Package ‘RobinHood’

April 22, 2022

Type Package

Title Interface for the RobinHood.com No Commission Investing Platform

Version 1.6.5

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

License GPL-3

URL https://github.com/JestonBlu/RobinHood

RoxygenNote 7.1.2

NeedsCompilation no

Repository CRAN

Date/Publication 2022-04-22 03:10:02 UTC

R topics documented:

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
### Topics Documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>api_login</code></td>
<td>9</td>
</tr>
<tr>
<td><code>api_logout</code></td>
<td>9</td>
</tr>
<tr>
<td><code>api_marketdata</code></td>
<td>10</td>
</tr>
<tr>
<td><code>api_markets</code></td>
<td>10</td>
</tr>
<tr>
<td><code>api_orders</code></td>
<td>11</td>
</tr>
<tr>
<td><code>api_orders_crypto</code></td>
<td>12</td>
</tr>
<tr>
<td><code>api_orders_options</code></td>
<td>13</td>
</tr>
<tr>
<td><code>api_portfolios</code></td>
<td>14</td>
</tr>
<tr>
<td><code>api_portfolios_crypto</code></td>
<td>14</td>
</tr>
<tr>
<td><code>api_positions</code></td>
<td>14</td>
</tr>
<tr>
<td><code>api_positions_crypto</code></td>
<td>15</td>
</tr>
<tr>
<td><code>api_positions_options</code></td>
<td>15</td>
</tr>
<tr>
<td><code>api_quote</code></td>
<td>16</td>
</tr>
<tr>
<td><code>api_quote_crypto</code></td>
<td>16</td>
</tr>
<tr>
<td><code>api_ratings</code></td>
<td>17</td>
</tr>
<tr>
<td><code>api_tag</code></td>
<td>17</td>
</tr>
<tr>
<td><code>api_tickers</code></td>
<td>17</td>
</tr>
<tr>
<td><code>api_user</code></td>
<td>18</td>
</tr>
<tr>
<td><code>api_watchlist</code></td>
<td>18</td>
</tr>
<tr>
<td><code>cancel_ach_transfer</code></td>
<td>19</td>
</tr>
<tr>
<td><code>cancel_order</code></td>
<td>19</td>
</tr>
<tr>
<td><code>cancel_order_crypto</code></td>
<td>20</td>
</tr>
<tr>
<td><code>cancel_order_options</code></td>
<td>21</td>
</tr>
<tr>
<td><code>check_rh</code></td>
<td>21</td>
</tr>
<tr>
<td><code>get_accounts</code></td>
<td>22</td>
</tr>
<tr>
<td><code>get_ach</code></td>
<td>22</td>
</tr>
<tr>
<td><code>get_contracts</code></td>
<td>23</td>
</tr>
<tr>
<td><code>get_fundamentals</code></td>
<td>23</td>
</tr>
<tr>
<td><code>get_historicals</code></td>
<td>24</td>
</tr>
<tr>
<td><code>get_historicals_crypto</code></td>
<td>25</td>
</tr>
<tr>
<td><code>get_historicals_options</code></td>
<td>26</td>
</tr>
<tr>
<td><code>get_market_hours</code></td>
<td>27</td>
</tr>
<tr>
<td><code>get_order_history</code></td>
<td>27</td>
</tr>
<tr>
<td><code>get_order_history_crypto</code></td>
<td>28</td>
</tr>
<tr>
<td><code>get_order_history_options</code></td>
<td>28</td>
</tr>
<tr>
<td><code>get_order_status</code></td>
<td>29</td>
</tr>
<tr>
<td><code>get_order_status_crypto</code></td>
<td>30</td>
</tr>
<tr>
<td><code>get_order_status_options</code></td>
<td>31</td>
</tr>
<tr>
<td><code>get_portfolios</code></td>
<td>31</td>
</tr>
<tr>
<td><code>get_positions</code></td>
<td>32</td>
</tr>
<tr>
<td><code>get_positions_crypto</code></td>
<td>33</td>
</tr>
<tr>
<td><code>get_positions_options</code></td>
<td>33</td>
</tr>
<tr>
<td><code>get_quote</code></td>
<td>34</td>
</tr>
<tr>
<td><code>get_quote_crypto</code></td>
<td>34</td>
</tr>
<tr>
<td><code>get_ratings</code></td>
<td>35</td>
</tr>
<tr>
<td><code>get_tag</code></td>
<td>36</td>
</tr>
<tr>
<td><code>get_tickers</code></td>
<td>36</td>
</tr>
<tr>
<td><code>get_user</code></td>
<td>37</td>
</tr>
</tbody>
</table>
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_contracts

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**

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

**Arguments**

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

**api_historicals**  
*RobinHood API:Historicals*

**Description**

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

**Usage**

```r
api_historicals(RH, historicals_url, body)
```

**Arguments**

- **RH**  
  object of class RobinHood
- **historicals_url**  
  (string) api url
- **body**  
  (string) api body
**api_historicals_crypto**

*Robin Hood API: Historicals crypto*

**Description**

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

**Usage**

```r
api_historicals_crypto(RH, url)
```

**Arguments**

- **RH**: object of class RobinHood
- **url**: (string) full url coming from get_historicals_crypto

---

**api_historicals_options**

*Robin Hood API: Historicals Options*

**Description**

Backend function called by get_historicals_options().

**Usage**

```r
api_historicals_options(
    RH,       # object of class RobinHood
    chain_symbol,  # (string) stock symbol
    type,  # (string) one of ("put", "call")
    strike_price,  # (numeric) strike price
    expiration_date,  # (string) date
    interval = NULL,  # (numeric) interval
    span = NULL  # (numeric) span
)
```

**Arguments**

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

**RobinHood API: Instruments**

#### Description

Backend function called by get_tag(), get_position(), watchlist(). Returns a list of instrument data.

#### Usage

```r
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

**RobinHood API: Option Contracts**

#### Description

Backend function or retrieving option contracts.

#### Usage

```r
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 
)
```
**api_login**

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RH</td>
<td>object of class RobinHood</td>
</tr>
<tr>
<td>method</td>
<td>(string) one of (&quot;url&quot;, &quot;symbol&quot;)</td>
</tr>
<tr>
<td>option_instrument_url</td>
<td>(string) direct url for an option contract</td>
</tr>
<tr>
<td>chain_symbol</td>
<td>(string) stock symbol</td>
</tr>
<tr>
<td>type</td>
<td>(string) one of (&quot;put&quot;, &quot;call&quot;)</td>
</tr>
<tr>
<td>state</td>
<td>(string) one of (&quot;active&quot;, &quot;inactive&quot;)</td>
</tr>
<tr>
<td>strike_price</td>
<td>(numeric) strike price</td>
</tr>
<tr>
<td>tradability</td>
<td>(string) one of (&quot;tradable&quot;, &quot;untradable&quot;)</td>
</tr>
<tr>
<td>expiration_date</td>
<td>(string) expiration date (&quot;YYYY-MM-DD&quot;)</td>
</tr>
</tbody>
</table>

**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**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>(string) RobinHood username</td>
</tr>
<tr>
<td>password</td>
<td>(string) RobinHood password</td>
</tr>
<tr>
<td>mfa_code</td>
<td>(string) Provided by your authentication app</td>
</tr>
</tbody>
</table>

**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**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RH</td>
<td>object of class RobinHood</td>
</tr>
</tbody>
</table>
**api_marketdata**  
*RobinHood API: Market Data*

**Description**

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

**Usage**

```r
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**  
*RobinHood API: Markets*

**Description**

Backend function called by get_market_hours(). Returns a data frame of markets data and trading hours.

**Usage**

```r
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'
**api_orders**

**RobinHood API: Orders**

**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**

```r
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**

- **RH** object of class RobinHood
- **action** (string) one of "order", "status", "cancel", or "history"
- **status_url** (string) if action = "status", status_url is required (output from place_order())
- **cancel_url** (string) if action = "cancel", cancel_url is required (output from place_order())
- **instrument_id** (string) URL of the instrument_id
- **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
- **stop_price** (number) if trigger = stop, enter stop price, otherwise leave blank
- **quantity** (int) number of shares you wish to transact
- **side** (string) "buy" or "sell"
- **page_size** (int) for get_order_history, refers to the number of historical records to return
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

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

RobinHood API: Option Orders

Usage

```r
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) 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 |
Description

Backend function called by `get_positions()`. Returns a data frame of crypto position data via the Nummus api.

Usage

```r
api_positions_crypto(RH)
```

Arguments

- **RH**: object of class RobinHood

Description

Backend function called by `get_positions_options()`. Returns a data frame of owned options contracts.

Usage

```r
api_positions_options(RH)
```

Arguments

- **RH**: object of class RobinHood
api_quote

RobinHood API: Quote

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

RobinHood API: Quotes for cryptocurrency

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

RobinHood API: Analyst ratings

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

RobinHood API: Tag

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

RobinHood API: Tickers

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**  
*RobinHood API: User*

**Description**
Backend function called by `get_user()` to return user data.

**Usage**
```r
api_user(RH)
```

**Arguments**
- `RH` object of class RobinHood

**api_watchlist**  
*RobinHood API: watchlist*

**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**
```r
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

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

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
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 RobinHood object

Description

Checks to see if you have a valid RobinHood object

Usage

check_rh(RH)
**get_accounts**  
Get data related to your RobinHood account

**Usage**

```r
get_accounts(RH)
```

**Arguments**

- `RH`  
  object of class RobinHood

**Examples**

```r
## 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**

```r
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

Get a option contracts from RobinHood

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

Get fundamental investment statistics for a particular ticker symbol

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

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

get_fundamentals(RH, "CAT")

## End(Not run)
```

---

### get_historicals

*Get historical price history from RobinHood*

**Description**

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

**Usage**

```r
get_historicals(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**

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

get_historicals(RH = RH, symbol = "CAT", interval = "day", span = "month")

## End(Not run)
```
get_historicals_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_historicals_crypto(RH, symbol, interval, span, bounds, tz = Sys.timezone())

Arguments

RH object of class RobinHood
symbol (string) ticker symbol of crypto (BTC, ETH, ETC)
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)
bounds (string) One of regular (6 hours), trading (9 hours), extended (16 hours), 24_7
tz (string) timezone returned by OlsonNames() (eg: "America/Chicago")

Examples

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

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

## End(Not run)
get_historicals_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_historicals_options(
  RH,
  chain_symbol,
  type,
  strike_price,
  expiration_date,
  interval = NULL,
  span = NULL
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RH</td>
<td>object of class RobinHood</td>
</tr>
<tr>
<td>chain_symbol</td>
<td>(string) stock symbol</td>
</tr>
<tr>
<td>type</td>
<td>(string) one of (&quot;put&quot;, &quot;call&quot;)</td>
</tr>
<tr>
<td>strike_price</td>
<td>(numeric) strike price</td>
</tr>
<tr>
<td>expiration_date</td>
<td>(string) expiration date (YYYY-MM-DD)</td>
</tr>
<tr>
<td>interval</td>
<td>(string) one of (&quot;5minute&quot;, &quot;10minute&quot;, &quot;hour&quot;, &quot;day&quot;, &quot;week&quot;)</td>
</tr>
<tr>
<td>span</td>
<td>(string) one of (&quot;day&quot;, &quot;week&quot;, &quot;month&quot;)</td>
</tr>
</tbody>
</table>

Examples

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

get_historicals_options(RH = RH, chain_symbol = "AAPL", interval = "10minute", type = "call",
                       expiration_date = "2021-03-12", strike_price = 122)

## End(Not run)
get_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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RH</td>
<td>object of class RobinHood</td>
</tr>
<tr>
<td>market_date</td>
<td>(string) date in the form 'yyyy-mm-dd', default today</td>
</tr>
<tr>
<td>tz</td>
<td>(string) one of timezone returned by OlsonNames(), defaults to local</td>
</tr>
</tbody>
</table>

Examples

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

get_market_hours(RH)

## End(Not run)
```

get_order_history

Description

Download all available order history for your RobinHood account

Usage

get_order_history(RH, page_size = 1000)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RH</td>
<td>object of class RobinHood</td>
</tr>
<tr>
<td>page_size</td>
<td>(int) number of historical records to fetch</td>
</tr>
</tbody>
</table>
## 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

```r
get_order_history_options(RH)
```

### Arguments

- `RH` object of class RobinHood

### Examples

```r
## 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

```r
get_order_history_crypto(RH)
```

### Arguments

- `RH` object of class RobinHood

### Examples

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

get_order_history_crypto(RH)

## End(Not run)
```
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 Get a current status of an order on RobinHood

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
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**

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

**Arguments**

- **RH**: object of class RobinHood
- **status_url**: (string) url of order returned from place_order

**Examples**

```r
## 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**

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

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RH</td>
<td>object of class RobinHood</td>
</tr>
<tr>
<td>interval</td>
<td>(string) Interval of time to aggregate to (examples: hour, day, week, month)</td>
</tr>
<tr>
<td>span</td>
<td>(string) Period of time you are interested in (examples: day, week, month, year)</td>
</tr>
</tbody>
</table>

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_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
get_quote_crypto

Get a quote from RobinHood

Description
Get a quote from RobinHood

Usage
get_quote_crypto(RH, symbol)

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

Get a quote from RobinHood

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_ratings*

**Arguments**

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

**Examples**

```r
## 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**

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

**Arguments**

- **RH**: object class RobinHood
- **symbol**: (string) Ticker symbol

**Examples**

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

get_ratings(RH, symbol = "CAT")
## End(Not run)
```
get_tag

Get ticker symbols for popular tags on RobinHood

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

Download all financial instruments available on RobinHood

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)
get_user

Examples

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

get_tickers(RH)

## End(Not run)
```

Description

Get personal user data related to your RobinHood account

Usage

```r
get_user(RH)
```

Arguments

- `RH`: object class RobinHood

Examples

```r
## 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

```r
logout(RH)
```
place_ach_transfer

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

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

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 conversion
   type (string) one of "fromJSON" or "toJSON"

place_ach_transfer

Place an ACH transfer to and from your RobinHood account

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

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

Examples

## Not run:

#使ってみましょう
# - Price cannot extend beyond 2 decimals
# - Quantity must be a whole number
# - 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 beyone 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"
place_order_options

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 TillCanceled ("gte"), Immediate or Cancel ("ioe"), or Opening ("opg")
Examples

```r
## 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 = NULL)
```

**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**

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

## End(Not run)
```
**watchlist**

Manage your RobinHood watchlist

**Description**

Add and delete instruments from your RobinHood watchlist.

**Usage**

```r
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**

```r
## 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_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