

# Package ‘ibkrpc’

May 6, 2026

**Title** R Client for the Interactive Brokers Client Portal API

**Version** 0.1.0

**Description** Provides a lightweight R interface to the Interactive Brokers (IBKR) Client Portal REST API <<https://www.interactivebrokers.com/campus/ibkr-api-page/cpapi-v1/>>. 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

**URL** <https://github.com/sactyr/ibkrpc>

**BugReports** <https://github.com/sactyr/ibkrpc/issues>

**Suggests** httpptest2, knitr, rmarkdown, spelling, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Imports** httr2

**Config/roxygen2/version** 8.0.0

**VignetteBuilder** knitr

**Language** en-US

**NeedsCompilation** no

**Author** Nagappan Karuppiah [aut, cre]

**Maintainer** Nagappan Karuppiah <[sactyr@gmail.com](mailto:sactyr@gmail.com)>

**Repository** CRAN

**Date/Publication** 2026-05-06 19:50:02 UTC

## Contents

ibkr_auth_status . . . . .	2
ibkr_cancel_order . . . . .	2

ibkr_get_accounts . . . . .	3
ibkr_get_orders . . . . .	3
ibkr_get_positions . . . . .	3
ibkr_get_price_history . . . . .	4
ibkr_get_summary . . . . .	4
ibkr_get_trading_schedule . . . . .	5
ibkr_place_order . . . . .	5
ibkr_reauthenticate . . . . .	6
ibkr_search_contracts . . . . .	6
ibkr_tickle . . . . .	7

## Index 8

---

ibkr_auth_status	<i>Get the current session authentication status</i>
------------------	--

---

### 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	<i>Cancel an open order</i>
-------------------	-----------------------------

---

### 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_get_orders())

### Value

Invisibly returns the response list

---

ibkr\_get\_accounts      *Get all accounts associated with the authenticated user*

---

**Description**

Get all accounts associated with the authenticated user

**Usage**

```
ibkr_get_accounts()
```

**Value**

Data frame with one row per account and columns: account\_id, type, currency, alias

---

ibkr\_get\_orders      *Get all live and open orders*

---

**Description**

Get all live and open orders

**Usage**

```
ibkr_get_orders()
```

**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\_get\_positions      *Get current positions for an account*

---

**Description**

Get current positions for an account

**Usage**

```
ibkr_get_positions(account_id)
```

**Arguments**

account\_id      IBKR account ID string

**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\_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\_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_get_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\_get\_trading\_schedule

*Get the trading schedule for an instrument*

---

**Description**

Returns the IBKR trading schedule for the given symbol and exchange, including trading and non-trading dates. Non-trading dates are those where the sessions field is empty (public holidays). Weekend dates are not included in the schedule.

**Usage**

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

**Arguments**

symbol	Ticker symbol (default: "VGS")
exchange	Exchange code (default: "ASX")
asset_class	Asset class (default: "STK")

**Value**

Data frame with columns: date (Date), is\_trading (logical), sessions (character, semicolon-separated session open/close times or NA for non-trading days). Template dates (year 2000) are excluded.

---

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_confirm_order_messages()`.

**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\_reauthenticate    *Reauthenticate the session*

---

**Description**

Call this if the session has timed out but the gateway is still running. After calling this function, wait a few seconds then call `ibkr_tickle()` to confirm the session is restored.

**Usage**

```
ibkr_reauthenticate()
```

**Value**

Invisibly returns the response list

---

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. To find ASX-listed contracts, filter the results for rows where `description` or `company_header` contains "ASX".

**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**

Data frame with one row per match and columns: conid, symbol, company\_name, description, company\_header. Returns an empty data frame if no matches are found.

---

ibkr_tickle	<i>Tickle the session to confirm it is alive</i>
-------------	--

---

**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_tickle()
```

**Value**

Invisibly returns the response list. Stops with an error if the session is not authenticated.

# Index

ibkr\_auth\_status, 2  
ibkr\_cancel\_order, 2  
ibkr\_get\_accounts, 3  
ibkr\_get\_orders, 3  
ibkr\_get\_positions, 3  
ibkr\_get\_price\_history, 4  
ibkr\_get\_summary, 4  
ibkr\_get\_trading\_schedule, 5  
ibkr\_place\_order, 5  
ibkr\_reauthenticate, 6  
ibkr\_search\_contracts, 6  
ibkr\_tickle, 7