

# Package ‘binancer’

October 12, 2022

**Type** Package

**Title** API Client to 'Binance'

**Version** 1.2.0

**Date** 2021-11-26

**Encoding** UTF-8

**Description** R client to the 'Binance' Public Rest API for data collection on cryptocurrencies, portfolio management and trading: <<https://github.com/binance/binance-spot-api-docs/blob/master/rest-api.md>>.

**License** AGPL

**RoxygenNote** 7.1.1

**Imports** data.table, httr, digest, snakecase, logger, jsonlite

**URL** <https://daroczig.github.io/binancer/>

**NeedsCompilation** no

**Author** Gergely Daróczy [aut, cre],  
David Anel [aut]

**Maintainer** Gergely Daróczy <daroczig@rapporter.net>

**Repository** CRAN

**Date/Publication** 2021-11-29 20:10:09 UTC

## R topics documented:

binance_account . . . . .	2
binance_all_orders . . . . .	2
binance_avg_price . . . . .	3
binance_balances . . . . .	4
binance_cancel_order . . . . .	4
binance_coins . . . . .	5
binance_coins_prices . . . . .	5
binance_credentials . . . . .	6
binance_depth . . . . .	6
binance_exchange_info . . . . .	7

binance_filters . . . . .	7
binance_klines . . . . .	8
binance_mytrades . . . . .	9
binance_new_order . . . . .	9
binance_open_orders . . . . .	11
binance_ping . . . . .	11
binance_query_order . . . . .	12
binance_symbols . . . . .	12
binance_ticker_24hr . . . . .	13
binance_ticker_all_books . . . . .	13
binance_ticker_all_prices . . . . .	14
binance_ticker_book . . . . .	14
binance_ticker_price . . . . .	15
binance_ticks . . . . .	15
binance_time . . . . .	16
binance_trades . . . . .	16

**Index** 17

---

binance_account	<i>Get current general Binance account information, without balances</i>
-----------------	--

---

**Description**

Get current general Binance account information, without balances

**Usage**

binance\_account()

**Value**

data.table

---

binance_all_orders	<i>Fetch all orders from the Binance account</i>
--------------------	--

---

**Description**

Fetch all orders from the Binance account

**Usage**

binance\_all\_orders(symbol, order\_id, start\_time, end\_time, limit)

**Arguments**

symbol	string
order_id	optional number
start_time	optional POSIX timestamp
end_time	optional POSIX timestamp
limit	optional int

**Value**

data.table

**Examples**

```
## Not run:  
binance_all_orders('ARKETH')  
binance_all_orders('ARKBTC', order_id = '123456')  
  
## End(Not run)
```

---

binance\_avg\_price      *Get current average price for a symbol*

---

**Description**

Get current average price for a symbol

**Usage**

```
binance_avg_price(symbol)
```

**Arguments**

symbol	string
--------	--------

**Value**

data.table

---

binance\_balances      *Get current Binance balances in a nice table*

---

**Description**

Get current Binance balances in a nice table

**Usage**

```
binance_balances(threshold = -1, usdt = FALSE)
```

**Arguments**

threshold	optional show assets with greater number of coins
usdt	optional to include balance in USDT too

**Value**

data.table

---

binance\_cancel\_order      *Cancel order on the Binance account*

---

**Description**

Cancel order on the Binance account

**Usage**

```
binance_cancel_order(symbol, order_id, client_order_id)
```

**Arguments**

symbol	string
order_id	optional number
client_order_id	optional string

**Value**

data.table

**Examples**

```
## Not run:  
binance_cancel_order('ARKETH', order_id = 123456)  
binance_cancel_order('ARKBTC', client_order_id = 'myOrder7')  
  
## End(Not run)
```

---

binance_coins	<i>Get all currently valid coin names from Binance</i>
---------------	--

---

**Description**

Get all currently valid coin names from Binance

**Usage**

```
binance_coins()
```

**Value**

character vector of coin names

---

binance_coins_prices	<i>Get all currently valid coin names from Binance along with the USDT prices</i>
----------------------	---

---

**Description**

Get all currently valid coin names from Binance along with the USDT prices

**Usage**

```
binance_coins_prices(unit = "USDT")
```

**Arguments**

unit            to set quote asset

**Value**

data.table with symbol and usd columns

---

binance\_credentials    *Sets the API key and secret to interact with the Binance API*

---

**Description**

Sets the API key and secret to interact with the Binance API

**Usage**

```
binance_credentials(key, secret)
```

**Arguments**

key	string
secret	string

**Value**

No return values, setting config in the package namespace.

**Examples**

```
## Not run:  
binance_credentials('foo', 'bar')  
  
## End(Not run)
```

---

binance\_depth    *Get orderbook depth data from Binance*

---

**Description**

Get orderbook depth data from Binance

**Usage**

```
binance_depth(symbol, limit)
```

**Arguments**

symbol	string
limit	int optional

**Value**

data.table

### Examples

```
## Not run:  
binance_depth('ETHUSD')  
binance_depth('ETHUSD', limit = 1000)  
  
## End(Not run)
```

---

binance\_exchange\_info *Get exchangeInfo from Binance*

---

### Description

Get exchangeInfo from Binance

### Usage

```
binance_exchange_info()
```

### Value

list

---

binance\_filters *Get current filters for a symbol*

---

### Description

Get current filters for a symbol

### Usage

```
binance_filters(symbol)
```

### Arguments

symbol            string

### Value

data.table

---

binance_klines	<i>Get kline/candlestick data from Binance</i>
----------------	--

---

### Description

Get kline/candlestick data from Binance

### Usage

```
binance_klines(  
    symbol,  
    interval = c("1m", "3m", "5m", "15m", "30m", "1h", "2h", "4h", "6h", "8h", "12h",  
                "1d", "3d", "1w", "1M"),  
    limit,  
    start_time,  
    end_time  
)
```

### Arguments

symbol	string
interval	enum
limit	optional int
start_time	optional POSIX timestamp
end_time	optional POSIX timestamp

### Value

data.table with open-high-low-close values

### Examples

```
## Not run:  
binance_klines('ETHUSDT')  
binance_klines('ETHUSDT', interval = '1h', limit = 24*7)  
binance_klines('ETHUSDT', interval = '1h', start_time = '2018-01-01', end_time = '2018-01-08')  
  
## End(Not run)
```



---

binance_mytrades	<i>Get trades for a specific symbol on the Binance account</i>
------------------	--

---

**Description**

Get trades for a specific symbol on the Binance account

**Usage**

```
binance_mytrades(symbol, limit, from_id, start_time, end_time)
```

**Arguments**

symbol	string
limit	optional int number of trades to fetch
from_id	optional trade id to fetch from
start_time	optional POSIX timestamp
end_time	optional POSIX timestamp

**Value**

data.table

**Examples**

```
## Not run:  
binance_mytrades('ARKETH')  
binance_mytrades(c('ARKBTC', 'ARKETH'))  
  
## End(Not run)
```

---

binance_new_order	<i>Open new order on the Binance account</i>
-------------------	--

---

**Description**

Open new order on the Binance account

**Usage**

```
binance_new_order(  
  symbol,  
  side = c("BUY", "SELL"),  
  type = c("LIMIT", "MARKET", "STOP_LOSS", "STOP_LOSS_LIMIT", "TAKE_PROFIT",  
    "TAKE_PROFIT_LIMIT", "LIMIT_MAKER"),  
  time_in_force = c("GTC", "IOC", "FOK"),  
  quantity,  
  price,  
  stop_price,  
  iceberg_qty,  
  test = TRUE  
)
```

**Arguments**

symbol	string
side	enum
type	enum
time_in_force	optional enum
quantity	number
price	optional number
stop_price	optional number
iceberg_qty	optional number
test	bool

**Value**

data.table

**Examples**

```
## Not run:  
binance_new_order('ARKETH', side = 'BUY', type = 'MARKET', quantity = 1)  
binance_new_order('ARKBTC', side = 'BUY', type = 'LIMIT', quantity = 1,  
  price = 0.5, time_in_force = 'GTC')  
  
## End(Not run)
```

---

binance\_open\_orders     *Fetch open orders from the Binance account*

---

**Description**

Fetch open orders from the Binance account

**Usage**

```
binance_open_orders(symbol)
```

**Arguments**

symbol            optional string

**Value**

data.table

**Examples**

```
## Not run:  
binance_open_orders('ARKETH')  
binance_open_orders() # all symbols - binance.weight 40  
  
## End(Not run)
```

---

binance\_ping            *Test connectivity to the Rest API*

---

**Description**

Test connectivity to the Rest API

**Usage**

```
binance_ping()
```

**Value**

"OK" string on success

---

binance\_query\_order     *Query order on the Binance account*

---

**Description**

Query order on the Binance account

**Usage**

```
binance_query_order(symbol, order_id, client_order_id)
```

**Arguments**

```
symbol           string
order_id         optional number
client_order_id  optional string
```

**Value**

data.table

**Examples**

```
## Not run:
binance_query_order('ARKETH')
binance_query_order('ARKBTC', client_order_id = 'myOrder7')

## End(Not run)
```

---

binance\_symbols     *Get all currently valid symbol names from Binance*

---

**Description**

Get all currently valid symbol names from Binance

**Usage**

```
binance_symbols(all = FALSE)
```

**Arguments**

```
all                optional bool include non-trading symbols
```

**Value**

character vector of symbol names

---

binance\_ticker\_24hr    *24 hour rolling window price change statistics*

---

**Description**

24 hour rolling window price change statistics

**Usage**

```
binance_ticker_24hr(symbol)
```

**Arguments**

symbol            optional string

**Value**

data.table

**Examples**

```
## Not run:  
binance_ticker_24hr('ARKETH')  
binance_ticker_24hr() # all symbols - binance.weight 40  
  
## End(Not run)
```

---

binance\_ticker\_all\_books  
*Get latest Binance bids and asks on all symbol pairs*

---

**Description**

Get latest Binance bids and asks on all symbol pairs

**Usage**

```
binance_ticker_all_books()
```

**Value**

data.table

binance\_ticker\_all\_prices

*Get latest Binance conversion rates and USD prices on all symbol pairs*

---

**Description**

Get latest Binance conversion rates and USD prices on all symbol pairs

**Usage**

```
binance_ticker_all_prices()
```

**Value**

data.table

---

binance\_ticker\_book *Get last bids and asks for a symbol or all symbols*

---

**Description**

Get last bids and asks for a symbol or all symbols

**Usage**

```
binance_ticker_book(symbol)
```

**Arguments**

symbol            optional string

**Value**

data.table

---

binance\_ticker\_price *Get last price for a symbol or all symbols*

---

**Description**

Get last price for a symbol or all symbols

**Usage**

```
binance_ticker_price(symbol)
```

**Arguments**

symbol            optional string

**Value**

data.table

---

binance\_ticks            *Get tick data from Binance*

---

**Description**

Get tick data from Binance

**Usage**

```
binance_ticks(symbol, from_id, start_time, end_time, limit)
```

**Arguments**

symbol            string  
from\_id           optional number  
start\_time        optional POSIX timestamp  
end\_time          optional POSIX timestamp  
limit             optional int

**Value**

data.table

**Examples**

```
## Not run:
binance_ticks('ETHUSDT')
binance_ticks('ETHUSDT', start_time = '2018-01-01 00:00:00', end_time = '2018-01-01 01:00:00')

## End(Not run)
```

---

binance_time	<i>Get the current server time from Binance</i>
--------------	---

---

**Description**

Get the current server time from Binance

**Usage**

```
binance_time()
```

**Value**

POSIXct

---

binance_trades	<i>Get last trades from Binance</i>
----------------	-------------------------------------

---

**Description**

Get last trades from Binance

**Usage**

```
binance_trades(symbol, limit)
```

**Arguments**

symbol	string
limit	optional int

**Value**

data.table

**Examples**

```
## Not run:
binance_trades('ETHUSDT')
binance_trades('ETHUSDT', limit = 1000)

## End(Not run)
```



# Index

binance\_account, 2  
binance\_all\_orders, 2  
binance\_avg\_price, 3  
binance\_balances, 4  
binance\_cancel\_order, 4  
binance\_coins, 5  
binance\_coins\_prices, 5  
binance\_credentials, 6  
binance\_depth, 6  
binance\_exchange\_info, 7  
binance\_filters, 7  
binance\_klines, 8  
binance\_mytrades, 9  
binance\_new\_order, 9  
binance\_open\_orders, 11  
binance\_ping, 11  
binance\_query\_order, 12  
binance\_symbols, 12  
binance\_ticker\_24hr, 13  
binance\_ticker\_all\_books, 13  
binance\_ticker\_all\_prices, 14  
binance\_ticker\_book, 14  
binance\_ticker\_price, 15  
binance\_ticks, 15  
binance\_time, 16  
binance\_trades, 16