positions_crypto . . . . .	33
get_positions_options . . . . .	33
get_quote . . . . .	34

- get\_quote\_crypto . . . . . 34
- get\_ratings . . . . . 35
- get\_tag . . . . . 36
- get\_tickers . . . . . 36
- get\_user . . . . . 37
- logout . . . . . 37
- mod\_json . . . . . 38
- place\_ach\_transfer . . . . . 38
- place\_order . . . . . 39
- place\_order\_crypto . . . . . 40
- place\_order\_options . . . . . 41
- RobinHood . . . . . 42
- watchlist . . . . . 43

**Index** **44**

api\_accounts *RobinHood API: User*

**Description**

Backend function called by get\_user(), returns a list of user account data.

**Usage**

api\_accounts(RH)

**Arguments**

RH                    object of class RobinHood

api\_accounts\_crypto *RobinHood API: User Crypto*

**Description**

Backend function which calls the Nummus API to return the account id.

**Usage**

api\_accounts\_crypto(RH)

**Arguments**

RH                    object of class RobinHood

---

 api\_ach

*RobinHood API: ACH*


---

### Description

Backend function for interacting and getting data on linked bank accounts.

### Usage

```
api_ach(
  RH,
  action,
  amount = NULL,
  status_url = NULL,
  cancel_url = NULL,
  transfer_url = NULL
)
```

### Arguments

RH	object of class RobinHood
action	(string) one of "transfers", "relationships", "schedules", "status", "cancel", "deposit", "withdraw"
amount	(numeric) amount in dollars you want to deposit or withdraw (NULL if not one of those actions)
status_url	(string) URL returned by place_ach_transfer()
cancel_url	(string) URL returned by place_ach_transfer()
transfer_url	(string) url of your linked account, output of get_ach(RH, "relationships")

---

 api\_contracts

*RobinHood API: Option Contract Instruments*


---

### Description

RobinHood API: Option Contract Instruments

### Usage

```
api_contracts(RH, chain_symbol, type)
```

### Arguments

RH	object of class RobinHood
chain_symbol	(string) a single ticker symbol
type	(string) one of call or put

---

api\_currency\_pairs      *RobinHood API: Currency Pairs*

---

**Description**

Returns the ID for a particular paired currency

**Usage**

```
api_currency_pairs(RH)
```

**Arguments**

RH                      object of class RobinHood

---

api\_endpoints              *RobinHood API: Endpoints*

---

**Description**

Backend function called by almost every function. Returns the appropriate starting URL for a given endpoint.

**Usage**

```
api_endpoints(endpoint, source = "equity")
```

**Arguments**

endpoint                (string) which api endpoint to look up?  
source                   (string) directs api to api.robinhood.com (equity) vs nummus.robinhood.com (crypto)

---

api\_fundamentals      *RobinHood API: Fundamentals*

---

### Description

Backend function called by `get_fundamentals()`, `watchlist()`. Returns a data frame of descriptive data for a given ticker symbol.

### Usage

```
api_fundamentals(RH, ticker)
```

### Arguments

RH                    object of class RobinHood  
ticker                (string) vector of ticker symbols

---

api\_historical      *RobinHood API: Historicals*

---

### Description

Backend function called by `get_historical()`. Returns a data frame of historical price data.

### Usage

```
api_historical(RH, historicals_url, body)
```

### Arguments

RH                    object of class RobinHood  
historicals\_url      (string) api url  
body                  (string) api body

---

`api_historical_crypto`*RobinHood API: Historicals crypto*

---

**Description**

Backend function called by `get_historical_crypto()`. Returns a data frame of historical price data.

**Usage**

```
api_historical_crypto(RH, url)
```

**Arguments**

RH	object of class RobinHood
url	(string) full url coming from <code>get_historical_crypto</code>

---

`api_historical_options`*RobinHood API: Historicals Options*

---

**Description**

Backend function called by `get_historical_options()`.

**Usage**

```
api_historical_options(  
  RH,  
  chain_symbol,  
  type,  
  strike_price,  
  expiration_date,  
  interval = NULL,  
  span = NULL  
)
```

**Arguments**

RH	object of class RobinHood
chain_symbol	(string) stock symbol
type	(string) one of ("put", "call")
strike_price	(numeric) strike price

expiration_date	(string) expiration date (YYYY-MM-DD)
interval	(string) one of ("5minute", "10minute", "hour", "day", "week")
span	(string) one of ("day", "week", "month")

---

api_instruments	<i>RobinHood API: Instruments</i>
-----------------	-----------------------------------

---

### Description

Backend function called by get\_tag(), get\_position(), watchlist(). Returns a list of instrument data.

### Usage

```
api_instruments(RH, symbol = NULL, instrument_url = NULL)
```

### Arguments

RH	object of class RobinHood
symbol	(string) a single symbol
instrument_url	(string) instrument url

---

api_instruments_options	<i>RobinHood API: Option Contracts</i>
-------------------------	--

---

### Description

Backend function for retrieving option contracts.

### Usage

```
api_instruments_options(
  RH,
  method = "url",
  option_instrument_url = NULL,
  chain_symbol = NULL,
  type = NULL,
  state = NULL,
  strike_price = NULL,
  tradability = NULL,
  expiration_date = NULL
)
```



**Arguments**

RH	object of class RobinHood
method	(string) one of ("url", "symbol")
option_instrument_url	(string) direct url for an option contract
chain_symbol	(string) stock symbol
type	(string) one of ("put", "call")
state	(string) one of ("active", "inactive")
strike_price	(numeric) strike price
tradability	(string) one of ("tradable", "untradable")
expiration_date	(string) expiration date ("YYYY-MM-DD")

---

 api\_login

*RobinHood API: Login*


---

**Description**

Backend function called by RobinHood(). Returns a list like object of class RobinHood which stores tokens required by all other functions.

**Usage**

```
api_login(username, password, mfa_code)
```

**Arguments**

username	(string) RobinHood username
password	(string) RobinHood password
mfa_code	(string) Provided by your authentication app

---

 api\_logout

*RobinHood API: Logout*


---

**Description**

Backend function called by logout(). Sends a logout call and disables your oauth2 token.

**Usage**

```
api_logout(RH)
```

**Arguments**

RH	object of class RobinHood
----	---------------------------

---

api_marketdata	<i>RobinHood API: Market Data</i>
----------------	-----------------------------------

---

**Description**

Returns a dataframe of quantitative market information for a particular option instrument

**Usage**

```
api_marketdata(RH, instrument, type = "instrument_id")
```

**Arguments**

RH	object of class RobinHood
instrument	(string) a single instrument_id or multiple instrument_urls
type	(string) one of instrument_id or instrument_url

---

api_markets	<i>RobinHood API: Markets</i>
-------------	-------------------------------

---

**Description**

Backend function called by get\_market\_hours(). Returns a data frame of markets data and trading hours.

**Usage**

```
api_markets(RH, markets_url, type = "df")
```

**Arguments**

RH	object of class RobinHood
markets_url	(string) a single market url
type	(string) structure of data returned, 'df' or 'list'

**Description**

Backend function called by `place_order()`, `get_order_status()`, `cancel_order()`. Issues a buy/sell order or returns the status of an order. When issuing a buy order use the `url` column in the return data to check the status or cancel the order.

**Usage**

```
api_orders(
    RH,
    action,
    status_url = NULL,
    cancel_url = NULL,
    instrument_id = NULL,
    symbol = NULL,
    type = NULL,
    time_in_force = NULL,
    trigger = NULL,
    price = NULL,
    stop_price = NULL,
    quantity = NULL,
    side = NULL,
    page_size = NULL
)
```

**Arguments**

<code>RH</code>	object of class <code>RobinHood</code>
<code>action</code>	(string) one of "order", "status", "cancel", or "history"
<code>status_url</code>	(string) if <code>action = "status"</code> , <code>status_url</code> is required (output from <code>place_order()</code> )
<code>cancel_url</code>	(string) if <code>action = "cancel"</code> , <code>cancel_url</code> is required (output from <code>place_order()</code> )
<code>instrument_id</code>	(string) URL of the <code>instrument_id</code>
<code>symbol</code>	(string) Ticket symbol you are attempting to buy or sell
<code>type</code>	(string) "market" or "limit"
<code>time_in_force</code>	(string) Good For Day ("gfd"), Good Till Canceled ("gtc"), Immediate or Cancel ("ioc"), or Opening ("opg")
<code>trigger</code>	(string) "immediate" or "stop"
<code>price</code>	(number) the price you are willing to sell or buy at
<code>stop_price</code>	(number) if <code>trigger = stop</code> , enter stop price, otherwise leave blank
<code>quantity</code>	(int) number of shares you wish to transact
<code>side</code>	(string) "buy" or "sell"
<code>page_size</code>	(int) for <code>get_order_history</code> , refers to the number of historical records to return

---

api\_orders\_crypto      *RobinHood API: Orders Crypto*

---

### Description

Backend function called by `place_order()`, `get_order_status()`, `cancel_order()`. Issues a buy/sell order or returns the status of an order. When issuing a buy order use the `url` column in the return data to check the status or cancel the order.

### Usage

```
api_orders_crypto(
    RH,
    action,
    order_id = NULL,
    cancel_url = NULL,
    currency_pair_id = NULL,
    type = NULL,
    time_in_force = NULL,
    price = NULL,
    quantity = NULL,
    side = NULL
)
```

### Arguments

<code>RH</code>	object of class <code>RobinHood</code>
<code>action</code>	(string) one of "order", "status", "cancel", or "history"
<code>order_id</code>	(string) action is "status" or "cancel", only <code>order_id</code> is required
<code>cancel_url</code>	(string) url for posting a cancel order
<code>currency_pair_id</code>	(string) currency pair id
<code>type</code>	(string) "market" or "limit"
<code>time_in_force</code>	(string) Good For Day ("gfd"), Good Till Canceled ("gtc"), Immediate or Cancel ("ioc"), or Opening ("opg")
<code>price</code>	(number) the price you are willing to sell or buy at
<code>quantity</code>	(int) number of shares you wish to transact
<code>side</code>	(string) "buy" or "sell"

---

api\_orders\_options      *RobinHood API: Option Orders*

---

## Description

RobinHood API: Option Orders

## Usage

```
api_orders_options(  
    RH,  
    action,  
    status_url = NULL,  
    cancel_url = NULL,  
    quantity = NULL,  
    direction = NULL,  
    stop_price = NULL,  
    type = NULL,  
    time_in_force = NULL,  
    side = NULL,  
    option_id = NULL  
)
```

## Arguments

RH	object of class RobinHood
action	(string) one of "order", "status", "cancel", "history"
status_url	(string) if action = "status", status_url is required (output from place_order_options())
cancel_url	(string) if action = "cancel", cancel_url is required (output from place_order_options())
quantity	(int) number of contracts you want to buy
direction	(string) one of "debit" or "credit"
stop_price	(numeric) stop price on a limit order
type	(string) one of "limit" or "market"
time_in_force	(string) Good Till Canceled ("gtc"), Immediate or Cancel ("ioc"), or Opening ("opg")
side	(string) one of "buy" or "sell"
option_id	(string) id column returned by get_contracts()

---

api\_portfolios      *RobinHood API: Portfolio*

---

**Description**

Backend function called by `get_portfolio()`. Returns a data frame of account summaries.

**Usage**

```
api_portfolios(RH, portfolio_url)
```

**Arguments**

RH                    object of class RobinHood  
portfolio\_url      portfolio url

---

api\_portfolios\_crypto      *RobinHood API: Portfolio Crypto*

---

**Description**

Backend function called by `get_portfolio(..., source = "crypto")`. Returns a data frame of the current crypto portfolio summary.

**Usage**

```
api_portfolios_crypto(RH)
```

**Arguments**

RH                    object of class RobinHood

---

api\_positions              *RobinHood API: Positions*

---

**Description**

Backend function called by `get_positions()`. Returns a data frame of instrument position data.

**Usage**

```
api_positions(RH)
```

**Arguments**

RH                    object of class RobinHood

---

api\_positions\_crypto *RobinHood API: Cryptocurrency Positions*

---

**Description**

Backend function called by get\_positions(). Returns a data frame of crypto position data via the Nummus api.

**Usage**

```
api_positions_crypto(RH)
```

**Arguments**

RH                    object of class RobinHood

---

api\_positions\_options *RobinHood API: Option Contract Positions*

---

**Description**

Backend function called by get\_positions\_options(). Returns a data frame of owned options contracts.

**Usage**

```
api_positions_options(RH)
```

**Arguments**

RH                    object of class RobinHood

---

api_quote	<i>RobinHood API: Quote</i>
-----------	-----------------------------

---

**Description**

Backend function called by `get_positions()`, `get_quote()`, `place_order()`. Returns a data frame of quote data

**Usage**

```
api_quote(RH, symbols_url)
```

**Arguments**

RH	object of class RobinHood
symbols_url	(string) url of query with ticker symbols

---

api_quote_crypto	<i>RobinHood API: Quotes for cryptocurrency</i>
------------------	---

---

**Description**

Returns a dataframe of current quotes

**Usage**

```
api_quote_crypto(RH, symbols_url)
```

**Arguments**

RH	object of class RobinHood
symbols_url	(string) url of query with ticker symbols



---

api_ratings	<i>RobinHood API: Analyst ratings</i>
-------------	---------------------------------------

---

**Description**

Backend function called by `get_ratings()` to get analyst rating from RobinHood

**Usage**

```
api_ratings(RH, symbol)
```

**Arguments**

RH	object of class RobinHood
symbol	(string)

---

api_tag	<i>RobinHood API: Tag</i>
---------	---------------------------

---

**Description**

Backend function called by `get_tag()`. Returns a list of instrument ids.

**Usage**

```
api_tag(RH, tag)
```

**Arguments**

RH	object of class RobinHood
tag	(string) a hyphenated tag such as "100-most-popular"

---

api_tickers	<i>RobinHood API: Tickers</i>
-------------	-------------------------------

---

**Description**

Backend function called by `get_ticker`. Returns a data frame of all instruments listed on RobinHood.

**Usage**

```
api_tickers(RH)
```

**Arguments**

RH	object of class RobinHood
----	---------------------------

---

api_user	<i>RobinHood API: User</i>
----------	----------------------------

---

**Description**

Backend function called by get\_user() to return user data

**Usage**

api\_user(RH)

**Arguments**

RH	object of class RobinHood
----	---------------------------

---

api_watchlist	<i>RobinHood API: watchlist</i>
---------------	---------------------------------

---

**Description**

Backend function called by watchlist(). Adds or remove instruments from the default watchlist. The create and delete watchlist features are disabled as it appears that the functionality is not currently available on the platform.

**Usage**

api\_watchlist(RH, watchlist\_url, detail = FALSE, delete = FALSE)

**Arguments**

RH	object of class RobinHood
watchlist_url	(string) a single watchlist url
detail	(logical) if null use header api only, otherwise pass options
delete	(logical) send delete call

---

cancel\_ach\_transfer     *Cancel an ACH transfer from your RobinHood account*

---

**Description**

Cancel an ACH transfer from your RobinHood account

**Usage**

```
cancel_ach_transfer(RH, cancel_url)
```

**Arguments**

RH                    object of class RobinHood  
cancel\_url            (string) cancel url returned from initiating a transfer with place\_ach\_transfer()

---

cancel\_order             *Cancel an existing order on RobinHood*

---

**Description**

Send a cancel signal for a particular order to RobinHood. You will need to retain the buy/sell order url returned from place\_order.

**Usage**

```
cancel_order(RH, cancel_url)
```

**Arguments**

RH                    object of class RobinHood  
cancel\_url            (string) cancel url returned from place\_order()

**Examples**

```
## Not run:
# Login in to your RobinHood account
RH <- RobinHood("username", "password")

# Place an order, should generate an email confirmation
x <- place_order(RH = RH,
                 symbol = "GE",             # Ticker symbol you want to trade
                 type = "market",         # Type of market order
                 time_in_force = "gfd",   # Time period the order is good for (gfd: good for day)
                 trigger = "immediate",   # Trigger or delay order
                 price = 8.96,             # The highest price you are willing to pay
```

```

        quantity = 1,          # Number of shares you want
        side = "buy")         # buy or sell

# Cancel the order, should also generate an email confirmation
cancel_order(RH, x$cancel_url)

## End(Not run)

```

---

cancel\_order\_crypto    *Cancel an existing crypto order on RobinHood*

---

## Description

Send a cancel signal for a particular order to RobinHood. You will need to retain the buy/sell order url returned from place\_order.

## Usage

```
cancel_order_crypto(RH, cancel_url)
```

## Arguments

RH                    object of class RobinHood  
cancel\_url            (string) cancel url returned from place\_order\_crypto()

## Examples

```

## Not run:
# Login in to your RobinHood account
RH <- RobinHood("username", "password")

# Place an order, should generate an email confirmation
x <- place_order_crypto(RH = RH,
                        symbol = "ETC",          # Ticker symbol
                        type = "market",         # Type of market order (market, limit)
                        time_in_force = "gtc",   # Time period (gfd: good for day)
                        price = 1,              # The highest price you are willing to pay
                        quantity = 1,           # Number of shares you want
                        side = "buy")           # buy or sell

# Cancel the order, should also generate an email confirmation
cancel_order_crypto(RH, x$cancel_url)

## End(Not run)

```

---

cancel\_order\_options    *Cancel an existing options order on RobinHood*

---

### Description

Send a cancel signal for a particular order to RobinHood. You will need to retain the buy/sell order url returned from place\_order\_options().

### Usage

```
cancel_order_options(RH, cancel_url)
```

### Arguments

RH                    object of class RobinHood  
cancel\_url            (string) cancel url returned from place\_order()

### Examples

```
## Not run:
# Login in to your RobinHood account
RH <- RobinHood("username", "password")

# Place an order, should generate an email confirmation
x <- place_order_options(RH = RH,
  option_id = "346e46af-380e-4052-a7c2-15748f0fc0ca",
  direction = "debit",    # one of "debit" or "credit"
  side = "buy",            # one of "buy" or "sell"
  quantity = 1,            # number of contracts
  stop_price = .01,        # Time period (gfd: good for day)
  type = "limit",         # limit or market (only limit is currently supported)
  time_in_force = "gtc") # "gfd", "gtc", "ioc", "opg"

# Cancel the order, should also generate an email confirmation
cancel_order_options(RH, x$cancel_url)

## End(Not run)
```

---

check\_rh                    *Checks to see if you have a valid RobihHood object*

---

### Description

Checks to see if you have a valid RobihHood object

### Usage

```
check_rh(RH)
```

**Arguments**

RH                    object of class RobinHood

---

get\_accounts                    *Get data related to your RobinHood account*

---

**Description**

Get data related to your RobinHood account

**Usage**

```
get_accounts(RH)
```

**Arguments**

RH                    object of class RobinHood

**Examples**

```
## Not run:
# Login in to your RobinHood account
RH <- RobinHood("username", "password")

get_accounts(RH)

## End(Not run)
```

---

get\_ach                    *Get ACH data from your RobinHood linked bank accounts*

---

**Description**

Get ACH data from your RobinHood linked bank accounts

**Usage**

```
get_ach(RH, action, status_url = NULL)
```

**Arguments**

RH                    object of class RobinHood  
action                (string) one of "transfers", "relationships", "schedules", "status"  
status\_url            (string) URL returned by place\_ach\_transfer()

---

get_contracts	<i>Get a option contracts from RobinHood</i>
---------------	--

---

**Description**

Get a option contracts from RobinHood

**Usage**

```
get_contracts(RH, chain_symbol, type, detail = FALSE)
```

**Arguments**

RH	object class RobinHood
chain_symbol	(string) a single ticket symbol
type	(string) one of call or put
detail	(logical) if TRUE (default) return additional info on greeks, previous day, high/low fill rate prices

**Examples**

```
## Not run:  
# Login in to your RobinHood account  
RH <- RobinHood("username", "password")  
  
get_contracts(RH, "IR")  
  
## End(Not run)
```

---

get_fundamentals	<i>Get fundamental investment statistics for a particular ticker symbol</i>
------------------	---

---

**Description**

Get fundamental investment statistics for a particular ticker symbol

**Usage**

```
get_fundamentals(RH, ticker, include_description = FALSE)
```

**Arguments**

RH	object of class RobinHood
ticker	(string) vector of ticker symbols
include_description	(logical) include a long description of the company (default: FALSE)

**Examples**

```
## Not run:  
# Login in to your RobinHood account  
RH <- RobinHood("username", "password")  
  
get_fundamentals(RH, "CAT")  
  
## End(Not run)
```

---

get_historical	<i>Get historical price history from RobinHood</i>
----------------	--

---

**Description**

Returns a data frame of historical price history for a given symbol.

**Usage**

```
get_historical(RH, symbol, interval, span, tz = Sys.timezone())
```

**Arguments**

RH	object of class RobinHood
symbol	(string) Stock symbol to query, single symbol only
interval	(string) Interval of time to aggregate to (examples: hour, day, week, month)
span	(string) Period of time you are interested in (examples: day, week, month, year)
tz	(string) timezone returned by OlsonNames() (eg: "America/Chicago")

**Examples**

```
## Not run:  
# Login in to your RobinHood account  
RH <- RobinHood("username", "password")  
  
get_historical(RH = RH, symbol = "CAT", interval = "day", span = "month")  
  
## End(Not run)
```



---

`get_historical_crypto`*Get historical crypto price history from RobinHood*

---

## Description

Returns a data frame of historical crypto price history for a given symbol. Note that not all combinations of interval/span and bounds will return results. Spans that go beyond day (week, month, year) will not return results for certain bounds values (trading, extended). The function should return a message if you specify a combination of inputs that returns no results.

## Usage

```
get_historical_crypto(RH, symbol, interval, span, bounds, tz = Sys.timezone())
```

## Arguments

<code>RH</code>	object of class <code>RobinHood</code>
<code>symbol</code>	(string) ticker symbol of crypto (BTC, ETH, ETC)
<code>interval</code>	(string) Interval of time to aggregate to (examples: hour, day, week, month)
<code>span</code>	(string) Period of time you are interested in (examples: day, week, month, year)
<code>bounds</code>	(string) One of regular (6 hours), trading (9 hours), extended (16 hours), 24_7
<code>tz</code>	(string) timezone returned by <code>OlsonNames()</code> (eg: "America/Chicago")

## Examples

```
## Not run:
# Login in to your RobinHood account
RH <- RobinHood("username", "password")

get_historical_crypto(RH, symbol = "ETC", interval = "5minute", span = "day", bounds = 'regular')
get_historical_crypto(RH, symbol = "ETC", interval = "5minute", span = "day", bounds = 'trading')
get_historical_crypto(RH, symbol = "ETC", interval = "5minute", span = "day", bounds = 'extended')
get_historical_crypto(RH, symbol = "ETC", interval = "5minute", span = "day", bounds = '24_7')

## End(Not run)
```

---

`get_historical_options`*Get historical options from RobinHood*

---

### Description

Returns a data frame of historical options for a given symbol, strike price, and expiration date.

### Usage

```
get_historical_options(  
  RH,  
  chain_symbol,  
  type,  
  strike_price,  
  expiration_date,  
  interval = NULL,  
  span = NULL  
)
```

### Arguments

<code>RH</code>	object of class <code>RobinHood</code>
<code>chain_symbol</code>	(string) stock symbol
<code>type</code>	(string) one of ("put", "call")
<code>strike_price</code>	(numeric) strike price
<code>expiration_date</code>	(string) expiration date (YYYY-MM-DD)
<code>interval</code>	(string) one of ("5minute", "10minute", "hour", "day", "week")
<code>span</code>	(string) one of ("day", "week", "month")

### Examples

```
## Not run:  
# Login in to your RobinHood account  
RH <- RobinHood("username", "password")  
  
get_historical_options(RH = RH, chain_symbol = "AAPL", interval = "10minute", type = "call",  
  expiration_date = "2021-03-12", strike_price = 122)  
  
## End(Not run)
```

---

get\_market\_hours      *Get open market hours*

---

**Description**

Get a list of markets available on RobinHood with trading hours for a specific date.

**Usage**

```
get_market_hours(RH, market_date = NULL, tz = Sys.timezone())
```

**Arguments**

RH	object of class RobinHood
market_date	(string) date in the form 'yyyy-mm-dd', default today
tz	(string) one of timezone returned by OlsonNames(), defaults to local

**Examples**

```
## Not run:  
# Login in to your RobinHood account  
RH <- RobinHood("username", "password")  
  
get_market_hours(RH)  
  
## End(Not run)
```

---

get\_order\_history      *Download all available order history for your RobinHood account*

---

**Description**

Download all available order history for your RobinHood account

**Usage**

```
get_order_history(RH, page_size = 1000)
```

**Arguments**

RH	object of class RobinHood
page_size	(int) number of historical records to fetch

**Examples**

```
## Not run:  
# Login in to your RobinHood account  
RH <- RobinHood("username", "password")  
  
get_order_history(RH)  
  
## End(Not run)
```

---

get\_order\_history\_crypto

*Download all available crypto currency order history for your RobinHood account*

---

**Description**

Download all available crypto currency order history for your RobinHood account

**Usage**

```
get_order_history_crypto(RH)
```

**Arguments**

RH                    object of class RobinHood

**Examples**

```
## Not run:  
# Login in to your RobinHood account  
RH <- RobinHood("username", "password")  
  
get_order_history_crypto(RH)  
  
## End(Not run)
```

---

get\_order\_history\_options

*Download all available options order history for your RobinHood account*

---

**Description**

Download all available options order history for your RobinHood account

**Usage**

```
get_order_history_options(RH)
```

**Arguments**

RH                    object of class RobinHood

**Examples**

```
## Not run:
# Login in to your RobinHood account
RH <- RobinHood("username", "password")

get_order_history(RH)

## End(Not run)
```

---

get_order_status	<i>Get a current status of an order on RobinHood</i>
------------------	--

---

**Description**

Returns a list of order information given a buy/sell order url returned from place\_order().

**Usage**

```
get_order_status(RH, status_url, limit_output = TRUE)
```

**Arguments**

RH                    object of class RobinHood  
status\_url            (string) url of order returned from place\_order  
limit\_output        (logical) return limited info on the order (default TRUE)

**Examples**

```
## Not run:
# Login in to your RobinHood account
RH <- RobinHood("username", "password")

# Place an order, should generate an email confirmation
x <- place_order(RH = RH,
                 symbol = "GE",            # Ticker symbol you want to trade
                 type = "market",        # Type of market order
                 time_in_force = "gfd", # Time period the order is good for (gfd: good for day)
                 trigger = "immediate", # Trigger or delay order
                 price = 8.96,            # The highest price you are willing to pay
                 quantity = 1,            # Number of shares you want
```

```

        side = "buy")           # buy or sell

get_order_status(RH, x$status_url)

## End(Not run)

```

---

```
get_order_status_crypto
```

*Get a current status of an crypto currency order on RobinHood*

---

## Description

Returns a list of order information given a buy/sell order url returned from place\_order().

## Usage

```
get_order_status_crypto(RH, order_id)
```

## Arguments

RH	object of class RobinHood
order_id	(string) id field of the object returned by place_order_crypto

## Examples

```

## Not run:
# Login in to your RobinHood account
RH <- RobinHood("username", "password")

# Place an order, should generate an email confirmation
x <- place_order_crypto(RH = RH,
  symbol = "GE",           # Ticker symbol you want to trade
  type = "market",        # Type of market order
  time_in_force = "gfd",  # Time period (gfd: good for day)
  trigger = "immediate",  # Trigger or delay order
  price = 8.96,           # The highest price you are willing to pay
  quantity = 1,          # Number of shares you want
  side = "buy")          # buy or sell

get_order_status_crypto(RH, order_id = x$id)

## End(Not run)

```

---

```
get_order_status_options
```

*Get a current status of an option order on RobinHood*

---

### Description

Returns a list of order information given a buy/sell order url returned from place\_order\_options().

### Usage

```
get_order_status_options(RH, status_url)
```

### Arguments

RH                    object of class RobinHood  
 status\_url            (string) url of order returned from place\_order

### Examples

```
## Not run:
# Login in to your RobinHood account
RH <- RobinHood("username", "password")

# Place an order, should generate an email confirmation
x <- place_order_options(RH = RH,
  option_id = "346e46af-380e-4052-a7c2-15748f0fc0ca",
  direction = "debit", # one of "debit" or "credit"
  side = "buy", # one of "buy" or "sell"
  quantity = 1, # number of contracts
  stop_price = .01, # Time period (gfd: good for day)
  type = "limit", # limit or market (only limit is currently supported)
  time_in_force = "gtc") # "gfd", "gtc", "ioc", "opg"

get_order_status_options(RH, x$status_url)

## End(Not run)
```

---

```
get_portfolios
```

*Get portfolio summaries related to your RobinHood Account*

---

### Description

Returns a dataframe of portfolio summaries for a specific period of time. Default is current day.

### Usage

```
get_portfolios(RH, interval = NULL, span = NULL)
```

**Arguments**

RH                    object of class RobinHood  
interval            (string) Interval of time to aggregate to (examples: hour, day, week, month)  
span                (string) Period of time you are interested in (examples: day, week, month, year)

**Examples**

```
## Not run:  
# Login in to your RobinHood account  
RH <- RobinHood("username", "password")  
  
get_portfolios(RH)  
get_portfolios(RH, interval = "day", span = "3month")  
  
## End(Not run)
```

---

get\_positions                    *Get the currently held positions for your RobinHood account*

---

**Description**

Get the currently held positions for your RobinHood account

**Usage**

```
get_positions(RH, limit_output = TRUE)
```

**Arguments**

RH                    object class RobinHood  
limit\_output        (logical) if true, return a simplified positions table, false returns all position details

**Examples**

```
## Not run:  
# Login in to your RobinHood account  
RH <- RobinHood("username", "password")  
  
get_positions(RH)  
  
## End(Not run)
```



---

get\_positions\_crypto *Get the currently held crypto positions for your RobinHood account*

---

**Description**

Get the currently held crypto positions for your RobinHood account

**Usage**

```
get_positions_crypto(RH)
```

**Arguments**

RH                    object class RobinHood

**Examples**

```
## Not run:  
# Login in to your RobinHood account  
RH <- RobinHood("username", "password")  
  
get_positions_crypto(RH)  
  
## End(Not run)
```

---

get\_positions\_options *Get the currently held positions for your RobinHood account*

---

**Description**

Get the currently held positions for your RobinHood account

**Usage**

```
get_positions_options(RH, trim_pending = TRUE)
```

**Arguments**

RH                    object class RobinHood  
trim\_pending        (logical) if FALSE, then return pending and intraday columns

**Examples**

```
## Not run:  
# Login in to your RobinHood account  
RH <- RobinHood("username", "password")  
  
get_positions_options(RH)  
  
## End(Not run)
```

---

get_quote	<i>Get a quote from RobinHood</i>
-----------	-----------------------------------

---

**Description**

Get a quote from RobinHood

**Usage**

```
get_quote(RH, symbol, limit_output = TRUE)
```

**Arguments**

RH	object class RobinHood
symbol	(string) of ticker symbols
limit_output	(logical) if TRUE (default) return less quote detail

**Examples**

```
## Not run:  
# Login in to your RobinHood account  
RH <- RobinHood("username", "password")  
  
get_quote(RH, "IR")  
  
## End(Not run)
```

---

get_quote_crypto	<i>Get a crypto currency quote from RobinHood</i>
------------------	---

---

**Description**

Get a crypto currency quote from RobinHood

**Usage**

```
get_quote_crypto(RH, symbol)
```

**Arguments**

RH                    object class RobinHood  
symbol                (string) crypto currency symbol such as BTC, ETH, DOGE

**Examples**

```
## Not run:  
# Login in to your RobinHood account  
RH <- RobinHood("username", "password")  
  
get_quote_crypto(RH, "BTC")  
  
## End(Not run)
```

---

get\_ratings                    *Get analyst ratings for a given equity symbol from RobinHood*

---

**Description**

Get analyst ratings for a given equity symbol from RobinHood

**Usage**

```
get_ratings(RH, symbol)
```

**Arguments**

RH                    object class RobinHood  
symbol                (string) Ticker symbol

**Examples**

```
## Not run:  
# Login in to your RobinHood account  
RH <- RobinHood("username", "password")  
  
get_ratings(RH, symbol = "CAT")  
  
## End(Not run)
```

---

get_tag	<i>Get ticker symbols for popular tags on RobinHood</i>
---------	---

---

**Description**

Get a data frame of ticker symbols and names linked to trending tags on the RobinHood website.

**Usage**

```
get_tag(RH, tag)
```

**Arguments**

RH	object class RobinHood
tag	(string) a hyphenated tag such as "100-most-popular"

**Examples**

```
## Not run:  
# Login in to your RobinHood account  
RH <- RobinHood("username", "password")  
  
get_tag(RH, "100-most-popular")  
  
## End(Not run)
```

---

get_tickers	<i>Download all financial instruments available on RobinHood</i>
-------------	--

---

**Description**

Get a dataframe of all listed companies including ticker symbols and tradeability indicators.

**Usage**

```
get_tickers(RH, add_fundamentals = FALSE)
```

**Arguments**

RH	object of class RobinHood
add_fundamentals	(logical) if TRUE then return fundamental data (long run time)

**Examples**

```
## Not run:  
# Login in to your RobinHood account  
RH <- RobinHood("username", "password")  
  
get_tickers(RH)  
  
## End(Not run)
```

---

`get_user`*Get personal user data related to your RobinHood account*

---

**Description**

Get personal user data related to your RobinHood account

**Usage**

```
get_user(RH)
```

**Arguments**

RH                    object class RobinHood

**Examples**

```
## Not run:  
# Login in to your RobinHood account  
RH <- RobinHood("username", "password")  
  
get_user(RH)  
  
## End(Not run)
```

---

`logout`*Logout of RobinHood*

---

**Description**

Send a logout call through the RobinHood API service and disable your token.

**Usage**

```
logout(RH)
```

**Arguments**

RH                    object of class RobinHood

**Examples**

```
## Not run:
# Login in to your RobinHood account
RH <- RobinHood("username", "password")

logout(RH)

## End(Not run)
```

---

mod_json	<i>Converts a data frame to json minus the square brackets on the ends</i>
----------	--

---

**Description**

Converts a data frame to json minus the square brackets on the ends

**Usage**

```
mod_json(x, type)
```

**Arguments**

x                    dataframe intended for json conversoin  
type                (string) one of "fromJSON" or "toJSON"

---

place_ach_transfer	<i>Place an ACH transfer to and from your RobinHood account</i>
--------------------	---

---

**Description**

Place an ACH transfer to and from your RobinHood account

**Usage**

```
place_ach_transfer(RH, action, amount, transfer_url)
```

**Arguments**

RH                    object of class RobinHood  
action                (string) one of "deposit", "withdraw"  
amount                (numeric) amount in dollars you want to deposit or withdraw  
transfer\_url        (string) url of your linked account, output of get\_ach(RH, "relationships")

---

place\_order

*Place a buy or sell order against your RobinHood account*


---

**Description**

Place a buy or sell order against your RobinHood account

**Usage**

```
place_order(
    RH,
    symbol,
    type,
    time_in_force,
    trigger,
    price = NA,
    stop_price = NA,
    quantity,
    side
)
```

**Arguments**

RH	object of class RobinHood
symbol	(string) Ticket symbol you are attempting to buy or sell
type	(string) "market" or "limit"
time_in_force	(string) Good For Day ("gfd"), Good Till Canceled ("gtc"), Immediate or Cancel ("ioc"), or Opening ("opg")
trigger	(string) "immediate" or "stop"
price	(number) the price you are willing to sell or buy at (max 2 decimals) (Note: $P * Q > 0.01$ )
stop_price	(number) if trigger = stop, enter stop price, otherwise leave blank
quantity	(int) number of shares you wish to transact ( $> 0$ ) (Note: $P * Q > 0.01$ )
side	(string) "buy" or "sell"

**Examples**

```
## Not run:
# ***** ATTENTION *****
# - Price cannot extend beyond 2 decimals
# - Price * Quantity > $0.01
# - Stop triggers requires stop_price > 0
# ***** ATTENTION *****

# Login in to your RobinHood account
```

```

RH <- RobinHood("username", "password")

# Place an order, should generate an email confirmation
place_order(RH = RH,
            symbol = "GE",          # Ticker symbol you want to trade
            type = "market",       # Type of market order (market, limit)
            time_in_force = "gfd", # Time period the order is good for (gfd: good for day)
            trigger = "immediate", # Trigger or delay order
            price = 8.96,          # The highest price you are willing to pay
            quantity = 1,         # Number of shares you want
            side = "buy")         # buy or sell

# Stop loss example
place_order(RH,
            symbol="ABC",
            type = 'market',
            trigger = 'stop',
            stop_price = 100,
            time_in_force = "gtc", # Good till close
            quantity = 10,
            side = 'sell')

## End(Not run)

```

---

place\_order\_crypto      *Place a crypto currency buy or sell order against your RobinHood account*

---

### Description

**\*\*Note\*\***: Price and Quantity can both extend beyond 2 decimals **\*\*Note\*\***: Price \* Quantity > \$0.01

### Usage

```
place_order_crypto(RH, symbol, type, time_in_force, price, quantity, side)
```

### Arguments

RH	object of class RobinHood
symbol	(string) Ticket symbol you are attempting to buy or sell
type	(string) "market" or "limit"
time_in_force	(string) Good Till Canceled ("gtc"), Immediate or Cancel ("ioc"), or Opening ("opg")
price	(number) the price you are willing to sell or buy at
quantity	(number) number of shares you wish to transact
side	(string) "buy" or "sell"



**Examples**

```
## Not run:
# Login in to your RobinHood account
RH <- RobinHood("username", "password")

# Place an order, should generate an email confirmation
x <- place_order_crypto(RH = RH,
  symbol = "DOGE",      # Ticker symbol
  type = "market",     # Type of market order (market, limit)
  time_in_force = "gtc", # Time period (gfd: good for day)
  price = .003,        # The highest price you are willing to pay
  quantity = 500,      # Number of shares you want
  side = "buy")        # buy or sell

## End(Not run)
```

---

place\_order\_options    *Place a options buy or sell order against your RobinHood account*

---

**Description**

Place an order on an option contract. Currently only limit orders are supported so you must supply a price

**Usage**

```
place_order_options(
  RH,
  option_id,
  direction,
  side,
  quantity,
  stop_price = NULL,
  type = "limit",
  time_in_force
)
```

**Arguments**

RH	object of class RobinHood
option_id	(string) id returned from get_contracts()
direction	(string) one of "debit" or "credit"
side	(string) one of "buy" or "sell"
quantity	(integer) number of contracts to buy
stop_price	(numeric) stop price for a limit order
type	(string) "limit" or "market" (only limit is currently supported)
time_in_force	(string) Good Till Canceled ("gtc"), Immediate or Cancel ("ioc"), or Opening ("opg")

**Examples**

```
## Not run:
# Login in to your RobinHood account
RH <- RobinHood("username", "password")

# Place an options order, should generate an email confirmation
x <- place_order_options(RH = RH,
  option_id = "346e46af-380e-4052-a7c2-15748f0fc0ca",
  direction = "debit", # one of "debit" or "credit"
  side = "buy", # one of "buy" or "sell"
  quantity = 1, # number of contracts
  stop_price = .01, # Time period (gfd: good for day)
  type = "limit", # limit or market (only limit is currently supported)
  time_in_force = "gtc") # "gfd", "gtc", "ioc", "opg"

## End(Not run)
```

---

RobinHood

*RobinHood Account Authentication*


---

**Description**

This function returns an object of S3 class RobinHood and establishes a connection to a RobinHood account. It is a required input for every other function in the package.

**Usage**

```
RobinHood(username, password, mfa_code)
```

**Arguments**

username	(string) account email address
password	(string) password
mfa_code	(string) mfa_code provided by your authentication app (required if mfa is enabled)

**Examples**

```
## Not run:
RH <- RobinHood("username", "password")

## End(Not run)
```

---

watchlist	<i>Manage your RobinHood watchlist</i>
-----------	--

---

**Description**

Add and delete instruments from your RobinHood watchlist.

**Usage**

```
watchlist(RH, action, watchlist = "", ticker = "")
```

**Arguments**

RH	object class RobinHood
action	(string) one of: get, add, delete
watchlist	(string) name of watchlist to add, delete, or get instruments, null will return a list of watchlist
ticker	(string) list of tickers to add or delete, null will add or delete watchlist

**Examples**

```
## Not run:  
# Login in to your RobinHood account  
RH <- RobinHood("username", "password")  
  
# Get a vector of your watchlist names  
watchlist(RH, action = "get")  
  
# Get a vectors of symbols on your watchlist  
watchlist(RH, action = "get", watchlist = "Default")  
  
# Add a symbol to your watchlist  
watchlist(RH, action = "add", watchlist = "Default", ticker = "CAT")  
  
# Delete a symbol from your watchlist  
watchlist(RH, action = "delete", watchlist = "Default", ticker = "CAT")  
  
## End(Not run)
```

# Index

api\_accounts, 3  
api\_accounts\_crypto, 3  
api\_ach, 4  
api\_contracts, 4  
api\_currency\_pairs, 5  
api\_endpoints, 5  
api\_fundamentals, 6  
api\_historical, 6  
api\_historical\_crypto, 7  
api\_historical\_options, 7  
api\_instruments, 8  
api\_instruments\_options, 8  
api\_login, 9  
api\_logout, 9  
api\_marketdata, 10  
api\_markets, 10  
api\_orders, 11  
api\_orders\_crypto, 12  
api\_orders\_options, 13  
api\_portfolios, 14  
api\_portfolios\_crypto, 14  
api\_positions, 14  
api\_positions\_crypto, 15  
api\_positions\_options, 15  
api\_quote, 16  
api\_quote\_crypto, 16  
api\_ratings, 17  
api\_tag, 17  
api\_tickers, 17  
api\_user, 18  
api\_watchlist, 18  
  
cancel\_ach\_transfer, 19  
cancel\_order, 19  
cancel\_order\_crypto, 20  
cancel\_order\_options, 21  
check\_rh, 21  
  
get\_accounts, 22  
get\_ach, 22  
get\_contracts, 23  
get\_fundamentals, 23  
get\_historical, 24  
get\_historical\_crypto, 25  
get\_historical\_options, 26  
get\_market\_hours, 27  
get\_order\_history, 27  
get\_order\_history\_crypto, 28  
get\_order\_history\_options, 28  
get\_order\_status, 29  
get\_order\_status\_crypto, 30  
get\_order\_status\_options, 31  
get\_portfolios, 31  
get\_positions, 32  
get\_positions\_crypto, 33  
get\_positions\_options, 33  
get\_quote, 34  
get\_quote\_crypto, 34  
get\_ratings, 35  
get\_tag, 36  
get\_tickers, 36  
get\_user, 37  
  
logout, 37  
  
mod\_json, 38  
  
place\_ach\_transfer, 38  
place\_order, 39  
place\_order\_crypto, 40  
place\_order\_options, 41  
  
RobinHood, 42  
  
watchlist, 43