Package 'etrader'

November 26, 2020

Title 'ETRADE' API Interface for R

Version 0.1.2

Author Tony Trevisan [aut, cre]

Maintainer Tony Trevisan <anthonytrevisan@gmail.com>

URL https://tonytrevisan.github.io/etrader/

BugReports https://github.com/tonytrevisan/etrader/issues

Description Use R to interface with the 'ETRADE' API <https://developer.etrade.com/home>.
  Functions include authentication, trading, quote requests, account information, and option
  chains. A user will need an ETRADE brokerage account and 'ETRADE' API ap-
  proval. See README
  for authentication process and examples.

License GPL-3

Encoding UTF-8

LazyData true

Imports magrittr, httr, urltools, dplyr, rvest, purrr, jsonlite

RoxygenNote 7.1.1

Suggests testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2020-11-26 16:20:02 UTC

R topics documented:

---

etrd_account_balance   *ETrade Account Balances*

Description

This API returns detailed balance information for a specified account for the current user. The information returned includes account type, option level, and details on up to four balances - account balance, margin account balance, day trade balance, and cash account balance.

Usage

```
etrd_account_balance(
  account,
  realTimeNAV = "false",
  output = c("df", "list"),
  access_tokens = NULL,
  etrade_cred = NULL,
  sandbox = FALSE
)
```

Arguments

account	The account ID key, not the account ID associated with the specific account. Use `etrd_account_list` to identify the proper account id key.
realTimeNAV	Default is false. If true, fetches real time balance
output	Indicate whether the output should be in the form of a data frame ('df') or list ('list'). Data frame is returned by default.
access_tokens	Access tokens are created using `etrd_auth_access_token`. This entry is not required because the output is saved and retrieved from R options automatically.

etrade_cred	The output created from [etrd_auth_credentials](#) when a valid ETRADE key and secret have been passed. This entry is not required because the output is saved and retrieved from R options automatically.
sandbox	ETRADE offers a sandbox environment for validating API calls and responses. If using the sandbox environment, this must be set to TRUE in each function called throughout etrader. ETRADE states "Sandbox responses use stored data that's intended to provide typical responses for basic use cases. So the responses you receive will not contain current data, and may not exactly match your requests in other ways." Essentially, the responses will not match the requests entered but successful pull will indicate whether the entry was valid or not.

Value

df or list of account balance information

Examples

```
## Not run:

account_id = etrd_account_list()$accountIdKey[1]
etrd_account_balance(account_id)


## End(Not run)
```

---

etrd_account_list *ETrade Account Data*

Description

This API returns the account information for the current user. The information returned includes account type, mode, and details. The Account ID Key is also given which is required for other etrader functions.

Usage

```
etrd_account_list(
  output = c("df", "list"),
  access_tokens = NULL,
  etrade_cred = NULL,
  sandbox = FALSE
)
```

Arguments

output	Indicate whether the output should be in the form of a data frame ('df') or list ('list'). Data frame is returned by default.

access_tokens	Access tokens are created using [etrd_auth_access_token](#). This entry is not required because the output is saved and retrieved from R options automatically.
etrade_cred	The output created from [etrd_auth_credentials](#) when a valid ETRADE key and secret have been passed. This entry is not required because the output is saved and retrieved from R options automatically.
sandbox	ETRADE offers a sandbox environment for validating API calls and responses. If using the sandbox environment, this must be set to TRUE in each function called throughout etrader. ETRADE states "Sandbox responses use stored data that's intended to provide typical responses for basic use cases. So the responses you receive will not contain current data, and may not exactly match your requests in other ways." Essentially, the responses will not match the requests entered but successful pull will indicate whether the entry was valid or not.

Value

a data frame or list of account data

Examples

```
## Not run:

# Once valid access tokens are obtained, the function can be run without parameters
etrd_account_list()


## End(Not run)
```

---

etrd_account_portfolio

ETrade Account Portfolios

Description

Provides detailed portfolio information for a selected brokerage account

Usage

```
etrd_account_portfolio(
  account,
  count = 50,
  marketSession = "REGULAR",
  totalsRequired = "false",
  lotsRequired = "false",
  view = "QUICK",
  access_tokens = NULL,
  etrade_cred = NULL,
  sandbox = FALSE
)
```

Arguments

account	The account ID key, not the account ID associated with the specific account. Use `etrd_account_list` to identify the proper account id key.
count	The number of positions to return in the response. If not specified, defaults to 50. 50 is also the maximum.
marketSession	The market session. Default: REGULAR. Options: REGULAR, EXTENDED
totalsRequired	It gives the total values of the portfolio. Default: false.
lotsRequired	It gives position lots for positions. Default: false.
view	The view query. Default: Quick. Options: PERFORMANCE, FUNDAMENTAL, OPTIONSWATCH, QUICK, COMPLETE
access_tokens	Access tokens are created using `etrd_auth_access_token`. This entry is not required because the output is saved and retrieved from R options automatically.
etrade_cred	The output created from `etrd_auth_credentials` when a valid ETRADE key and secret have been passed. This entry is not required because the output is saved and retrieved from R options automatically.
sandbox	ETRADE offers a sandbox environment for validating API calls and responses. If using the sandbox environment, this must be set to TRUE in each function called throughout etrader. ETRADE states "Sandbox responses use stored data that's intended to provide typical responses for basic use cases. So the responses you receive will not contain current data, and may not exactly match your requests in other ways." Essentially, the responses will not match the requests entered but successful pull will indicate whether the entry was valid or not.

Value

a list of portfolio objects based on request

Examples

```
## Not run:

account_id = etrd_account_list()$accountIdKey[1]
etrd_account_portfolio(account_id)


## End(Not run)
```

---

etrd_alerts	*ETrade Alert Search*

Description

Search alerts associated with the provided ETrade Account

Usage

```
etrd_alerts(
  account,
  count = 25,
  category = NULL,
  status = NULL,
  search = NULL,
  output = c("df", "list"),
  access_tokens = NULL,
  etrade_cred = NULL,
  sandbox = FALSE
)
```

Arguments

account	The account ID key, not the account ID associated with the specific account. Use `etrd_account_list` to identify the proper account id key.
count	The alert count. By default it returns 25. Max values that can be returned: 300.
category	The alert category. By default it will return STOCK and ACCOUNT. Options: STOCK, ACCOUNT
status	The alert status. By default it will return READ and UNREAD. Options: READ, UNREAD, DELETED
search	The alert search. Search is done based on the subject.
output	Indicate whether the output should be in the form of a data frame ('df') or list ('list'). Data frame is returned by default.
access_tokens	Access tokens are created using `etrd_auth_access_token`. This entry is not required because the output is saved and retrieved from R options automatically.
etrade_cred	The output created from `etrd_auth_credentials` when a valid ETRADE key and secret have been passed. This entry is not required because the output is saved and retrieved from R options automatically.
sandbox	ETRADE offers a sandbox environment for validating API calls and responses. If using the sandbox environment, this must be set to TRUE in each function called throughout etrader. ETRADE states "Sandbox responses use stored data that's intended to provide typical responses for basic use cases. So the responses you receive will not contain current data, and may not exactly match your requests in other ways." Essentially, the responses will not match the requests entered but successful pull will indicate whether the entry was valid or not.

Value

a list or data frame of alert data

Examples

```
## Not run:
```

```
account_id = etrd_account_list()$accountIdKey[1]
etrd_alerts(account_id)


## End(Not run)
```

___

etrd_alert_detail   *ETrade Alert Details*

___

Description

Get the details for a specific alert based on the alert ID

Usage

```
etrd_alert_detail(
  alertId,
  access_tokens = NULL,
  etrade_cred = NULL,
  sandbox = FALSE
)
```

Arguments

alertId	An alert id that comes from etrd_alerts
access_tokens	Access tokens are created using etrd_auth_access_token. This entry is not required because the output is saved and retrieved from R options automatically.
etrade_cred	The output created from etrd_auth_credentials when a valid ETRADE key and secret have been passed. This entry is not required because the output is saved and retrieved from R options automatically.
sandbox	ETRADE offers a sandbox environment for validating API calls and responses. If using the sandbox environment, this must be set to TRUE in each function called throughout etrader. ETRADE states "Sandbox responses use stored data that's intended to provide typical responses for basic use cases. So the responses you receive will not contain current data, and may not exactly match your requests in other ways." Essentially, the responses will not match the requests entered but successful pull will indicate whether the entry was valid or not.

Value

the alert details in list form

Examples

```
## Not run:

# Get an alert detail for a specific account and alert
account_id = etrd_account_list()$accountIdKey[1]
alert_id = etrd_alerts(account_id)$id[1]
etrd_alert_detail(alert_id)


## End(Not run)
```

---

etrd_auth_access_token

Auth Step 3: Create Access Tokens to use etrader

Description

Step 3 of ETRADE authentication. Pass the verification code generated after a successful log into the URL created from `etrd_auth_login_url`.

Usage

```
etrd_auth_access_token(verif_code, etrade_cred = NULL, sandbox = FALSE)
```

Arguments

verif_code	a 5 digit alpha numeric code created after successfully logging into the ETRADE URL generated from `etrd_auth_login_url`
etrade_cred	The output created from `etrd_auth_credentials` when a valid ETRADE key and secret have been passed. This entry is not required because the output is saved and retrieved from R options automatically.
sandbox	ETRADE offers a sandbox environment for validating API calls and responses. If using the sandbox environment, this must be set to TRUE in each function called throughout etrader. ETRADE states "Sandbox responses use stored data that's intended to provide typical responses for basic use cases. So the responses you receive will not contain current data, and may not exactly match your requests in other ways." Essentially, the responses will not match the requests entered but successful pull will indicate whether the entry was valid or not.

Details

Authentication into the ETRADE environment requires a three step process: set ETRADE credentials, generate request tokens to create a login URL, and use the verification code to create access tokens. The verification code is generated after successfully logging into the URL that comes from `etrd_auth_login_url`.

The access tokens expire each day at midnight Eastern Time. The next day Auth step 1-3 will be required to get new access tokens. If two hours have elapsed between API calls during a trading

day, the access token is inactivated. Use `etrd_auth_manage_token` to refresh an inactivated access token.

Value

a key and secret oauth access token

See Also

`etrd_auth_credentials` to set the key and secret into an oauth app, `etrd_auth_login_url` to generate request tokens and create a login URL, `etrd_auth_access_token` to use the verification code to create access tokens, `etrd_auth_manage_token` to renew or revoke access tokens

Examples

```
## Not run:

# The etrade_cred has been stored into options so does not need to be passed
# After a successful log in, paste the verification code as shown
access_tok = etrd_auth_access_token(verif_code = 'XX123')


## End(Not run)
```

---

etrd_auth_credentials   *Auth Step 1: Set ETRADE Key and Secret Credentials*

Description

Step 1 of ETRADE authentication. Create an oauth app using the key and secret that can be obtained after completing the agreements on ETRADE Developer.

Usage

```
etrd_auth_credentials(etrade_key, etrade_secret, sandbox = FALSE)
```

Arguments

`etrade_key`	Either a sandbox or production key provided by ETRADE after completing the agreements on ETRADE Developer
`etrade_secret`	Either a sandbox or production secret provided by ETRADE after completing the agreements on ETRADE Developer
`sandbox`	ETRADE offers a sandbox environment for validating API calls and responses. If using the sandbox environment, this must be set to TRUE in each function called throughout etrader. ETRADE states "Sandbox responses use stored data that's intended to provide typical responses for basic use cases. So the responses you receive will not contain current data, and may not exactly match your requests in other ways." Essentially, the responses will not match the requests entered but successful pull will indicate whether the entry was valid or not.

Details

Authentication into the ETRADE environment requires a three step process: set ETRADE credentials, generate request tokens to create a login URL, and use the verification code to create access tokens. The Key and Secret need to be fed into this function to set the credentials into an oauth app that can be fed into the next function: `etrd_auth_login_url`.

Value

an oauth app stored into options by default

See Also

`etrd_auth_credentials` to set the key and secret into an oauth app, `etrd_auth_login_url` to generate request tokens and create a login URL, `etrd_auth_access_token` to use the verification code to create access tokens, `etrd_auth_manage_token` to renew or revoke access tokens

Examples

```
## Not run:

# Pass the key and secret from ETRADE
etrade_cred = etrd_auth_credentials(etrade_key = 'PRODUCTION_ALPHANUMERIC_OAUTH_KEY',
                                    etrade_secret = 'PRODUCTION_ALPHANUMERIC_OAUTH_SECRET')


## End(Not run)
```

---

etrd_auth_login_url   *Auth Step 2: Generate ETRADE URL for log in*

Description

Step 2 of ETRADE authentication. Use the output from `etrd_auth_credentials` to generate Request Tokens and a login URL

Usage

```
etrd_auth_login_url(etrade_cred = NULL, sandbox = FALSE, auto_open = TRUE)
```

Arguments

etrade_cred   The output created from `etrd_auth_credentials` when a valid ETRADE key and secret have been passed. This entry is not required because the output is saved and retrieved from R options automatically.

sandbox   ETRADE offers a sandbox environment for validating API calls and responses. If using the sandbox environment, this must be set to TRUE in each function called throughout etrader. ETRADE states "Sandbox responses use stored data that's intended to provide typical responses for basic use cases. So the responses you receive will not contain current data, and may not exactly match your requests in other ways." Essentially, the responses will not match the requests entered but successful pull will indicate whether the entry was valid or not.

auto_open   indicate whether the browser should open automatically to the login URL

Details

Authentication into the ETRADE environment requires a three step process: set ETRADE credentials, generate request tokens to create a login URL, and use the verification code to create access tokens. The output from `etrd_auth_credentials` needs to be fed into this function to create the URL.

Once the URL is generated it will pop up automatically. Log into the page. If the redirect is to a normal ETRADE landing page, re-paste the URL into the browser. The page header should read: Indicate Terms Agreement. Press "Accept" to generate a 5 digit alpha-numeric Verification Code. The Verification Code will feed into `etrd_auth_access_token` to generate Access Tokens and complete the authorization process.

Value

a list of three items: a login URL valid for 5 minutes and two request tokens, a key and secret

See Also

`etrd_auth_credentials` to set the key and secret into an oauth app, `etrd_auth_login_url` to generate request tokens and create a login URL, `etrd_auth_access_token` to use the verification code to create access tokens, `etrd_auth_manage_token` to renew or revoke access tokens

Examples

```
## Not run:

# The etrade_cred has been stored into options so does not need to be passed
loginURL = etrd_auth_login_url()


## End(Not run)
```

---

etrd_auth_manage_token

Auth - Renew or Revoke Access Tokens as needed

Description

#' During the trading day, if two hours have elapsed between API calls, the token is inactivated. Use this function to refresh an inactivated access token. This function can also be used to revoke an active access token. Once a token has been revoked, Auth steps 1-3 will need to be run to get new access tokens.

Usage

```
etrd_auth_manage_token(
  action = c("renew", "revoke"),
  access_tokens = NULL,
  etrade_cred = NULL,
  sandbox = FALSE
)
```

Arguments

action	Enter 'renew' to activate an inactive access token. Enter 'revoke' to invalidate a current access token.
access_tokens	Access tokens are created using `etrd_auth_access_token`. This entry is not required because the output is saved and retrieved from R options automatically.
etrade_cred	The output created from `etrd_auth_credentials` when a valid ETRADE key and secret have been passed. This entry is not required because the output is saved and retrieved from R options automatically.
sandbox	ETRADE offers a sandbox environment for validating API calls and responses. If using the sandbox environment, this must be set to TRUE in each function called throughout etrader. ETRADE states "Sandbox responses use stored data that's intended to provide typical responses for basic use cases. So the responses you receive will not contain current data, and may not exactly match your requests in other ways." Essentially, the responses will not match the requests entered but successful pull will indicate whether the entry was valid or not.

Details

The access tokens expire each day at midnight Eastern Time. The next day, Auth Step 1-3 will need to be run again. This function cannot renew an expired access token.

Value

a key and secret oauth access token

See Also

`etrd_auth_credentials` to set the key and secret into an oauth app, `etrd_auth_login_url` to generate request tokens and create a login URL, `etrd_auth_access_token` to use the verification code to create access tokens, `etrd_auth_manage_token` to renew or revoke access tokens

Examples

```
## Not run:

# Will renew production tokens
etrd_auth_manage_token('renew')

# Will revoke a sandbox access token
etrd_auth_manage_token('revoke', sandbox = TRUE)


## End(Not run)
```

---

etrd_cancel_order   *Cancel an existing order*

Description

Cancel an open order that has been submitted. Note: Verify the cancel request was received and processed

Usage

```
etrd_cancel_order(
  account,
  orderId,
  output = c("df", "list"),
  access_tokens = NULL,
  etrade_cred = NULL,
  sandbox = FALSE
)
```

Arguments

account	The account ID key, not the account ID associated with the specific account. Use [etrd_account_list] to identify the proper account id key.
orderId	Order confirmation Id for the order placed.
output	Indicate whether the output should be in the form of a data frame ('df') or list ('list'). Data frame is returned by default.
access_tokens	Access tokens are created using [etrd_auth_access_token]. This entry is not required because the output is saved and retrieved from R options automatically.
etrade_cred	The output created from [etrd_auth_credentials] when a valid ETRADE key and secret have been passed. This entry is not required because the output is saved and retrieved from R options automatically.

sandbox   ETRADE offers a sandbox environment for validating API calls and responses.
                        If using the sandbox environment, this must be set to TRUE in each function
                        called throughout etrader. ETRADE states "Sandbox responses use stored data
                        that's intended to provide typical responses for basic use cases. So the responses
                        you receive will not contain current data, and may not exactly match your re-
                        quests in other ways." Essentially, the responses will not match the requests
                        entered but successful pull will indicate whether the entry was valid or not.

Value

a response validating that the order has been canceled

Examples

```
## Not run:
# Place Mutual Fund Buy order for $10
acts = etrd_account_list()
account = acts$accountIdKey[1]
mforder = etrd_place_mf_order(account = account,
                     symbol = 'SWTSX',
                     quantityType = 'DOLLAR',
                     investmentAmount = 10,
                     mfTransaction = 'buy',
                     reInvestOption = 'reinvest',
                     previewOrder = 'df')

etrd_cancel_order(mforder$accountidKey, mforder$orderid.orderId)


## End(Not run)
```

---

etrd_get_url   *Get request for a specific URL*

Description

Some functions may return a URL (for example etrd_option_chain) that requires a further GET
request. This function can return the contents. The access token and etrade credentials must be
explicitly passed.

Usage

```
etrd_get_url(etrade_url, access_tokens, etrade_cred)
```

Arguments

etrade_url   The URL for the get request

access_tokens   Access tokens are created using etrd_auth_access_token. This entry is not
                    required because the output is saved and retrieved from R options automatically.

etrade_cred   The output created from [etrd_auth_credentials](#) when a valid ETRADE key
                and secret have been passed. This entry is not required because the output is
                saved and retrieved from R options automatically.

Value

content of the GET request

Examples

```
## Not run:

etrd_get_url('https://api.etrade.com/v1/market/quote/SPY:2020:11:20:PUT:185.000000',
             access_tokens, etrade_cred)


## End(Not run)
```

---

etrd_market_quote   *ETrade Quotes*

Description

Retrieves the quote information for one or more specified symbols. To receive access to real-time
market data, you would need to sign the market data agreement.Otherwise, you will receive delayed
market data.

Usage

```
etrd_market_quote(
  symbols = c("AAPL", "MSFT", "SPY"),
  detailFlag = "ALL",
  output = c("df", "list"),
  access_tokens = NULL,
  etrade_cred = NULL,
  sandbox = FALSE
)
```

Arguments

symbols   One or more symbols for equities or options, up to a maximum of 50 Sym-
              bols for equities are simple, for example, GOOG. Symbols for options are more
              complex, consisting of six elements separated by colons, in this format: under-
              lier:year:month:day:optionType:strikePrice.

detailFlag   Determines the market fields returned from a quote request. The default is
              ALL. Options: ALL, FUNDAMENTAL, INTRADAY, OPTIONS, WEEK_52,
              MF_DETAIL

output	Indicate whether the output should be in the form of a data frame ('df') or list ('list'). Data frame is returned by default.
access_tokens	Access tokens are created using etrd_auth_access_token. This entry is not required because the output is saved and retrieved from R options automatically.
etrade_cred	The output created from etrd_auth_credentials when a valid ETRADE key and secret have been passed. This entry is not required because the output is saved and retrieved from R options automatically.
sandbox	ETRADE offers a sandbox environment for validating API calls and responses. If using the sandbox environment, this must be set to TRUE in each function called throughout etrader. ETRADE states "Sandbox responses use stored data that's intended to provide typical responses for basic use cases. So the responses you receive will not contain current data, and may not exactly match your requests in other ways." Essentially, the responses will not match the requests entered but successful pull will indicate whether the entry was valid or not.

Value

a list or data frame of quote data

Examples

```
## Not run:

# Get quotes for 3 symbols
etrd_market_quote(c('AAPL','MSFT','SPY'))


## End(Not run)
```

---

etrd_option_chain   *Option Chains*

Description

Returns a list of option chains for a specific underlying instrument. The request must specify an instrument, and can include the month the option expires and to show calls, puts, or both. Values returned include the option pair count and information about each option pair, including the type, call count, symbol, product, date, and strike price.

Usage

```
etrd_option_chain(
  symbol = "SPY",
  expiryYear = NULL,
  expiryMonth = NULL,
  expiryDay = NULL,
  strikePriceNear = NULL,
```

```
    noOfStrikes = NULL,
    includeWeekly = "false",
    skipAdjusted = "true",
    optionCategory = NULL,
    chainType = NULL,
    priceType = NULL,
    output = c("df", "list"),
    access_tokens = NULL,
    etrade_cred = NULL,
    sandbox = FALSE
)
```

Arguments

symbol	The market symbol for the instrument
expiryYear	Indicates the expiry year corresponding to which the optionchain needs to be fetched
expiryMonth	Indicates the expiry month corresponding to which the optionchain needs to be fetched
expiryDay	Indicates the expiry day corresponding to which the optionchain needs to be fetched
strikePriceNear	
	The optionchians fetched will have strike price nearer to this value
noOfStrikes	Indicates number of strikes for which the optionchain needs to be fetched
includeWeekly	The include weekly options request. Default: false. Can also be true
skipAdjusted	The skip adjusted request. Default: true. Can also be false
optionCategory	The option category. Default: STANDARD. options include: STANDARD, ALL, MINI
chainType	The type of option chain. Default: CALLPUT. Options include: CALL, PUT, CALLPUT
priceType	The price type. Default: ATNM. Options include ATNM, ALL
output	Indicate whether the output should be in the form of a data frame ('df') or list ('list'). Data frame is returned by default.
access_tokens	Access tokens are created using [etrd_auth_access_token](#). This entry is not required because the output is saved and retrieved from R options automatically.
etrade_cred	The output created from [etrd_auth_credentials](#) when a valid ETRADE key and secret have been passed. This entry is not required because the output is saved and retrieved from R options automatically.
sandbox	ETRADE offers a sandbox environment for validating API calls and responses. If using the sandbox environment, this must be set to TRUE in each function called throughout etrader. ETRADE states "Sandbox responses use stored data that's intended to provide typical responses for basic use cases. So the responses you receive will not contain current data, and may not exactly match your requests in other ways." Essentially, the responses will not match the requests entered but successful pull will indicate whether the entry was valid or not.

Value

a list or data frame of options chains

Examples

```
## Not run:

# Get SPY Expirations
etrd_option_chain('SPY')


## End(Not run)
```

etrd_option_expiration

Option Expiration Dates

Description

Returns a list or data frame of dates suitable for structuring an option table display. The dates are used to group option data (returned by the option chains method) for a specified underlier, creating a table display.

Usage

```
etrd_option_expiration(
  symbol = "SPY",
  output = c("df", "list"),
  access_tokens = NULL,
  etrade_cred = NULL,
  sandbox = FALSE
)
```

Arguments

symbol	The symbol in the request
output	Indicate whether the output should be in the form of a data frame ('df') or list ('list'). Data frame is returned by default.
access_tokens	Access tokens are created using [etrd_auth_access_token](). This entry is not required because the output is saved and retrieved from R options automatically.
etrade_cred	The output created from [etrd_auth_credentials]() when a valid ETRADE key and secret have been passed. This entry is not required because the output is saved and retrieved from R options automatically.

sandbox   ETRADE offers a sandbox environment for validating API calls and responses. If using the sandbox environment, this must be set to TRUE in each function called throughout etrader. ETRADE states "Sandbox responses use stored data that's intended to provide typical responses for basic use cases. So the responses you receive will not contain current data, and may not exactly match your requests in other ways." Essentially, the responses will not match the requests entered but successful pull will indicate whether the entry was valid or not.

Value

a data frame of expiration dates for the symbol requested

Examples

```
## Not run:

# Get SPY Expirations
etrd_option_expiration('SPY')


## End(Not run)
```

---

etrd_order_history   *ETrade order-related information*

Description

Provides the order details for a selected brokerage account based on the search criteria provided.

Usage

```
etrd_order_history(
  account,
  count = 100,
  status = NULL,
  fromDate = Sys.Date() - 30,
  toDate = Sys.Date(),
  symbol = NULL,
  securityType = NULL,
  transactionType = NULL,
  marketSession = NULL,
  output = c("df", "list"),
  access_tokens = NULL,
  etrade_cred = NULL,
  sandbox = FALSE
)
```

Arguments

account	The account ID key, not the account ID associated with the specific account. Use [etrd_account_list](#) to identify the proper account id key.
count	Number of orders to return with a max of 100
status	a valid status: OPEN, EXECUTED, CANCELLED, INDIVIDUAL_FILLS, CANCEL_REQUESTED, EXPIRED, REJECTED, PARTIAL, DO_NOT_EXERCISE, DONE_TRADE_EXECUTED
fromDate	a date object for the start of the filter
toDate	a data object for the end of the filter
symbol	a specific symbol to filter for
securityType	a valid security type: EQ, OPTN, MF, MMF
transactionType	
	a valid transaction type: ATNM, BUY, SELL, SELL_SHORT, BUY_TO_COVER, MF_EXCHANGE
marketSession	REGULAR, EXTENDED
output	Indicate whether the output should be in the form of a data frame ('df') or list ('list'). Data frame is returned by default.
access_tokens	Access tokens are created using [etrd_auth_access_token](#). This entry is not required because the output is saved and retrieved from R options automatically.
etrade_cred	The output created from [etrd_auth_credentials](#) when a valid ETRADE key and secret have been passed. This entry is not required because the output is saved and retrieved from R options automatically.
sandbox	ETRADE offers a sandbox environment for validating API calls and responses. If using the sandbox environment, this must be set to TRUE in each function called throughout etrader. ETRADE states "Sandbox responses use stored data that's intended to provide typical responses for basic use cases. So the responses you receive will not contain current data, and may not exactly match your requests in other ways." Essentially, the responses will not match the requests entered but successful pull will indicate whether the entry was valid or not.

Value

a list or df of orders

Examples

```
## Not run:

# Search for mutual fund buys over the last month
account_id = etrd_account_list()$accountIdKey[1]
etrd_order_history(account_id, transactionType = 'BUY', securityType = 'mf',
                   status = 'EXECUTED', fromDate = Sys.Date()-30, toDate = Sys.Date())


## End(Not run)
```

---

etrd_place_eq_order   *Place an Equity Order*

Description

Execute an equity order on the ETrade platform for the selected account. Users are given the option to preview the order before submitting.

Usage

```
etrd_place_eq_order(
  account,
  symbol,
  quantity,
  orderAction,
  priceType,
  stopPrice = "",
  limitPrice = "",
  stopLimitPrice = "",
  quantityType = "quantity",
  orderTerm = "good_for_day",
  marketSession = "regular",
  allOrNone = "false",
  previewOrder = "df",
  etrade_cred = NULL,
  access_tokens = NULL,
  sandbox = FALSE
)
```

Arguments

account	The account ID key, not the account ID associated with the specific account. Use [etrd_account_list](#) to identify the proper account id key.
symbol	The market symbol for the security being bought or sold
quantity	The number of shares to buy or sell
orderAction	The action that the broker is requested to perform: BUY, SELL, BUY_TO_COVER, SELL_SHORT
priceType	The type of pricing: MARKET, LIMIT, STOP, STOP_LIMIT, TRAILING_STOP_CNST, TRAILING_STOP_PRCT, MARKET_ON_OPEN, MARKET_ON_CLOSE, LIMIT_ON_OPEN, LIMIT_ON_CLOSE
stopPrice	The designated boundary price for a stop order. For trailing stop orders this will represent the dollar amount or percentage trailing value. Enter percentages as whole numbers.
limitPrice	The highest price at which to buy or the lowest price at which to sell if specified in a limit order

stopLimitPrice	The designated boundary price for a stop-limit order
quantityType	The type of the quantity - QUANTITY, DOLLAR, ALL_I_OWN (note: DOLLAR is not yet an option)
orderTerm	The term for which the order is in effect: GOOD_UNTIL_CANCEL, GOOD_FOR_DAY, IMMEDIATE_OR_CANCEL, FILL_OR_KILL
marketSession	The session in which the order will be placed: REGULAR, EXTENDED
allOrNone	If TRUE, the transactions specified in the order must be executed all at once or not at all; default is FALSE
previewOrder	Elect to preview the order before submitting. Options are: 'df', 'json', or 'none' for an instant order entry. In a non interactive environment will default to 'none'. This also indicates how the order output is shown. json will return the output as a list, otherwise it will return as a dataframe
etrade_cred	The output created from `etrd_auth_credentials` when a valid ETRADE key and secret have been passed. This entry is not required because the output is saved and retrieved from R options automatically.
access_tokens	Access tokens are created using `etrd_auth_access_token`. This entry is not required because the output is saved and retrieved from R options automatically.
sandbox	ETRADE offers a sandbox environment for validating API calls and responses. If using the sandbox environment, this must be set to TRUE in each function called throughout etrader. ETRADE states "Sandbox responses use stored data that's intended to provide typical responses for basic use cases. So the responses you receive will not contain current data, and may not exactly match your requests in other ways." Essentially, the responses will not match the requests entered but successful pull will indicate whether the entry was valid or not.

Value

A list or data frame of the order entry details

Warning

TRADES THAT ARE SUCCESSFULLY ENTERED WILL BE SENT FOR EXECUTION. THIS FUNCTION HAS HUNDREDS OF POTENTIAL COMBINATIONS AND ONLY A HANDFUL HAVE BEEN TESTED. IT IS STRONGLY RECOMMENDED TO TEST THE DESIRED ORDER USING THE SANDBOX ENVIRONMENT. PLEASE NOTE THE SANDBOX ENVIRONMENT WILL NOT RETURN THE ORDER ENTERED, IT IS USED FOR CONFIRMING THE ORDER ENTRY HAS CORRECT SYNTAX. ORDERS CAN ALSO BE TESTED IN OFF MARKET HOURS ON A VERY SMALL QUANTITY WITH LITTLE MONEY AT STAKE. ANOTHER OPTION IS TO USE LIMIT ORDERS FAR FROM THE CURRENT PRICE. ETRADE HAS THEIR OWN ERROR HANDLING BUT IF A SUCCESSFUL COMBINATION IS ENTERED IT COULD BE EXECUTED UNINTENTIONALLY. DOUBLE CHECK ALL ENTRIES BEFORE SUBMITTING.

Examples

```
## Not run:
```

```
# Place Equity Limit order
acts = etrd_account_list()
account = acts$accountIdKey[1]
etrd_place_eq_order(account = account,
                    symbol = 'PSLV',
                    quantity = 1,
                    orderAction = 'buy',
                    priceType = 'limit',
                    limitPrice = 8,
                    previewOrder = 'df')



## End(Not run)
```

---

etrd_place_mf_order   *Place a Mutual Fund Order*

Description

Submit a mutual fund order on the ETrade platform for the selected account. Users are given the option to preview the order before submitting. Mutual fund orders must be received before 4pm or will be executed the following day.

Usage

```
etrd_place_mf_order(
  account,
  symbol,
  quantity,
  mfTransaction,
  reInvestOption = "REINVEST",
  quantityType = "DOLLAR",
  previewOrder = "df",
  etrade_cred = NULL,
  access_tokens = NULL,
  sandbox = FALSE
)
```

Arguments

account	The account ID key, not the account ID associated with the specific account. Use [etrd_account_list](#) to identify the proper account id key.
symbol	The market symbol for the security being bought or sold
quantity	The amount of the investment in either DOLLARS or SHARES depending on the input for quantityType
mfTransaction	The transaction for the mutual fund order. Options: BUY, SELL

reInvestOption	Indicator flag to specify whether to reinvest profit on mutual funds. Options: REINVEST, DEPOSIT, CURRENT_HOLDING
quantityType	The type of the quantity - QUANTITY, DOLLAR, ALL_I_OWN (note: DOLLAR is not yet an option)
previewOrder	Elect to preview the order before submitting. Options are: 'df', 'json', or 'none' for an instant order entry. In a non interactive environment will default to 'none'. This also indicates how the order output is shown. json will return the output as a list, otherwise it will return as a dataframe
etrade_cred	The output created from `etrd_auth_credentials` when a valid ETRADE key and secret have been passed. This entry is not required because the output is saved and retrieved from R options automatically.
access_tokens	Access tokens are created using `etrd_auth_access_token`. This entry is not required because the output is saved and retrieved from R options automatically.
sandbox	ETRADE offers a sandbox environment for validating API calls and responses. If using the sandbox environment, this must be set to TRUE in each function called throughout etrader. ETRADE states "Sandbox responses use stored data that's intended to provide typical responses for basic use cases. So the responses you receive will not contain current data, and may not exactly match your requests in other ways." Essentially, the responses will not match the requests entered but successful pull will indicate whether the entry was valid or not.

Value

A list or data frame of the order entry details

Warning

TRADES THAT ARE SUCCESSFULLY ENTERED WILL BE SENT FOR EXECUTION. THIS FUNCTION HAS HUNDREDS OF POTENTIAL COMBINATIONS AND ONLY A HANDFUL HAVE BEEN TESTED. IT IS STRONGLY RECOMMENDED TO TEST THE DESIRED ORDER USING THE SANDBOX ENVIRONMENT. PLEASE NOTE THE SANDBOX ENVIRONMENT WILL NOT RETURN THE ORDER ENTERED, IT IS USED FOR CONFIRMING THE ORDER ENTRY HAS CORRECT SYNTAX. ORDERS CAN ALSO BE TESTED IN OFF MARKET HOURS ON A VERY SMALL QUANTITY WITH LITTLE MONEY AT STAKE. ETRADE HAS THEIR OWN ERROR HANDLING BUT IF A SUCCESSFUL COMBINATION IS ENTERED IT COULD BE EXECUTED UNINTENTIONALLY. DOUBLE CHECK ALL ENTRIES BEFORE SUBMITTING.

Examples

```
## Not run:

# Place Mutual Fund Buy order for $10
acts = etrd_account_list()
account = acts$accountIdKey[1]
etrd_place_mf_order(account = account,
                    symbol = 'SWTSX',
                    quantityType = 'DOLLAR',
```

```
                        quantity = 10,
                        mfTransaction = 'buy',
                        reInvestOption = 'reinvest',
                        previewOrder = 'df')


## End(Not run)
```

---

etrd_place_optn_order *Place an Option Order*

Description

Execute an option order on the ETrade platform for the selected account. Users are given the option to preview the order before submitting. Note: ETrade offers significantly more complex order structures than what is offered in this function. See the ETrade documentation for more details on submitting complex option strategies.

Usage

```
etrd_place_optn_order(
  account,
  symbol,
  callPut,
  expiryYear,
  expiryMonth,
  expiryDay,
  strikePrice,
  quantity,
  orderAction,
  priceType,
  stopPrice = "",
  limitPrice = "",
  stopLimitPrice = "",
  quantityType = "quantity",
  orderTerm = "good_for_day",
  marketSession = "regular",
  allOrNone = "false",
  previewOrder = "df",
  etrade_cred = NULL,
  access_tokens = NULL,
  sandbox = FALSE
)
```

Arguments

account   The account ID key, not the account ID associated with the specific account. Use `etrd_account_list` to identify the proper account id key.

symbol	The market symbol for the security being bought or sold
callPut	The option type: CALL, PUT
expiryYear	The four-digit year the option will expire
expiryMonth	The month (1-12) the option will expire
expiryDay	The day (1-31) the option will expire
strikePrice	The strike price for the option
quantity	The number of shares to buy or sell
orderAction	The action that the broker is requested to perform: BUY_OPEN, BUY_CLOSE, SELL_OPEN, SELL_CLOSE
priceType	The type of pricing: MARKET, LIMIT, STOP, STOP_LIMIT, TRAILING_STOP_CNST, TRAILING_STOP_PRCT, MARKET_ON_OPEN, MARKET_ON_CLOSE, LIMIT_ON_OPEN, LIMIT_ON_CLOSE
stopPrice	The designated boundary price for a stop order. For trailing stop orders this will represent the dollar amount or percentage trailing value. Enter percentages as whole numbers.
limitPrice	The highest price at which to buy or the lowest price at which to sell if specified in a limit order
stopLimitPrice	The designated boundary price for a stop-limit order
quantityType	The type of the quantity - QUANTITY, DOLLAR, ALL_I_OWN (note: DOLLAR is not yet an option)
orderTerm	The term for which the order is in effect: GOOD_UNTIL_CANCEL, GOOD_FOR_DAY, IMMEDIATE_OR_CANCEL, FILL_OR_KILL
marketSession	The session in which the order will be placed: REGULAR, EXTENDED
allOrNone	If TRUE, the transactions specified in the order must be executed all at once or not at all; default is FALSE
previewOrder	Elect to preview the order before submitting. Options are: 'df', 'json', or 'none' for an instant order entry. In a non interactive environment will default to 'none'. This also indicates how the order output is shown. json will return the output as a list, otherwise it will return as a dataframe
etrade_cred	The output created from [etrd_auth_credentials](#) when a valid ETRADE key and secret have been passed. This entry is not required because the output is saved and retrieved from R options automatically.
access_tokens	Access tokens are created using [etrd_auth_access_token](#). This entry is not required because the output is saved and retrieved from R options automatically.
sandbox	ETRADE offers a sandbox environment for validating API calls and responses. If using the sandbox environment, this must be set to TRUE in each function called throughout etrader. ETRADE states "Sandbox responses use stored data that's intended to provide typical responses for basic use cases. So the responses you receive will not contain current data, and may not exactly match your requests in other ways." Essentially, the responses will not match the requests entered but successful pull will indicate whether the entry was valid or not.

Value

A list or data frame of the order entry details

Warning

TRADES THAT ARE SUCCESSFULLY ENTERED WILL BE SENT FOR EXECUTION. THIS FUNCTION HAS HUNDREDS OF POTENTIAL COMBINATIONS AND ONLY A HANDFUL HAVE BEEN TESTED. IT IS STRONGLY RECOMMENDED TO TEST THE DESIRED ORDER USING THE SANDBOX ENVIRONMENT. PLEASE NOTE THE SANDBOX ENVIRONMENT WILL NOT RETURN THE ORDER ENTERED, IT IS USED FOR CONFIRMING THE OR-DER ENTRY HAS CORRECT SYNTAX. ORDERS CAN ALSO BE TESTED IN OFF MARKET HOURS ON A VERY SMALL QUANTITY WITH LITTLE MONEY AT STAKE. ANOTHER OPTION IS TO USE LIMIT ORDERS FAR FROM THE CURRENT PRICE. ETRADE HAS THEIR OWN ERROR HANDLING BUT IF A SUCCESSFUL COMBINATION IS ENTERED IT COULD BE EXECUTED UNINTENTIONALLY. DOUBLE CHECK ALL ENTRIES BEFORE SUBMITTING.

Examples

```
## Not run:

# Place Option Buy_to_open order
acts = etrd_account_list()
account = acts$accountIdKey[1]
etrd_place_optn_order(account = account,
                      symbol = 'FB',
                      callPut = 'call',
                      expiryYear = '2018',
                      expiryMonth = '12',
                      expiryDay  = '21',
                      strikePrice = '140',
                      quantity = 1,
                      orderAction = 'BUY_OPEN',
                      priceType = 'market',
                      previewOrder = 'df')


## End(Not run)
```

---

etrd_product_search   *Product Search*

Description

Returns a list of securities of a specified type (e.g., equity stock) based on a full or partial match of any part of the company name. For instance, a search for "jones" returns a list of securities associated with "Jones Soda Co", "Stella Jones Inc", and many others. The list contains the company name, the exchange that lists the security, the security type, and the symbol, for up to 10 matches.

The result may include some unexpected matches, because the search includes more than just the display version of the company name. For instance, searching on "etrade" returns securities for "E TRADE" - notice the space in the name. This API is for searching on the company name, not a security symbol. It's commonly used to look up a symbol based on the company name, e.g., "What is the symbol for Google stock?". To look up company information based on a symbol, or to find detailed information on a security, use the quote API.

Usage

```
etrd_product_search(
  search = "tech",
  access_tokens = NULL,
  etrade_cred = NULL,
  sandbox = FALSE
)
```

Arguments

search	The search request
access_tokens	Access tokens are created using `etrd_auth_access_token`. This entry is not required because the output is saved and retrieved from R options automatically.
etrade_cred	The output created from `etrd_auth_credentials` when a valid ETRADE key and secret have been passed. This entry is not required because the output is saved and retrieved from R options automatically.
sandbox	ETRADE offers a sandbox environment for validating API calls and responses. If using the sandbox environment, this must be set to TRUE in each function called throughout etrader. ETRADE states "Sandbox responses use stored data that's intended to provide typical responses for basic use cases. So the responses you receive will not contain current data, and may not exactly match your requests in other ways." Essentially, the responses will not match the requests entered but successful pull will indicate whether the entry was valid or not.

Value

a Data frame of the search results

Examples

```
## Not run:

# search for tech firms
etrd_product_search('tech')


## End(Not run)
```

---

etrd_transactions *ETrade Transaction Search*

Description

Provides the transactions for a selected brokerage account based on the search criteria provided.

Usage

```
etrd_transactions(
  account,
  count = 50,
  fromDate = Sys.Date() - 30,
  toDate = Sys.Date(),
  output = c("df", "list"),
  access_tokens = NULL,
  etrade_cred = NULL,
  sandbox = FALSE
)
```

Arguments

account	The account ID key, not the account ID associated with the specific account. Use `etrd_account_list` to identify the proper account id key.
count	Number of transactions to return in the response with a max of 50. If not specified, defaults to 50.
fromDate	a date object for the start of the filter
toDate	a data object for the end of the filter
output	Indicate whether the output should be in the form of a data frame ('df') or list ('list'). Data frame is returned by default.
access_tokens	Access tokens are created using `etrd_auth_access_token`. This entry is not required because the output is saved and retrieved from R options automatically.
etrade_cred	The output created from `etrd_auth_credentials` when a valid ETRADE key and secret have been passed. This entry is not required because the output is saved and retrieved from R options automatically.
sandbox	ETRADE offers a sandbox environment for validating API calls and responses. If using the sandbox environment, this must be set to TRUE in each function called throughout etrader. ETRADE states "Sandbox responses use stored data that's intended to provide typical responses for basic use cases. So the responses you receive will not contain current data, and may not exactly match your requests in other ways." Essentially, the responses will not match the requests entered but successful pull will indicate whether the entry was valid or not.

Value

a list or df of transactions

Examples

```
## Not run:

# Search for mutual fund buys over the last month
account_id = etrd_account_list()$accountIdKey[1]
etrd_transactions(account_id, fromDate = Sys.Date()-30, toDate = Sys.Date())


## End(Not run)
```

---

etrd_transaction_detail

ETrade Transaction Details

Description

Get transaction details for the specified transaction

Usage

```
etrd_transaction_detail(
  account,
  transactionId,
  output = c("df", "list"),
  access_tokens = NULL,
  etrade_cred = NULL,
  sandbox = FALSE
)
```

Arguments

account	The account ID key, not the account ID associated with the specific account. Use `etrd_account_list` to identify the proper account id key.
transactionId	A transaction id that comes from `etrd_transactions`
output	Indicate whether the output should be in the form of a data frame ('df') or list ('list'). Data frame is returned by default.
access_tokens	Access tokens are created using `etrd_auth_access_token`. This entry is not required because the output is saved and retrieved from R options automatically.
etrade_cred	The output created from `etrd_auth_credentials` when a valid ETRADE key and secret have been passed. This entry is not required because the output is saved and retrieved from R options automatically.
sandbox	ETRADE offers a sandbox environment for validating API calls and responses. If using the sandbox environment, this must be set to TRUE in each function called throughout etrader. ETRADE states "Sandbox responses use stored data that's intended to provide typical responses for basic use cases. So the responses you receive will not contain current data, and may not exactly match your requests in other ways." Essentially, the responses will not match the requests entered but successful pull will indicate whether the entry was valid or not.

Value

a list or data frame of transaction details

Examples

```
## Not run:

# Search for mutual fund buys over the last month
account_id = etrd_account_list()$accountIdKey[1]
transID = etrd_transactions(account_id)$transactionId[1]
etrd_transaction_detail(account_id, transID)


## End(Not run)
```

Index