

# Package ‘cryptowatchR’

May 8, 2026

**Type** Package

**Title** An API Wrapper for 'Cryptowatch'

**Version** 0.2.0

**Description**

An API wrapper for 'Cryptowatch' to get prices and other information (e.g., volume, trades, order books, bid and ask prices, live quotes, and more) about cryptocurrencies and crypto exchanges. See <<https://docs.cryptowat.ch/rest-api>> for a detailed documentation.

**Imports** httr, jsonlite, lubridate

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**URL** <https://github.com/lorenzbr/cryptowatchR>

**BugReports** <https://github.com/lorenzbr/cryptowatchR/issues>

**NeedsCompilation** no

**Author** Lorenz Brachtendorf [aut, cre] (ORCID:  
<<https://orcid.org/0000-0001-7066-0838>>)

**Maintainer** Lorenz Brachtendorf <[lorenz.brachtendorf@gmx.de](mailto:lorenz.brachtendorf@gmx.de)>

**Repository** CRAN

**Date/Publication** 2021-06-09 21:20:02 UTC

## Contents

get_assets . . . . .	2
get_current_price . . . . .	3
get_exchanges . . . . .	4
get_markets . . . . .	5
get_ohlc . . . . .	6
get_orderbook . . . . .	7
get_orderbook_calculator . . . . .	8
get_orderbook_liquidity . . . . .	10

get_pairs . . . . .	11
get_summary . . . . .	12
get_trades . . . . .	13
markets . . . . .	14

## Index 16

---

get_assets	<i>Get asset details</i>
------------	--------------------------

---

### Description

Get asset information on cryptocurrencies.

### Usage

```
get_assets(asset = NULL, api_key = NULL, allowance = FALSE)
```

### Arguments

asset	A string containing an asset symbol, e.g. <i>btc</i> (optional argument). Run <code>get_assets()</code> to get all available assets.
api_key	A string containing the API key. See <a href="https://docs.cryptowat.ch/rest-api/rate-limit">https://docs.cryptowat.ch/rest-api/rate-limit</a> to learn how to create an account and how to generate an API key.
allowance	A logical (default is FALSE). If TRUE the function returns a list which includes allowance information, i.e. cost of the request, remaining credits and your account name.

### Value

A list or data.frame containing data on assets.

### References

See <https://docs.cryptowat.ch/rest-api> for further information

### See Also

[get\\_markets](#), [get\\_exchanges](#), [get\\_pairs](#)

### Examples

```
## Not run:
# Get all assets available on 'Cryptowatch'
df.assets <- get_assets()
# Bitcoin asset details
asset.btc <- get_assets("btc")

## End(Not run)
```

---

get\_current\_price      *Get current price of cryptocurrencies*

---

### Description

Get current price of cryptocurrencies using the REST API of cryptowat.ch. The route is *price* or *prices* and returns the current price of a given pair or current prices of all pairs. See <https://docs.cryptowat.ch/rest-api/markets/price> for further information.

### Usage

```
get_current_price(pair, exchange = "kraken", api_key = NULL, allowance = FALSE)
```

### Arguments

pair	A string containing a pair symbol, e.g. <i>btccusd</i> (required argument). Run <code>get_pairs()</code> to find other available pairs.
exchange	A string containing the exchange. Default is <i>kraken</i> . Run <code>get_exchanges()</code> to find other available exchanges.
api_key	A string containing the API key. See <a href="https://docs.cryptowat.ch/rest-api/rate-limit">https://docs.cryptowat.ch/rest-api/rate-limit</a> to learn how to create an account and how to generate an API key.
allowance	A logical (default is FALSE). If TRUE the function returns a list which includes allowance information, i.e. cost of the request, remaining credits and your account name.

### Value

Current price of a given pair of currencies. If allowance is TRUE, `get_current_price()` returns a list.

### References

See <https://docs.cryptowat.ch/rest-api> for further information.

### See Also

[get\\_markets](#), [get\\_ohlcv](#), [get\\_exchanges](#), [get\\_pairs](#)

### Examples

```
## Not run:  
# Daily prices of Bitcoin in USD  
current.price <- get_current_price("btccusd")  
current.prices <- get_current_price()  
  
## End(Not run)
```

---

`get_exchanges`*Get exchange details*

---

### Description

Get information on exchanges.

### Usage

```
get_exchanges(exchange = NULL, api_key = NULL, allowance = FALSE)
```

### Arguments

<code>exchange</code>	A string containing an exchange symbol, e.g. <i>kraken</i> (optional argument). Run <code>get_exchanges()</code> to get a list of exchanges.
<code>api_key</code>	A string containing the API key. See <a href="https://docs.cryptowat.ch/rest-api/rate-limit">https://docs.cryptowat.ch/rest-api/rate-limit</a> to learn how to create an account and how to generate an API key.
<code>allowance</code>	A logical (default is FALSE). If TRUE the function returns a list which includes allowance information, i.e. cost of the request, remaining credits and your account name.

### Value

A list or `data.frame` containing data on exchanges.

### References

See <https://docs.cryptowat.ch/rest-api> for further information

### See Also

[get\\_markets](#), [get\\_assets](#), [get\\_pairs](#)

### Examples

```
## Not run:  
# Get all available exchanges  
df.exchanges <- get_exchanges()  
# Get information on the exchange Kraken  
exchange.kraken <- get_exchanges("kraken")  
  
## End(Not run)
```

---

get_markets	<i>Get prices of cryptocurrencies</i>
-------------	---------------------------------------

---

### Description

Get prices of cryptocurrencies using the REST API of cryptowat.ch.

### Usage

```
get_markets(route, pair = NULL, exchange = NULL, params = NULL,  
            api_key = NULL, allowance = FALSE)
```

### Arguments

route	A character string containing a market endpoint. Possible values: <i>price</i> , <i>prices</i> , <i>trades</i> , <i>summary</i> , <i>summaries</i> , <i>orderbook</i> , <i>orderbook/liquidity</i> , <i>orderbooks/calculator</i> , <i>ohlc</i> (required argument). See <a href="https://docs.cryptowat.ch/rest-api/markets">https://docs.cryptowat.ch/rest-api/markets</a> for further information.
pair	A character string containing a pair symbol, e.g. <i>btcusd</i> (optional argument). Run <code>get_pairs()</code> to find other available pairs.
exchange	A character string containing the exchange (optional argument). Run <code>get_exchanges()</code> to find other available exchanges.
params	A list containing query parameters. E.g., for the route <i>ohlc</i> , this is <i>before</i> , <i>after</i> and <i>periods</i> (optional). See <a href="https://docs.cryptowat.ch/rest-api/markets">https://docs.cryptowat.ch/rest-api/markets</a> for further information.
api_key	A character string containing the API key. See <a href="https://docs.cryptowat.ch/rest-api/rate-limit">https://docs.cryptowat.ch/rest-api/rate-limit</a> to learn how to create an account and how to generate an API key.
allowance	A logical (default is FALSE). If TRUE the function returns a list which includes allowance information, i.e. cost of the request, remaining credits and your account name.

### Value

A list containing markets data.

### References

See <https://docs.cryptowat.ch/rest-api> for further information

### See Also

[markets](#), [get\\_assets](#), [get\\_exchanges](#), [get\\_pairs](#)

## Examples

```
## Not run:
# Prices of Bitcoin in USD for all periods
btcusd.ohlc.all <- get_markets(route = "ohlc", pair = "btcusd", exchange = "kraken")
# Hourly prices of Bitcoin in USD for a specific time period
btcusd.ohlc.hourly <- get_markets(route = "ohlc", pair = "btcusd", exchange = "kraken",
                                 list( periods = 3600, before = 1609851600, after = 1609506000))

## End(Not run)
```

---

get\_ohlc

*Get prices of cryptocurrencies*

---

## Description

Get data.frame with prices of cryptocurrencies using the REST API of cryptowat.ch. The route is *ohlc* and returns OHLC candlestick prices. The default is daily prices but can be changed with periods. See <https://docs.cryptowat.ch/rest-api/markets/ohlc> for further information.

## Usage

```
get_ohlc(pair, before = NULL, after = NULL, periods = NULL, exchange = "kraken",
         datetime = TRUE, api_key = NULL, allowance = FALSE)
```

## Arguments

pair	A character string containing a pair symbol, e.g. <i>btcusd</i> (required argument). Run <code>get_pairs()</code> to find other available pairs.
before	An integer if <i>datetime</i> is FALSE and a character string if it is TRUE: Only returns candles opening before this time (optional).
after	An integer if <i>datetime</i> is FALSE and a character string if it is TRUE: Only returns candles opening after this time (optional).
periods	A integer or integer vector. Only return these periods. Periods are measured in seconds (optional). Examples: 60, 180, 108000.
exchange	A character string containing the exchange. Default is <i>kraken</i> . Run <code>get_exchanges()</code> to find other available exchanges.
datetime	A logical. TRUE indicates that datetime type is used. FALSE indicates <i>unix timestamps</i> . Default is TRUE.
api_key	A character string containing the API key. See <a href="https://docs.cryptowat.ch/rest-api/rate-limit">https://docs.cryptowat.ch/rest-api/rate-limit</a> to learn how to create an account and how to generate an API key.
allowance	A logical (default is FALSE). If TRUE the function returns a list which includes allowance information, i.e. cost of the request, remaining credits and your account name.

**Value**

A data.frame containing OHLC candlestick prices of a given pair of currencies. If allowance is TRUE, get\_ohlc() returns a list.

**References**

See <https://docs.cryptowat.ch/rest-api> for further information.

**See Also**

[get\\_markets](#), [get\\_assets](#), [get\\_exchanges](#), [get\\_pairs](#)

**Examples**

```
## Not run:
# Daily prices of Bitcoin in USD
df.ohlc <- get_ohlc("btcusd")
# Hourly prices of Bitcoin in USD for a specific time period
df.ohlc.hourly <- get_ohlc("btcusd", periods = 3600, before = 1609851600,
                           after = 1609506000, datetime = FALSE)
# Daily prices of Bitcoin in Euro for a specific time period
df.ohlc.daily <- get_ohlc("btceur", periods = 86400, before = "2021-05-12",
                           after = "2021-01-01", datetime = TRUE)

## End(Not run)
```

---

get\_orderbook

*Get order book of cryptocurrencies*

---

**Description**

Get the order book of cryptocurrencies using the REST API of cryptowat.ch. The route is *orderbook* and returns two arrays, bids and asks. See <https://docs.cryptowat.ch/rest-api/markets/order-book> for further information.

**Usage**

```
get_orderbook(pair, depth = NULL, span = NULL, limit = NULL,
              exchange = "kraken", api_key = NULL, allowance = FALSE)
```

**Arguments**

pair	A string containing a pair symbol, e.g. <i>btcusd</i> (required argument). Run <code>get_pairs()</code> to find other available pairs.
depth	A number: Only return orders cumulating up to this size (optional).
span	A number: Only return orders within this percentage of the midpoint (optional). Example: 0.5 (meaning 0.5 percent).

limit	An integer limiting the number of orders on each side of the book (optional).
exchange	A character string containing the exchange. Default is <i>kraken</i> . Run <code>get_exchanges()</code> to find other available exchanges.
api_key	A character string containing the API key. See <a href="https://docs.cryptowat.ch/rest-api/rate-limit">https://docs.cryptowat.ch/rest-api/rate-limit</a> to learn how to create an account and how to generate an API key.
allowance	A logical (default is FALSE). If TRUE the function returns a list which includes allowance information, i.e. cost of the request, remaining credits and your account name.

### Value

A list containing an order book of a given pair of currencies. It contains two data.frames for bid and ask prices with columns *Price* and *Amount*. The function also returns the sequence number *seqNum*. If allowance is TRUE, `get_orderbook()` returns a list which additionally includes allowance information.

### References

See <https://docs.cryptowat.ch/rest-api> for further information.

### See Also

[get\\_markets](#), [get\\_orderbook\\_liquidity](#), [get\\_exchanges](#), [get\\_pairs](#)

### Examples

```
## Not run:
# Entire order book of Bitcoin in USD
orderbook <- get_orderbook("btcusd")
# Order book of Bitcoin in USD: only the best bid and best ask, i.e. the spread
orderbook.limit <- get_orderbook("btcusd", limit = 1)
# Order book of Bitcoin in USD for orders within 0.5% of the top of the book
orderbook.span <- get_orderbook("btcusd", span = 0.5)
# Order book of Bitcoin in Euro for orders adding up to 100 BTC on each side
orderbook.depth <- get_orderbook("btceur", depth = 100)

## End(Not run)
```

---

get\_orderbook\_calculator

*Get live quotes from order book*

---

### Description

Get a live quote from the order book for a given buy and sell amount. The route is *orderbook/calculator* and returns two data.frames containing buy and sell projections. See <https://docs.cryptowat.ch/rest-api/markets/order-book> for further information.

## Usage

```
get_orderbook_calculator(pair, amount = NULL, exchange = "kraken",
                          api_key = NULL, allowance = FALSE)
```

## Arguments

pair	A string containing a pair symbol, e.g. <i>btccusd</i> (required argument). Run <code>get_pairs()</code> to find other available pairs.
amount	A number: Amount to buy or sell (required).
exchange	A character string containing the exchange. Default is <i>kraken</i> . Run <code>get_exchanges()</code> to find other available exchanges.
api_key	A character string containing the API key. See <a href="https://docs.cryptowat.ch/rest-api/rate-limit">https://docs.cryptowat.ch/rest-api/rate-limit</a> to learn how to create an account and how to generate an API key.
allowance	A logical (default is FALSE). If TRUE the function returns a list which includes allowance information, i.e. cost of the request, remaining credits and your account name.

## Value

A list containing projections for the buy and sell side for a given amount. *Reach* indicates the farthest your reach would fill. *Avg* indicates the volume-weighted average. If `allowance` is TRUE, `get_orderbook_calculator()` returns a list which additionally includes allowance information.

## References

See <https://docs.cryptowat.ch/rest-api> for further information.

## See Also

[get\\_markets](#), [get\\_orderbook](#), [get\\_orderbook\\_liquidity](#), [get\\_exchanges](#), [get\\_pairs](#)

## Examples

```
## Not run:
# Live quote for 50 Bitcoins
calculator <- get_orderbook_calculator("btccusd", amount = 50)

## End(Not run)
```

---

`get_orderbook_liquidity`*Get liquidity sums in the order book of cryptocurrencies*

---

### Description

Get liquidity sums in the order book of cryptocurrencies using the REST API of cryptowat.ch. The route is *orderbook/liquidity*. See <https://docs.cryptowat.ch/rest-api/markets/order-book> for further information.

### Usage

```
get_orderbook_liquidity(pair, exchange = "kraken", api_key = NULL, allowance = FALSE)
```

### Arguments

<code>pair</code>	A character string containing a pair symbol, e.g. <i>btcusd</i> (required argument). Run <code>get_pairs()</code> to find other available pairs.
<code>exchange</code>	A character string containing the exchange. Default is <i>kraken</i> . Run <code>get_exchanges()</code> to find other available exchanges.
<code>api_key</code>	A character string containing the API key. See <a href="https://docs.cryptowat.ch/rest-api/rate-limit">https://docs.cryptowat.ch/rest-api/rate-limit</a> to learn how to create an account and how to generate an API key.
<code>allowance</code>	A logical (default is FALSE). If TRUE the function returns a list which includes allowance information, i.e. cost of the request, remaining credits and your account name.

### Value

A list containing liquidity sums at several basis point levels in the order book. If `allowance` is TRUE, `get_orderbook()` returns a list which additionally includes allowance information.

### References

See <https://docs.cryptowat.ch/rest-api> for further information.

### See Also

[get\\_markets](#), [get\\_orderbook](#), [get\\_exchanges](#), [get\\_pairs](#)

### Examples

```
## Not run:  
# Get liquidity sums in the order book of Bitcoin in USD  
liquidity <- get_orderbook_liquidity("btcusd")  
  
## End(Not run)
```

---

get_pairs	<i>Get pair of currencies</i>
-----------	-------------------------------

---

### Description

Get details on pairs of (crypto)currencies.

### Usage

```
get_pairs(pair = NULL, api_key = NULL, allowance = FALSE)
```

### Arguments

pair	A string containing a pair symbol, e.g. <i>btcusd</i> (optional argument). Run <code>get_pairs()</code> to get all available pairs of currencies.
api_key	A string containing the API key. See <a href="https://docs.cryptowat.ch/rest-api/rate-limit">https://docs.cryptowat.ch/rest-api/rate-limit</a> to learn how to create an account and how to generate an API key.
allowance	A logical (default is FALSE). If TRUE the function returns a list which includes allowance information, i.e. cost of the request, remaining credits and your account name.

### Value

A list or data.frame containing data on pairs.

### References

See <https://docs.cryptowat.ch/rest-api> for further information

### See Also

[get\\_markets](#), [get\\_assets](#), [get\\_exchanges](#)

### Examples

```
## Not run:  
# Get all available pairs of currencies  
df.pairs <- get_pairs()  
# Get details on the pair Bitcoin-USD  
pair.btcusd <- get_pairs("btcusd")  
  
## End(Not run)
```

---

get\_summary

*Get 24-hour summary of cryptocurrencies*


---

### Description

Get 24-hour summary of cryptocurrencies using the REST API of cryptowat.ch. The route is *summary* or *summaries* and returns the current 24-hour summary of a given pair or a summary of all pairs. See <https://docs.cryptowat.ch/rest-api/markets/summary> for further information.

### Usage

```
get_summary(pair, keyBy = NULL, exchange = "kraken", api_key = NULL, allowance = FALSE)
```

### Arguments

pair	A character string containing a pair symbol, e.g. <i>btccusd</i> (required argument). Run <code>get_pairs()</code> to find other available pairs.
keyBy	A single character string. Possible values are either <i>id</i> or <i>symbols</i> .
exchange	A character string containing the exchange. Default is <i>kraken</i> . Run <code>get_exchanges()</code> to find other available exchanges.
api_key	A character string containing the API key. See <a href="https://docs.cryptowat.ch/rest-api/rate-limit">https://docs.cryptowat.ch/rest-api/rate-limit</a> to learn how to create an account and how to generate an API key.
allowance	A logical (default is FALSE). If TRUE the function returns a list which includes allowance information, i.e. cost of the request, remaining credits and your account name.

### Value

Current 24-hour summary of a given pair of currencies. If allowance is TRUE, `get_summary()` returns a list with additional information on allowances.

### References

See <https://docs.cryptowat.ch/rest-api> for further information.

### See Also

[get\\_markets](#), [get\\_current\\_price](#), [get\\_exchanges](#), [get\\_pairs](#)

### Examples

```
## Not run:
# 24h-hour summary of cryptocurrency pairs
summary <- get_summary("btccusd")
summaries <- get_summary()
summaries2 <- get_summary(keyBy = "id")
```

```
summaries3 <- get_summary(keyBy = "symbols")

## End(Not run)
```

---

get\_trades                      *Get trades of cryptocurrencies*

---

### Description

Get data.frame with trades of cryptocurrencies using the REST API of cryptowat.ch. The route is *trades* and returns trades for a specified market. See <https://docs.cryptowat.ch/rest-api/markets/trades> for further information.

### Usage

```
get_trades(pair, since = NULL, limit = NULL, exchange = "kraken",
           datetime = TRUE, api_key = NULL, allowance = FALSE)
```

### Arguments

pair	A character string containing a pair symbol, e.g. <i>btccusd</i> (required argument). Run <code>get_pairs()</code> to find other available pairs.
since	An integer if <i>datetime</i> is FALSE and a character string if it is TRUE: Limit the response to trades after this date (optional). This can only be used to filter recent trades. Historical trades cannot be retrieved.
limit	An integer: Limit the number of trades (optional). Max: 1000.
exchange	A character string containing the exchange. Default is <i>kraken</i> . Run <code>get_exchanges()</code> to find other available exchanges.
datetime	A logical. TRUE indicates that datetime type is used. FALSE indicates <i>unix timestamp</i> . Default is TRUE.
api_key	A character string containing the API key. See <a href="https://docs.cryptowat.ch/rest-api/rate-limit">https://docs.cryptowat.ch/rest-api/rate-limit</a> to learn how to create an account and how to generate an API key.
allowance	A logical (default is FALSE). If TRUE the function returns a list which includes allowance information, i.e. cost of the request, remaining credits and your account name.

### Value

A data.frame containing trades of a given pair of currencies. If *allowance* is TRUE, `get_trades()` returns a list.

### References

See <https://docs.cryptowat.ch/rest-api> for further information.

**See Also**

[get\\_markets](#), [get\\_assets](#), [get\\_exchanges](#), [get\\_pairs](#)

**Examples**

```
## Not run:
# Most recent trades (default is 50)
trades <- get_trades(pair)
# 200 trades (maximum is 1000) since 1589571417 (unix timestamp)
trades.unix <- get_trades(pair, since = 1589571417, limit = 200, datetime = FALSE)
# 1000 trades and datetime is TRUE
trades.datetime <- get_trades(pair, since = "2021-06-01", limit = 1000)

## End(Not run)
```

---

markets	<i>Get prices of cryptocurrencies</i>
---------	---------------------------------------

---

**Description**

Get data.frame with prices of cryptocurrencies using the REST API of cryptowat.ch.

**Usage**

```
markets(pair, params = NULL, exchange = "kraken", route = "ohlcv", datetime = TRUE)
```

**Arguments**

pair	A string containing a pair symbol, e.g. <i>btccusd</i> (required argument).
params	A list containing before, after and periods (optional). See <a href="https://docs.cryptowat.ch/rest-api/markets/ohlcv">https://docs.cryptowat.ch/rest-api/markets/ohlcv</a> for further information.
exchange	A character string containing the exchange. Default is <i>kraken</i> .
route	A character string containing the route. Default is <i>ohlcv</i> .
datetime	A logical. TRUE indicates that datetime type is used. FALSE indicates <i>unix timestamp</i> . Default is TRUE.

**Value**

A data.frame containing OHLC candlestick prices of a given pair of currencies.

**References**

See <https://docs.cryptowat.ch/rest-api> for further information.

**See Also**

[get\\_markets](#), [get\\_assets](#), [get\\_exchanges](#), [get\\_pairs](#)

**Examples**

```
## Not run:
df.ohlc.hourly <- markets("btcusd")
df.ohlc.hourly2 <- markets("btcusd", list( periods = 3600, before = 1609851600,
                                           after = 1609506000), datetime = FALSE)
df.ohlc.daily <- markets("btceur", list( periods = 86400, before = "2021-05-12",
                                           after = "2021-01-01"), datetime = TRUE)

## End(Not run)
```

# Index

`get_assets`, [2](#), [4](#), [5](#), [7](#), [11](#), [14](#)  
`get_current_price`, [3](#), [12](#)  
`get_exchanges`, [2](#), [3](#), [4](#), [5](#), [7-12](#), [14](#)  
`get_markets`, [2-4](#), [5](#), [7-12](#), [14](#)  
`get_ohlc`, [3](#), [6](#)  
`get_orderbook`, [7](#), [9](#), [10](#)  
`get_orderbook_calculator`, [8](#)  
`get_orderbook_liquidity`, [8](#), [9](#), [10](#)  
`get_pairs`, [2-5](#), [7-10](#), [11](#), [12](#), [14](#)  
`get_summary`, [12](#)  
`get_trades`, [13](#)

`markets`, [5](#), [14](#)