Package ‘RobinHood’

October 6, 2019

Type Package

Title Interface for the RobinHood.com No Commission Investing Platform

Version 1.2.1

Date 2019-10-5

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 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-10-06 13:20:03 UTC

R topics documented:

api_accounts ......................................................... 3
api_accounts_crypto .................................................. 3
api_currency_pairs .................................................. 3
api_endpoints ......................................................... 4
api_fundamentals ..................................................... 4
api_historicals ....................................................... 5
api_instruments ....................................................... 5
api_login .............................................................. 6
api_logout .............................................................. 6
api_markets ............................................................. 7
api_option_instruments ............................................... 7
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>api_option_positions</td>
<td>8</td>
</tr>
<tr>
<td>api_orders</td>
<td>8</td>
</tr>
<tr>
<td>api_orders_crypto</td>
<td>9</td>
</tr>
<tr>
<td>api_portfolios</td>
<td>10</td>
</tr>
<tr>
<td>api_portfolios_crypto</td>
<td>10</td>
</tr>
<tr>
<td>api_positions</td>
<td>10</td>
</tr>
<tr>
<td>api_positions_crypto</td>
<td>11</td>
</tr>
<tr>
<td>api_quote</td>
<td>11</td>
</tr>
<tr>
<td>api_quote_crypto</td>
<td>12</td>
</tr>
<tr>
<td>api_tag</td>
<td>12</td>
</tr>
<tr>
<td>api_tickers</td>
<td>12</td>
</tr>
<tr>
<td>api_user</td>
<td>13</td>
</tr>
<tr>
<td>api_watchlist</td>
<td>13</td>
</tr>
<tr>
<td>cancel_order</td>
<td>14</td>
</tr>
<tr>
<td>cancel_order_crypto</td>
<td>14</td>
</tr>
<tr>
<td>check_rh</td>
<td>15</td>
</tr>
<tr>
<td>get_accounts</td>
<td>16</td>
</tr>
<tr>
<td>get_fundamentals</td>
<td>16</td>
</tr>
<tr>
<td>get_historicals</td>
<td>17</td>
</tr>
<tr>
<td>get_market_hours</td>
<td>18</td>
</tr>
<tr>
<td>get_order_history</td>
<td>18</td>
</tr>
<tr>
<td>get_order_history_crypto</td>
<td>19</td>
</tr>
<tr>
<td>get_order_status</td>
<td>19</td>
</tr>
<tr>
<td>get_order_status_crypto</td>
<td>20</td>
</tr>
<tr>
<td>get_portfolios</td>
<td>21</td>
</tr>
<tr>
<td>get_positions</td>
<td>22</td>
</tr>
<tr>
<td>get_positions_crypto</td>
<td>22</td>
</tr>
<tr>
<td>get_quote</td>
<td>23</td>
</tr>
<tr>
<td>get_quote_crypto</td>
<td>23</td>
</tr>
<tr>
<td>get_tag</td>
<td>24</td>
</tr>
<tr>
<td>get_tickers</td>
<td>25</td>
</tr>
<tr>
<td>get_user</td>
<td>25</td>
</tr>
<tr>
<td>logout</td>
<td>26</td>
</tr>
<tr>
<td>mod_json</td>
<td>26</td>
</tr>
<tr>
<td>place_order</td>
<td>27</td>
</tr>
<tr>
<td>place_order_crypto</td>
<td>28</td>
</tr>
<tr>
<td>RobinHood</td>
<td>29</td>
</tr>
<tr>
<td>watchlist</td>
<td>29</td>
</tr>
</tbody>
</table>
### api_accounts

**RobinHood API: User**

**Description**

Backend function called by `get_user()` returns a list of user account data.

**Usage**

```r
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**

```r
api_accounts_crypto(RH)
```

**Arguments**

- `RH` object of class RobinHood

### api_currency_pairs

**RobinHood API: Currency Pairs**

**Description**

Returns the ID for a particular paired currency

**Usage**

```r
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

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

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

---

**api_instruments**  
*RobinHood API: Instruments*

**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_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)

**Arguments**

<table>
<thead>
<tr>
<th>username</th>
<th>(string) RobinHood username</th>
</tr>
</thead>
<tbody>
<tr>
<td>password</td>
<td>(string) RobinHood password</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**

| RH           | object of class RobinHood |
**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_option_instruments**  
*RobinHood API: Option Contract Instruments*

**Description**

RobinHood API: Option Contract Instruments

**Usage**

```r
api_option_instruments(RH, option_instrument_url)
```

**Arguments**

- `RH`  
  object of class RobinHood
- `option_instrument_url`  
  (string) url
api_option_positions  RobinHood API: Option Contract Positions

Description
Backend function called by get_contract_positions(). Returns a data frame of owned options contracts.

Usage
api_option_positions(RH)

Arguments
RH  object of class RobinHood

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
api_orders(RH, action, order_url = NULL, instrument_id = NULL, symbol = NULL, type = NULL, time_in_force = NULL, trigger = NULL, price = NULL, stop_price = NULL, quantity = NULL, side = NULL)

Arguments
RH  object of class RobinHood
action  (string) one of "order", "status", "cancel", or "history"
order_url  (string) action is "status" or "cancel", only order_url is required
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 ("gte"), 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"

---

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

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

```r
api_positions_crypto(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**

```r
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**

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

**Arguments**

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

---

**api_tag**  
*RobinHood API: Tag*

**Description**

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

**Usage**

```r
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**

```r
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_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, order_url)

Arguments
RH | object of class RobinHood
order_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$ccancel)

## 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 = "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_crypto(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)

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_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)
get_historicals

Examples

## 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

get_historicals(RH, symbol, interval, span)

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)

Examples

## 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_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)

Arguments
RH object of class RobinHood
market_date (string) date in the form 'yyyy-mm-dd', default today

Examples
## 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)

Arguments
RH object of class RobinHood
get_order_history_crypto

Examples

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

get_order_history(RH)
## End(Not run)
```

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)
```

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

```r
get_order_status(RH, order_url, limit_output = TRUE)
```
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)

describe

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**
```r
## 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
get_quote

Examples

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

get_positions_crypto(RH)

## 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_quote_crypto

Get a crypto currency quote from RobinHood

Description

Get a crypto currency quote from RobinHood

Usage

get_quote_crypto(RH, symbol)
get_tag

Arguments

<table>
<thead>
<tr>
<th>RH</th>
<th>object class RobinHood</th>
</tr>
</thead>
<tbody>
<tr>
<td>symbol</td>
<td>(string) crypto currency symbol such as BTC, ETH, DOGE</td>
</tr>
</tbody>
</table>

Examples

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

get_quote_crypto(RH, "BTC")

## 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

<table>
<thead>
<tr>
<th>RH</th>
<th>object class RobinHood</th>
</tr>
</thead>
<tbody>
<tr>
<td>tag</td>
<td>(string) a hyphenated tag such as &quot;100-most-popular&quot;</td>
</tr>
</tbody>
</table>

Examples

```r
## 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)

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

```r
# Login in to your RobinHood account
RH <- RobinHood("username", "password")

get_user(RH)
```

## Description

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

## Usage

```r
logout(RH)
```

## Arguments

- **RH**: object of class RobinHood

## Examples

```r
# Login in to your RobinHood account
RH <- RobinHood("username", "password")

logout(RH)
```

## mod_json

### Description

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

### Usage

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

### Arguments

- **x**: dataframe intended for json conversion
- **type**: (string) one of "fromJSON" or "toJSON"
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,
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
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"

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 (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

## End(Not run)
place_order_crypto

Place a crypto currency buy or sell order against your RobinHood account

Description

Place a crypto currency buy or sell order against your RobinHood account

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 For Day ("gfd"), Good Till Canceled ("gtc"), Immediate or Cancel ("ioc"), or Opening ("opg")
price       (number) the price you are willing to sell or buy at
quantity    (int) 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 = "gfd", # 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)
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)

Arguments
username
user name or email
password
password

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

watchlist
Manage your RobinHood watchlist

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

```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_currency_pairs, 3
api_endpoints, 4
api_fundamentals, 4
api_historicals, 5
api_instruments, 5
api_login, 6
api_logout, 6
api_markets, 7
api_option_instruments, 7
api_option_positions, 8
api_orders, 8
api_orders_crypto, 9
api_portfolios, 10
api_portfolios_crypto, 10
api_positions, 10
api_positions_crypto, 11
api_quote, 11
api_quote_crypto, 12
api_tag, 12
api_tickers, 12
api_user, 13
api_watchlist, 13

cancel_order, 14
cancel_order_crypto, 14
check_rh, 15

get_accounts, 16
get_fundamentals, 16
get_historicals, 17
get_market_hours, 18
get_order_history, 18
get_order_history_crypto, 19
get_order_status, 19
get_order_status_crypto, 20
get_portfolios, 21
get_positions, 22
get_positions_crypto, 22