Package: ibkrcp (via r-universe)

May 14, 2026

Title R Client for the Interactive Brokers Client Portal API

Version 0.1.2

Description Provides a lightweight R interface to the Interactive Brokers (IBKR) Client Portal REST API. Functions cover session management, account and portfolio queries, market data retrieval, and order placement and cancellation. Requires a locally running IBKR Client Portal Gateway.

License MIT + file LICENSE

Depends R (>= 4.1.0)

Encoding UTF-8

Roxygen list(markdown = TRUE)

URL https://github.com/sactyr/ibkrcp

BugReports https://github.com/sactyr/ibkrcp/issues

Suggests httptest2, knitr, rmarkdown, spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

Imports httr2

Config/roxygen2/version 8.0.0

VignetteBuilder knitr

Language en-US

Config/pak/sysreqs libssl-dev

Repository https://sactyr.r-universe.dev

Date/Publication 2026-05-14 07:25:54 UTC

RemoteUrl https://github.com/sactyr/ibkrcp

RemoteRef HEAD

RemoteSha 57eb9be097a6b49c5c129996bf9fa975a5a54d16

Contents

`ibkr_auth_status` *Get the current session authentication status*

Description

Get the current session authentication status

Usage

```
ibkr_auth_status()
```

Value

Named list with session status fields including `authenticated`, `connected`, and `competing`

`ibkr_cancel_order` *Cancel an open order*

Description

Cancel an open order

Usage

```
ibkr_cancel_order(account_id, order_id)
```

Arguments

`account_id`	IBKR account ID string
`order_id`	Order ID to cancel (as returned by `ibkr_live_orders()`)

Value

Invisibly returns the response list

---

ibkr_get_price_history
Get OHLCV price history for an instrument

Description

Fetches daily OHLCV bars from the IBKR Client Portal API. The `t` timestamp field (Unix milliseconds) is converted to an `Australia/Sydney` date, which correctly handles the UTC offset for ASX bar open times (~09:59 AEST = 23:59 UTC the prior day).

Usage

```
ibkr_get_price_history(conid, period = "1y")
```

Arguments

`conid`	Integer conid of the instrument
`period`	History period string. Valid values: `"1m"`, `"3m"`, `"6m"`, `"1y"` (default), `"2y"`, `"3y"`, `"5y"`

Value

Data frame with columns: `date` (Date), `open`, `high`, `low`, `close` (numeric), `volume` (integer), sorted chronologically

---

ibkr_get_trading_schedule
Get the trading schedule for an instrument

Description

Returns the raw IBKR trading schedule for the given symbol and exchange.

Usage

```
ibkr_get_trading_schedule(symbol, exchange, asset_class = "STK")
```

Arguments

`symbol`	Ticker symbol (e.g. `"VGS"`)
`exchange`	Exchange code (e.g. `"ASX"`, `"NYSE"`)
`asset_class`	Asset class (default: `"STK"`)

Value

Raw response list as returned by IBKR. Stops with an error if no schedule is returned.

---

ibkr_live_orders *Get all live and open orders*

Description

Get all live and open orders

Usage

```
ibkr_live_orders(filters = NULL, force = FALSE)
```

Arguments

filters	Optional character vector of order status filters (e.g. `c("Filled", "Cancelled")`)
force	If `TRUE`, forces a fresh fetch bypassing the cache (default: `FALSE`)

Value

Data frame with one row per order and columns: `order_id`, `conid`, `symbol`, `side`, `order_type`, `quantity`, `status`. Returns an empty data frame if no open orders exist.

---

ibkr_ping *Ping the session to confirm it is alive*

Description

Should be called at the start of each trading session to verify the Client Portal Gateway is running and the session is authenticated. Sessions time out after approximately 5 minutes without a request.

Usage

```
ibkr_ping()
```

Value

Invisibly returns the response list

`ibkr_place_order` *Place a market order*

Description

Places a DAY market order for a single instrument. Confirmation prompts returned by the API (e.g. price deviation warnings) are handled automatically via `ibkr_reply_order()`.

Usage

```
ibkr_place_order(account_id, conid, side, quantity)
```

Arguments

`account_id`	IBKR account ID string (e.g. `"U1234567"`)
`conid`	Integer conid of the instrument
`side`	`"BUY"` or `"SELL"`
`quantity`	Number of shares (positive integer)

Value

Invisibly returns the final order response list

`ibkr_portfolio_accounts` *Get all accounts associated with the authenticated user*

Description

Get all accounts associated with the authenticated user

Usage

```
ibkr_portfolio_accounts()
```

Value

Data frame with one row per account and columns: `account_id`, `type`, `currency`, `alias`

ibkr_portfolio_positions

Get current positions for an account

Description

Get current positions for an account

Usage

```
ibkr_portfolio_positions(account_id, sort = "position", direction = "a")
```

Arguments

account_id	IBKR account ID string
sort	Field to sort by (default: "position"). Other valid values: "conid", "contractDesc", "mktValue", "unrealizedPnl"
direction	Sort direction: "a" for ascending (default), "d" for descending

Value

Data frame with one row per position and columns: conid, symbol, position, mkt_price, mkt_value, avg_cost, unrealised_pnl, currency. Returns an empty data frame if no positions are open.

ibkr_portfolio_summary

Get portfolio summary for an account

Description

Returns cash balances and other high-level account metrics as a named list. Each element corresponds to a summary field returned by the IBKR API (e.g. totalcashvalue, netliquidation, availablefunds).

Usage

```
ibkr_portfolio_summary(account_id)
```

Arguments

account_id	IBKR account ID string (e.g. "U1234567")

Value

Named list of summary fields. Each field is itself a list containing amount, currency, and isNull

ibkr_search_contracts *Search for contracts by symbol*

Description

Queries the IBKR contract search endpoint and returns all matching results. Use this to look up conids (IBKR contract IDs) before placing orders or fetching price history.

Usage

```
ibkr_search_contracts(symbol, sec_type = "STK")
```

Arguments

`symbol`	Ticker symbol to search for (e.g. `"VGS"`)
`sec_type`	Security type filter (default: `"STK"` for equities and ETFs)

Value

Raw response list as returned by IBKR. Stops with an error if no matches are found.

Index