

# Package ‘ibkrpc’

May 7, 2026

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

**Version** 0.1.1

**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

**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>

**Repository** CRAN

**Date/Publication** 2026-05-07 08:20:02 UTC

## Contents

ibkr_auth_status . . . . .	2
ibkr_cancel_order . . . . .	2
ibkr_get_price_history . . . . .	3
ibkr_get_trading_schedule . . . . .	3

ibkr_live_orders . . . . .	4
ibkr_ping . . . . .	4
ibkr_place_order . . . . .	5
ibkr_portfolio_accounts . . . . .	5
ibkr_portfolio_positions . . . . .	6
ibkr_portfolio_summary . . . . .	6
ibkr_search_contracts . . . . .	7

<b>Index</b>	<b>8</b>
--------------	----------

---

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_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**

<code>conid</code>	Integer conid of the instrument
<code>period</code>	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**

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

**Value**

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

---

ibkr_live_orders	<i>Get all live and open orders</i>
------------------	-------------------------------------

---

**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. <code>c("Filled", "Cancelled")</code> )
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	<i>Ping 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_ping()
```

**Value**

Invisibly returns the response list

---

ibkr_place_order	<i>Place a market order</i>
------------------	-----------------------------

---

**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	<i>Get all accounts associated with the authenticated user</i>
-------------------------	--

---

**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

ibkr\_auth\_status, [2](#)  
ibkr\_cancel\_order, [2](#)  
ibkr\_get\_price\_history, [3](#)  
ibkr\_get\_trading\_schedule, [3](#)  
ibkr\_live\_orders, [4](#)  
ibkr\_ping, [4](#)  
ibkr\_place\_order, [5](#)  
ibkr\_portfolio\_accounts, [5](#)  
ibkr\_portfolio\_positions, [6](#)  
ibkr\_portfolio\_summary, [6](#)  
ibkr\_search\_contracts, [7](#)