Package 'igfetchr'

September 26, 2025

Type Package

Title Simplified Access to IG Trading API

Version 0.1.0

Maintainer Saw Simeon <saw.s@ku.th>

Description Provides functions to fetch market data, search historical prices, execute trades, and get account details from the IG Trading REST API (labs.ig.com). Returns tidy tibbles for easy analysis. Trading CFDs and spread bets carry a high risk of losing money; this package is is not financial advice.

License GPL-3

Encoding UTF-8

Imports httr, jsonlite, tibble

Suggests testthat (>= 3.0.0), knitr, rmarkdown, lubridate, httptest

VignetteBuilder knitr

URL https://github.com/sawsimeon/igfetchr

BugReports https://github.com/sawsimeon/igfetchr/issues

RoxygenNote 7.3.3

NeedsCompilation no

Author Saw Simeon [aut, cre]

Contents

igfetchr-package 2
.ig_request 2
ig_auth 3
ig_close_session 4
ig_execute_trade 5
ig_get_accounts 6
ig_get_historical 7
ig_get_options 8
ig_get_price 9
ig_search_markets 10

Index **12**

igfetchr-package *igfetchr: Access the IG Trading REST API*

Description

igfetchr provides a small set of beginner-friendly functions to authenticate with and fetch market data, historical prices, account details, options positions, and execute trades from the IG Trading REST API (labs.ig.com). All functions return tidy tibbles for straightforward analysis.

Helper functions to authenticate and fetch market/account data from the IG Trading REST API. All exported functions return tibbles where appropriate. This package is for data access only and not financial advice. Trading CFDs and spread bets carries a high risk of losing money.

Details

Disclaimer: Trading CFDs, spread bets, and options carries a high risk of losing money. This package provides data access and trade execution functionality only and is not financial or trading advice.

Author(s)

Maintainer: Saw Simeon <saw.s@ku.th>

See Also

Useful links:

- https://github.com/sawsimeon/igfetchr
- Report bugs at https://github.com/sawsimeon/igfetchr/issues

Useful links:

- https://github.com/sawsimeon/igfetchr
- Report bugs at https://github.com/sawsimeon/igfetchr/issues

.ig_request *Internal function to make IG API requests*

Description

Makes HTTP requests to the IG API using authentication details from ‘ig_auth()‘. Handles GET, POST, and DELETE methods and processes JSON responses into tibbles. Not intended for direct use; called by higher-level functions like ‘ig_get_accounts()‘.

Usage

```
.ig_request(
  path,
  auth,
  method = c("GET", "POST", "DELETE"),
  query = list(),
  body = NULL,
  mock_response = NULL
)
```

Arguments

path	Character. API endpoint path (e.g., "/accounts").
auth	List. Authentication details from ‘ig_auth()‘, including ‘cst‘, ‘security‘, ‘base_url‘, ‘api_key‘, and ‘acc_number‘.
method	Character. HTTP method ("GET", "POST", or "DELETE"). Defaults to "GET".
query	List. Optional query parameters for GET requests.
body	List. Optional request body for POST requests.
mock_response	List or data frame. Optional mock response for testing.

Value

A tibble containing the API response data, or a tibble with a single column containing the raw JSON response if the response cannot be converted to a data frame.

---

ig_auth *Authenticate with the IG API*

Description

Authenticates with the IG API to obtain session tokens (CST and X-SECURITY-TOKEN) for subsequent API requests. Supports environment variables for credentials and optional account type and number.

Usage

```
ig_auth(
  username = Sys.getenv("IG_SERVICE_USERNAME"),
  password = Sys.getenv("IG_SERVICE_PASSWORD"),
  api_key = Sys.getenv("IG_SERVICE_API_KEY"),
  acc_type = Sys.getenv("IG_SERVICE_ACC_TYPE", "DEMO"),
  acc_number = Sys.getenv("IG_SERVICE_ACC_NUMBER")
)
```

Arguments

username	Character. IG account username. Defaults to ‘IG_SERVICE_USERNAME‘ environment variable.
password	Character. IG account password. Defaults to ‘IG_SERVICE_PASSWORD‘ environment variable.
api_key	Character. IG API key. Defaults to ‘IG_SERVICE_API_KEY‘ environment variable.
acc_type	Character. Account type, either "DEMO" or "LIVE". Defaults to "DEMO".
acc_number	Character. Optional account number. Defaults to ‘IG_SERVICE_ACC_NUMBER‘ or NULL.

Value

A list containing:

- `cst`: Client session token.
- `security`: Security token (X-SECURITY-TOKEN).
- `base_url`: Base URL for API requests (DEMO or LIVE).
- `api_key`: API key used for authentication.
- `acc_type`: Account type ("DEMO" or "LIVE").
- `acc_number`: Account number (or NULL if not provided).

Examples

```
## Not run:
# Using environment variables
Sys.setenv(IG_SERVICE_USERNAME = "your_username")
Sys.setenv(IG_SERVICE_PASSWORD = "your_password")
Sys.setenv(IG_SERVICE_API_KEY = "your_api_key")
Sys.setenv(IG_SERVICE_ACC_NUMBER = "ABC123")
Sys.setenv(IG_SERVICE_ACC_TYPE = "DEMO")
auth <- ig_auth()

# Using explicit arguments
auth <- ig_auth(
  username = "your_username",
  password = "your_password",
  api_key = "your_api_key",
  acc_type = "DEMO",
  acc_number = "ABC123"
)

## End(Not run)
```

---

ig_close_session *Close session*

Description

Closes the authenticated IG API session.

Alias for 'ig_close_session()'.

Usage

```
ig_close_session(auth, mock_response = NULL)

ig_logout(auth, mock_response = NULL)
```

Arguments

`auth`	List. Authentication details from 'ig_auth()', including 'cst', 'security', 'base_url', 'api_key', and 'acc_number'.
`mock_response`	Logical. Optional mock response for testing (returns 'TRUE' if provided). Defaults to 'NULL'.

Value

Logical ‘TRUE‘ if the session was closed successfully.

Examples

```
## Not run:
# Authenticate and close session
auth <- ig_auth(
  username = "your_username",
  password = "your_password",
  api_key = "your_api_key",
  acc_type = "DEMO",
  acc_number = "ABC123"
)
ig_close_session(auth)

# Using mock response for testing
ig_close_session(auth, mock_response = TRUE)

## End(Not run)
```

---

ig_execute_trade *Execute a trade (place OTC position)*

Description

Places a market trade (BUY/SELL) for a specified market epic. For testing, use ‘mock_response‘ to avoid network calls.

Usage

```
ig_execute_trade(
  epic,
  direction,
  size,
  auth,
  limit = NULL,
  stop = NULL,
  mock_response = NULL
)
```

Arguments

epic	Character. Market epic (e.g., "CS.D.USDCHF.CFD.IP").
direction	Character. Trade direction, either "BUY" or "SELL".
size	Numeric. Trade size (units).
auth	List. Authentication details from ‘ig_auth()‘, including ‘cst‘, ‘security‘, ‘base_url‘, ‘api_key‘, and ‘acc_number‘.
limit	Numeric. Optional limit price for the trade.
stop	Numeric. Optional stop price for the trade.
mock_response	List or data frame. Optional mock response for testing, bypassing the API call.

Value

A tibble with trade confirmation details, including columns like ‘dealId‘, ‘dealReference‘, ‘status‘, and others as returned by the IG API ‘/positions/otc‘ endpoint.

Examples

```
## Not run:
# Authenticate and execute a trade
auth <- ig_auth(
  username = "your_username",
  password = "your_password",
  api_key = "your_api_key",
  acc_type = "DEMO",
  acc_number = "ABC123"
)
res <- ig_execute_trade("CS.D.USDCHF.CFD.IP", "BUY", 1.0, auth, limit = 0.855, stop = 0.845)
print(res)

# Using mock response for testing
mock_response <- data.frame(
  dealId = "DIXXXX",
  dealReference = "REF123",
  status = "OPEN"
)
res <- ig_execute_trade("CS.D.USDCHF.CFD.IP", "BUY", 1.0, auth, mock_response = mock_response)

## End(Not run)
```

---

ig_get_accounts *Retrieve IG account details*

Description

Fetches details of IG accounts associated with the authenticated session. Returns a tibble containing account information such as account ID, name, balance, and status.

Usage

```
ig_get_accounts(auth, mock_response = NULL)
```

Arguments

auth	List. Authentication details from ‘ig_auth()‘, including ‘cst‘, ‘security‘, ‘base_url‘, ‘api_key‘, and ‘acc_number‘.
mock_response	List or data frame. Optional mock response for testing, bypassing the API call.

Value

A tibble with columns including ‘accountId‘, ‘accountName‘, ‘balance‘, ‘currency‘, and others as returned by the IG API ‘/accounts‘ endpoint.

Examples

```
## Not run:
# Authenticate and get accounts
auth <- ig_auth(
  username = "your_username",
  password = "your_password",
  api_key = "your_api_key",
  acc_type = "DEMO",
  acc_number = "ABC123"
)
accounts <- ig_get_accounts(auth)
print(accounts)

# Using mock response for testing
mock_response <- data.frame(
  accountId = "ABC123",
  accountName = "Demo Account",
  balance.balance = 10000,
  currency = "SEK"
)
accounts <- ig_get_accounts(auth, mock_response = mock_response)

## End(Not run)
```

---

ig_get_historical *Get historical prices*

Description

Fetches historical prices for a market epic between specified dates at a given resolution.

Usage

```
ig_get_historical(
  epic,
  from = NULL,
  to = NULL,
  resolution = "D",
  auth,
  mock_response = NULL
)
```

Arguments

epic	Character. Market epic (e.g., "CS.D.USDCHF.CFD.IP").
from	Character or Date. Start datetime or date (e.g., "2020-01-01"). Optional.
to	Character or Date. End datetime or date (e.g., "2020-12-31"). Optional.
resolution	Character. Resolution code (e.g., "D" for daily, "H" for hourly). Defaults to "D".
auth	List. Authentication details from ‘ig_auth()‘, including ‘cst‘, ‘security‘, ‘base_url‘, ‘api_key‘, and ‘acc_number‘.
mock_response	List or data frame. Optional mock response for testing, bypassing the API call.

Value

A tibble with historical OHLC data, including columns like ‘snapshotTime‘, ‘openPrice‘, ‘highPrice‘, ‘lowPrice‘, ‘closePrice‘, and others as returned by the IG API ‘/prices/{epic}‘ endpoint.

Examples

```
## Not run:
# Authenticate and get historical prices
auth <- ig_auth(
  username = "your_username",
  password = "your_password",
  api_key = "your_api_key",
  acc_type = "DEMO",
  acc_number = "ABC123"
)
hist <- ig_get_historical("CS.D.USDCHF.CFD.IP", "2020-01-01", "2020-12-31", "D", auth)
print(hist)

# Using mock response for testing
mock_response <- data.frame(
  snapshotTime = "2020/01/01 00:00:00",
  openPrice = 0.970,
  highPrice = 0.975,
  lowPrice = 0.965,
  closePrice = 0.971
)
hist <- ig_get_historical("CS.D.USDCHF.CFD.IP",
  "2020-01-01",
  "2020-12-31",
  "D",
  auth,
  mock_response = mock_response)

## End(Not run)
```

---

ig_get_options *Get options/derivative positions*

Description

Retrieves positions filtered for options/derivatives from the IG API. Returns a tibble with position details.

Usage

```
ig_get_options(auth, mock_response = NULL)
```

Arguments

auth	List. Authentication details from ‘ig_auth()‘, including ‘cst‘, ‘security‘, ‘base_url‘, ‘api_key‘, and ‘acc_number‘.
mock_response	List or data frame. Optional mock response for testing, bypassing the API call.

Value

A tibble with options/derivative position details, including columns like ‘dealId‘, ‘size‘, ‘direction‘, ‘instrumentType‘, and others as returned by the IG API ‘/positions‘ endpoint. If ‘instrumentType‘ is available, filters for "OPTION", "DERIVATIVE", or "OPTION_CONTRACT"; otherwise, returns all positions.

Examples

```
## Not run:
# Authenticate and get options positions
auth <- ig_auth(
  username = "your_username",
  password = "your_password",
  api_key = "your_api_key",
  acc_type = "DEMO",
  acc_number = "ABC123"
)
options <- ig_get_options(auth)
print(options)

# Using mock response for testing
mock_response <- data.frame(
  dealId = "DIXXXX",
  size = 1.5,
  direction = "BUY",
  instrumentType = "OPTION"
)
options <- ig_get_options(auth, mock_response = mock_response)

## End(Not run)
```

---

ig_get_price *Get current price for a market*

Description

Fetches current price(s) for the given market epic from the IG API.

Usage

```
ig_get_price(epic, auth, mock_response = NULL)
```

Arguments

epic	Character. Market epic (e.g., "CS.D.USDCHF.CFD.IP").
auth	List. Authentication details from ‘ig_auth()‘, including ‘cst‘, ‘security‘, ‘base_url‘, ‘api_key‘, and ‘acc_number‘.
mock_response	List or data frame. Optional mock response for testing, bypassing the API call.

Value

A tibble with price information, including columns like ‘snapshotTime‘, ‘openPrice‘, ‘closePrice‘, ‘highPrice‘, ‘lowPrice‘, and others as returned by the IG API ‘/prices/{epic}‘ endpoint.

Examples

```
## Not run:
# Authenticate and get price
auth <- ig_auth(
  username = "your_username",
  password = "your_password",
  api_key = "your_api_key",
  acc_type = "DEMO",
  acc_number = "ABC123"
)
price <- ig_get_price("CS.D.USDCHF.CFD.IP", auth)
print(price)

# Using mock response for testing
mock_response <- data.frame(
  snapshotTime = "2025/09/23 10:00:00",
  openPrice = 0.850,
  closePrice = 0.851,
  highPrice = 0.852,
  lowPrice = 0.849
)
price <- ig_get_price("CS.D.USDCHF.CFD.IP", auth, mock_response = mock_response)

## End(Not run)
```

---

ig_search_markets *Search markets*

Description

Search markets by text query. Returns a tibble of matching markets from the IG API.

Usage

```
ig_search_markets(query, auth, mock_response = NULL)
```

Arguments

query	Character. Search string for markets (e.g., "USD/CHF").
auth	List. Authentication details from ‘ig_auth()‘, including ‘cst‘, ‘security‘, ‘base_url‘, ‘api_key‘, and ‘acc_number‘.
mock_response	List or data frame. Optional mock response for testing, bypassing the API call.

Value

A tibble with market information, including columns like ‘epic‘, ‘instrumentName‘, ‘marketStatus‘, and others as returned by the IG API ‘/markets‘ endpoint.

Examples

```
## Not run:
# Authenticate and search markets
auth <- ig_auth(
  username = "your_username",
  password = "your_password",
  api_key = "your_api_key",
  acc_type = "DEMO",
  acc_number = "ABC123"
)
markets <- ig_search_markets("USD/CHF", auth)
print(markets)

# Using mock response for testing
mock_response <- data.frame(
  epic = "CS.D.USDCHF.MINI.IP",
  instrumentName = "USD/CHF Mini",
  marketStatus = "OPEN"
)
markets <- ig_search_markets("USD/CHF", auth, mock_response = mock_response)

## End(Not run)
```

Index

∗ **internal**
    .ig_request, 2
    igfetchr-package, 2
.ig_request, 2

ig_auth, 3
ig_close_session, 4
ig_execute_trade, 5
ig_get_accounts, 6
ig_get_historical, 7
ig_get_options, 8
ig_get_price, 9
ig_logout *(*ig_close_session*)*, 4
ig_search_markets, 10
igfetchr-package, 2