
python-allcoin Documentation

Release 0.2.0

Sam McHardy

Mar 07, 2018

Contents

1	Features	3
2	Quick Start	5
3	Donate	7
4	Other Exchanges	9
4.1	Contents	9
4.2	Index	30
	Python Module Index	31

This is an unofficial Python wrapper for the Allcoin exchanges REST API v1. I am in no way affiliated with Allcoin, use at your own risk.

PyPi <https://pypi.python.org/pypi/python-allcoin>

Source code <https://github.com/sammchardy/python-allcoin>

Documentation <https://python-allcoin.readthedocs.io/en/latest/>

Blog with examples <https://sammchardy.github.io>

CHAPTER 1

Features

- Implementation of all REST endpoints
- Simple handling of authentication
- Response exception handling

Register an account with [Allcoin](#).

To use signed account methods you are required to verify to create a Transaction Password and then [create an API Key](#) and apply appropriate permissions.

```
pip install python-allcoin
```

```
from allcoin.client import Client
client = Client(api_key, api_secret)

# get market depth
depth = client.get_order_book('eth_btc', size=50)

# get symbol klines
klines = client.get_klines('eth_btc', '1day')

# place a buy order
transaction = client.create_buy_order('eth_btc', '0.01', '1000')

# get list of open orders
orders = client.get_open_orders('eth_btc')

# cancel an order
orders = client.cancel_order('eth_btc', '1235')

# get order info
orders = client.get_order('eth_btc', '1235')
```

For more check out the [documentation](#).

CHAPTER 3

Donate

If this library helped you out feel free to donate.

- ETH: 0xD7a7fDdCfA687073d7cC93E9E51829a727f9fE70
- NEO: AVJB4ZgN7VgSUtArCt94y7ZYT6d5NDfpBo
- LTC: LPC5vw9ajR1YndE1hYVeo3kJ9LdHjcRCUZ
- BTC: 1Dknp6L6oRZrHDECRedihPzx2sSfmvEBys

If you use [Binance](#) check out my [python-binance](#) library.

If you use [Kucoin](#) check out my [python-kucoin](#) library.

If you use [Quoinex](#) or [Qryptos](#) check out my [python-quoine](#) library.

If you use [IDEX](#) check out my [python-idex](#) library.

If you use [BigONE](#) check out my [python-bigone](#) library.

4.1 Contents

4.1.1 Getting Started

Installation

`python-allcoin` is available on [PYPI](#). Install with `pip`:

```
pip install python-allcoin
```

Register on Allcoin

Firstly register an account with [Allcoin](#).

Generate an API Key

To use signed account methods you are required to verify to create a Transaction Password and then [create an API Key](#) and apply appropriate permissions.

Initialise the client

Pass your API Key and Secret

```
from allcoin.client import Client
client = Client(api_key, api_secret)
```

API Rate Limit

Unknown

4.1.2 User Endpoints

class allcoin.client.**Client** (*api_key, api_secret, requests_params=None*)

get_userinfo()
Get account info

```
# default
info = client.get_userinfo()
```

Returns API response

```
{
  "info": {
    "funds": {
      "free": {
        "btc": "0",
        "usd": "0",
        "ltc": "0"
      },
      "freezed": {
        "btc": "0",
        "usd": "0",
        "ltc": "0"
      }
    }
  },
  "result": true
}
```

Raises AllcoinResponseException, BinanceAPIException

4.1.3 Trading Endpoints

class allcoin.client.**Client** (*api_key, api_secret, requests_params=None*)

batch_orders (*symbol, order_data, order_type=None*)
Create an order

Parameters

- **symbol** (*str*) – required
- **order_data** (*list of dicts*) – list of dictionaries of price, amount and type
- **order_type** (*str*) – optional buy/sell default used if type not set in order_data dict

```
order_data = [
    {
        "price": "0.0123",
        "amount": "120",
        "type": "sell"
    },
    {
        "price": "0.0112",
        "amount": "110"
    }
]
order = client.create_order('eth_btc', order_data, type='buy')
```

Returns API response

```
{
  "order_info": [
    {"order_id": 41724206},
    {"error_code": 10011, "order_id": -1},
    {"error_code": 10014, "order_id": -1}
  ],
  "result": true
}
```

Raises AllcoinResponseException, BinanceAPIException

cancel_order (*symbol, order_id*)

Cancel an order or up to 3 orders

Parameters

- **symbol** (*str*) – required
- **order_id** (*str*) – order ID (multiple orders are separated by a comma ‘,’; 3 orders at most are allowed per request)

```
# single order
order = client.cancel_order('eth_btc', '123456')

# multiple orders
order = client.cancel_order('eth_btc', '123456,123457,123557')
```

Returns API response

```
# single order id
{
  "order_id": "123456",
  "result": true
}

# multiple order ids
{
```

```
"success": "123456,123457",
"error": "123458"
}
```

Raises AllcoinResponseException, BinanceAPIException

create_buy_order (*symbol, price, amount*)

Create a buy order

Parameters

- **symbol** (*str*) – required
- **price** (*str*) – order price
- **amount** (*str*) – order quantity

```
# default
order = client.create_buy_order('eth_btc', '0.2348', '100')
```

Returns API response

```
{
  "order_id": "123456",
  "result": true
}
```

Raises AllcoinResponseException, BinanceAPIException

create_order (*symbol, side, price, amount*)

Create an order

Parameters

- **symbol** (*str*) – required
- **side** (*str*) – order side buy/sell
- **price** (*str*) – order price
- **amount** (*str*) – order quantity

```
# default
order = client.create_order('eth_btc', 'buy', '0.2348', '100')
```

Returns API response

```
{
  "order_id": "123456",
  "result": true
}
```

Raises AllcoinResponseException, BinanceAPIException

create_sell_order (*symbol, price, amount*)

Create a sell order

Parameters

- **symbol** (*str*) – required
- **price** (*str*) – order price
- **amount** (*str*) – order quantity

```
# default
order = client.create_buy_order('eth_btc', '0.2348', '100')
```

Returns API response

```
{
  "order_id": "123456",
  "result": true
}
```

Raises AllcoinResponseException, BinanceAPIException

get_open_orders (*symbol*)

Get info about all open orders

Parameters **symbol** (*str*) – required

```
order = client.get_open_orders('eth_btc')
```

Returns API response

```
{
  "result": true,
  "orders": [
    {
      "amount": 0.1,
      "avg_price": 0,           # average transaction price
      "create_date": 1418008467000, # order time
      "deal_amount": 0,        # filled quantity
      "order_id": 10000591,    # order id
      "price": 500,           # entrustment price
      "status": 0,           # 0 = unfilled, 1 = partially_
      ←filled, 2 = fully filled, 10 = cancelled
      "symbol": "btc_usd",
      "type": "sell"
    },
    {
      "amount": 0.2,
      "avg_price": 0,
      "create_date": 1417417957000,
      "deal_amount": 0,
      "order_id": 10000724,
      "price": 0.1,
      "status": 0,
      "symbol": "btc_usd",
      "type": "buy"
    }
  ]
}
```

Raises AllcoinResponseException, BinanceAPIException

get_order (*symbol*, *order_id*)

Get info about a particular order

Parameters

- **symbol** (*str*) – required
- **order_id** (*str*) – order ID to fetch

```
order = client.get_order('eth_btc', '123456')
```

Returns API response

```
{
  "result": true,
  "orders": [
    {
      "amount": 0.1,
      "avg_price": 0,
      "create_date": 1418008467000,
      "deal_amount": 0,
      "order_id": 10000591,
      "price": 500,
      "status": 0,
      "symbol": "btc_usd",
      "type": "sell"
    },
    {
      "amount": 0.2,
      "avg_price": 0,
      "create_date": 1417417957000,
      "deal_amount": 0,
      "order_id": 10000724,
      "price": 0.1,
      "status": 0,
      "symbol": "btc_usd",
      "type": "buy"
    }
  ]
}
```

average transaction price
order time
filled quantity
order id
entrustment price
0 = unfilled, 1 = partially_
filled, 2 = fully filled, 10 = cancelled

Raises AllcoinResponseException, BinanceAPIException

get_order_history (*symbol*, *order_status*, *page=1*, *limit=200*)

Get history of orders for a symbol

Parameters

- **symbol** (*str*) – required
- **order_status** (*int*) – 0 for unfilled orders; 1 for filled orders
- **page** (*int*) – page to fetch
- **limit** (*int*) – amount on each page

```
order = client.get_order_history('eth_btc', 1)
```

Returns API response

```
[
  {
    "current_page": 1,
    "orders": [
      {
        "amount": 0,
        "avg_price": 0,
        "create_date": 1405562100000,
        "deal_amount": 0,
        "order_id": 0,
        "price": 0,
        "status": 2,
        "symbol": "btc_usd",
        "type": "sell"
      }
    ],
    "page_length": 1,
    "result": true,
    "total": 3
  }
]
```

Raises AllcoinResponseException, BinanceAPIException

get_orders (*symbol, order_status, order_ids*)

Get info about multiple orders

Parameters

- **symbol** (*str*) – required
- **order_status** (*int*) – 0 for unfilled orders; 1 for filled orders
- **order_ids** (*str*) – order IDs to fetch (multiple orders are separated by ‘,’, 50 orders at most are allowed per request)

```
order = client.get_orders('eth_btc', 1)
```

Returns API response

```
{
  "result": true,
  "orders": [
    {
      "amount": 0.1,
      "avg_price": 0,
      "create_date": 1418008467000,
      "deal_amount": 0,
      "order_id": 10000591,
      "price": 500,
      "status": 0,
      "symbol": "btc_usd",
      "type": "sell"
    },
    {
      "amount": 0.2,
      "avg_price": 0,
```

```

        "create_date": 1417417957000,
        "deal_amount": 0,
        "order_id": 10000724,
        "price": 0.1,
        "status": 0,
        "symbol": "btc_usd",
        "type": "buy"
    }
]
}

```

Raises AllcoinResponseException, BinanceAPIException

get_trade_history (*symbol*, *since=None*)

Get trade history - requires api key

Parameters

- **symbol** (*str*) – required
- **since** (*int*) – Transaction id (inclusive)

```

# default
trades = client.get_trade_history('eth_btc')

# default
trades = client.get_trade_history('eth_btc', since=230433)

```

Returns API response

```

[
  {
    "date": 1367130137,
    "date_ms": 1367130137000,
    "price": 787.71,
    "amount": 0.003,
    "tid": "230433",
    "type": "sell"
  },
  {
    "date": 1367130137,
    "date_ms": 1367130137000,
    "price": 787.65,
    "amount": 0.001,
    "tid": "230434",
    "type": "sell"
  },
  {
    "date": 1367130137,
    "date_ms": 1367130137000,
    "price": 787.5,
    "amount": 0.091,
    "tid": "230435",
    "type": "sell"
  }
]

```

Raises AllcoinResponseException, BinanceAPIException

4.1.4 Market Endpoints

class allcoin.client.**Client** (*api_key, api_secret, requests_params=None*)

get_klines (*symbol, kline_type, size=None, since=None*)
Get klines for a symbol

Parameters

- **symbol** (*str*) – required
- **kline_type** (*str*) – type of candlestick (1min, 3min, 5min, 15min, 30min, 1hour, 2hour, 4hour, 6hour, 12hour, 1day, 3day, 1week)
- **size** (*int*) – number of klines to return
- **since** (*int*) – timestamp in ms to return from

```
# default
klines = client.get_klines('eth_btc', '1min')

# optional params
klines = client.get_klines('eth_btc', '1hour', size=20, since=1417449600000)
```

Returns API response

```
[
  [
    1417449600000, # timestamp
    2339.11,      # open
    2383.15,      # high
    2322,         # low
    2369.85,      # close
    83850.06     # volume
  ],
  [
    1417536000000,
    2370.16,
    2380,
    2352,
    2367.37,
    17259.83
  ]
]
```

Raises AllcoinResponseException, BinanceAPIException

get_order_book (*symbol, size=None, merge=None*)
Get the Order Book for the market

Parameters

- **symbol** (*str*) – required
- **size** (*int*) – Default 100; max 100

- **merge** (*int*) – merge depth Default 1; max 100

```
# default
book = client.get_order_book('eth_btc')

# optional params
book = client.get_order_book('eth_btc', size=5, merge=5)
```

Returns API response

```
{
  "asks": [
    [792, 5],
    [789.68, 0.018],
    [788.99, 0.042],
    [788.43, 0.036],
    [787.27, 0.02]
  ],
  "bids": [
    [787.1, 0.35],
    [787, 12.071],
    [786.5, 0.014],
    [786.2, 0.38],
    [786, 3.217],
    [785.3, 5.322],
    [785.04, 5.04]
  ]
}
```

Raises AllcoinResponseException, BinanceAPIException

get_ticker (*symbol*)

Get the Ticker for the market

Parameters **symbol** (*str*) – required

```
ticker = client.get_ticker('eth_btc')
```

Returns API response

```
{
  "date": "1410431279",
  "ticker": {
    "buy": "33.15",
    "high": "34.15",
    "last": "33.15",
    "low": "32.05",
    "sell": "33.16",
    "vol": "10532696.39199642"
  }
}
```

Raises AllcoinResponseException, BinanceAPIException

get_trades (*symbol, since=None*)

Get the last 600 trades with optional since transaction id parameter

Parameters

- **symbol** (*str*) – required
- **since** (*int*) – Transaction id (inclusive)

```
# default
trades = client.get_trades('eth_btc')

# default
trades = client.get_trades('eth_btc', since=230433)
```

Returns API response

```
[
  {
    "date": "1367130137",
    "date_ms": "1367130137000",
    "price": 787.71,
    "amount": 0.003,
    "tid": "230433",
    "type": "sell"
  },
  {
    "date": "1367130137",
    "date_ms": "1367130137000",
    "price": 787.65,
    "amount": 0.001,
    "tid": "230434",
    "type": "sell"
  },
  {
    "date": "1367130137",
    "date_ms": "1367130137000",
    "price": 787.5,
    "amount": 0.091,
    "tid": "230435",
    "type": "sell"
  }
]
```

Raises AllcoinResponseException, BinanceAPIException

4.1.5 Exceptions

AllcoinResponseException

Raised if a non JSON response is returned

AllcoinAPIException

On an API call error a allcoin.exceptions.AllcoinAPIException will be raised.

The exception provides access to the

- *status_code* - response status code

- *response* - response object
- *code* - Allcoin error code
- *message* - Allcoin error message
- *request* - request object if available

```
try:
    client.get_ticker('eth_btc')
except AllcoinAPIException as e:
    print(e.status_code)
    print(e.code)
    print(e.message)
```

4.1.6 Changelog

v0.0.1 - 2018-03-02

Added

- Allcoin client interface
- Coverage for all main endpoints
- Constants for transfer type and status, order side and kline resolution
- Full documentation

4.1.7 Allcoin API

client module

```
class allcoin.client.Client (api_key, api_secret, requests_params=None)
```

Bases: object

```
API_URL = 'https://api.allcoin.com/api'
```

```
API_VERSION = 'v1'
```

```
ORDER_STATUS_UNFILLED = 0
```

```
ORDER_STATUS_PARTIALLY_FILLED = 1
```

```
ORDER_STATUS_FILLED = 2
```

```
ORDER_STATUS_CANCELLED = 10
```

```
__init__(api_key, api_secret, requests_params=None)
```

Allcoin API Client constructor

Parameters

- **api_key** (*str.*) – Api Key
- **api_secret** (*str.*) – Api Secret
- **requests_params** (*dict.*) – optional - Dictionary of requests params to use for all calls

```
get_ticker (symbol)
```

Get the Ticker for the market

Parameters `symbol` (*str*) – required

```
ticker = client.get_ticker('eth_btc')
```

Returns API response

```
{
  "date": "1410431279",
  "ticker": {
    "buy": "33.15",
    "high": "34.15",
    "last": "33.15",
    "low": "32.05",
    "sell": "33.16",
    "vol": "10532696.39199642"
  }
}
```

Raises AllcoinResponseException, BinanceAPIException

get_order_book (*symbol*, *size=None*, *merge=None*)

Get the Order Book for the market

Parameters

- **symbol** (*str*) – required
- **size** (*int*) – Default 100; max 100
- **merge** (*int*) – merge depth Default 1; max 100

```
# default
book = client.get_order_book('eth_btc')

# optional params
book = client.get_order_book('eth_btc', size=5, merge=5)
```

Returns API response

```
{
  "asks": [
    [792, 5],
    [789.68, 0.018],
    [788.99, 0.042],
    [788.43, 0.036],
    [787.27, 0.02]
  ],
  "bids": [
    [787.1, 0.35],
    [787, 12.071],
    [786.5, 0.014],
    [786.2, 0.38],
    [786, 3.217],
    [785.3, 5.322],
    [785.04, 5.04]
  ]
}
```

Raises AllcoinResponseException, BinanceAPIException

get_trades (*symbol*, *since=None*)

Get the last 600 trades with optional since transaction id parameter

Parameters

- **symbol** (*str*) – required
- **since** (*int*) – Transaction id (inclusive)

```
# default
trades = client.get_trades('eth_btc')

# default
trades = client.get_trades('eth_btc', since=230433)
```

Returns API response

```
[
  {
    "date": "1367130137",
    "date_ms": "1367130137000",
    "price": 787.71,
    "amount": 0.003,
    "tid": "230433",
    "type": "sell"
  },
  {
    "date": "1367130137",
    "date_ms": "1367130137000",
    "price": 787.65,
    "amount": 0.001,
    "tid": "230434",
    "type": "sell"
  },
  {
    "date": "1367130137",
    "date_ms": "1367130137000",
    "price": 787.5,
    "amount": 0.091,
    "tid": "230435",
    "type": "sell"
  }
]
```

Raises AllcoinResponseException, BinanceAPIException

get_trade_history (*symbol*, *since=None*)

Get trade history - requires api key

Parameters

- **symbol** (*str*) – required
- **since** (*int*) – Transaction id (inclusive)

```
# default
trades = client.get_trade_history('eth_btc')

# default
trades = client.get_trade_history('eth_btc', since=230433)
```

Returns API response

```
[
  {
    "date": 1367130137,
    "date_ms": 1367130137000,
    "price": 787.71,
    "amount": 0.003,
    "tid": "230433",
    "type": "sell"
  },
  {
    "date": 1367130137,
    "date_ms": 1367130137000,
    "price": 787.65,
    "amount": 0.001,
    "tid": "230434",
    "type": "sell"
  },
  {
    "date": 1367130137,
    "date_ms": 1367130137000,
    "price": 787.5,
    "amount": 0.091,
    "tid": "230435",
    "type": "sell"
  }
]
```

Raises AllcoinResponseException, BinanceAPIException

get_klines (*symbol*, *kline_type*, *size=None*, *since=None*)

Get klines for a symbol

Parameters

- **symbol** (*str*) – required
- **kline_type** (*str*) – type of candlestick (1min, 3min, 5min, 15min, 30min, 1hour, 2hour, 4hour, 6hour, 12hour, 1day, 3day, 1week)
- **size** (*int*) – number of klines to return
- **since** (*int*) – timestamp in ms to return from

```
# default
klines = client.get_klines('eth_btc', '1min')

# optional params
klines = client.get_klines('eth_btc', '1hour', size=20, since=1417449600000)
```

Returns API response

```
[
  [
    1417449600000, # timestamp
    2339.11,       # open
    2383.15,      # high
    2322,         # low
    2369.85,     # close
    83850.06     # volume
  ],
  [
    1417536000000,
    2370.16,
    2380,
    2352,
    2367.37,
    17259.83
  ]
]
```

Raises AllcoinResponseException, BinanceAPIException

get_userinfo()

Get account info

```
# default
info = client.get_userinfo()
```

Returns API response

```
{
  "info": {
    "funds": {
      "free": {
        "btc": "0",
        "usd": "0",
        "ltc": "0"
      },
      "freezed": {
        "btc": "0",
        "usd": "0",
        "ltc": "0"
      }
    }
  },
  "result": true
}
```

Raises AllcoinResponseException, BinanceAPIException

create_order (*symbol, side, price, amount*)

Create an order

Parameters

- **symbol** (*str*) – required
- **side** (*str*) – order side buy/sell
- **price** (*str*) – order price
- **amount** (*str*) – order quantity

```
# default
order = client.create_order('eth_btc', 'buy', '0.2348', '100')
```

Returns API response

```
{
  "order_id": "123456",
  "result": true
}
```

Raises AllcoinResponseException, BinanceAPIException

create_buy_order (*symbol, price, amount*)

Create a buy order

Parameters

- **symbol** (*str*) – required
- **price** (*str*) – order price
- **amount** (*str*) – order quantity

```
# default
order = client.create_buy_order('eth_btc', '0.2348', '100')
```

Returns API response

```
{
  "order_id": "123456",
  "result": true
}
```

Raises AllcoinResponseException, BinanceAPIException

create_sell_order (*symbol, price, amount*)

Create a sell order

Parameters

- **symbol** (*str*) – required
- **price** (*str*) – order price
- **amount** (*str*) – order quantity

```
# default
order = client.create_buy_order('eth_btc', '0.2348', '100')
```

Returns API response

```
{
  "order_id": "123456",
  "result": true
}
```

Raises AllcoinResponseException, BinanceAPIException

batch_orders (*symbol, order_data, order_type=None*)

Create an order

Parameters

- **symbol** (*str*) – required
- **order_data** (*list of dicts*) – list of dictionaries of price, amount and type
- **order_type** (*str*) – optional buy/sell default used if type not set in order_data dict

```
order_data = [
  {
    "price": "0.0123",
    "amount": "120",
    "type": "sell"
  },
  {
    "price": "0.0112",
    "amount": "110"
  }
]
order = client.create_order('eth_btc', order_data, type='buy')
```

Returns API response

```
{
  "order_info": [
    {"order_id": 41724206},
    {"error_code": 10011, "order_id": -1},
    {"error_code": 10014, "order_id": -1}
  ],
  "result": true
}
```

Raises AllcoinResponseException, BinanceAPIException

cancel_order (*symbol, order_id*)

Cancel an order or up to 3 orders

Parameters

- **symbol** (*str*) – required
- **order_id** (*str*) – order ID (multiple orders are separated by a comma ‘,’; 3 orders at most are allowed per request)

```
# single order
order = client.cancel_order('eth_btc', '123456')
```

```
# multiple orders
order = client.cancel_order('eth_btc', '123456,123457,123557')
```

Returns API response

```
# single order id
{
  "order_id": "123456",
  "result": true
}

# multiple order ids
{
  "success": "123456,123457",
  "error": "123458"
}
```

Raises AllcoinResponseException, BinanceAPIException

get_order (*symbol*, *order_id*)

Get info about a particular order

Parameters

- **symbol** (*str*) – required
- **order_id** (*str*) – order ID to fetch

```
order = client.get_order('eth_btc', '123456')
```

Returns API response

```
{
  "result": true,
  "orders": [
    {
      "amount": 0.1,
      "avg_price": 0,
      "create_date": 1418008467000,
      "deal_amount": 0,
      "order_id": 10000591,
      "price": 500,
      "status": 0,
      "symbol": "btc_usd",
      "type": "sell"
    },
    {
      "amount": 0.2,
      "avg_price": 0,
      "create_date": 1417417957000,
      "deal_amount": 0,
      "order_id": 10000724,
      "price": 0.1,
      "status": 0,
      "symbol": "btc_usd",

```

```

        "type": "buy"
    }
]
}

```

Raises AllcoinResponseException, BinanceAPIException

get_open_orders (*symbol*)

Get info about all open orders

Parameters **symbol** (*str*) – required

```
order = client.get_open_orders('eth_btc')
```

Returns API response

```

{
  "result": true,
  "orders": [
    {
      "amount": 0.1,
      "avg_price": 0,
      "create_date": 1418008467000,
      "deal_amount": 0,
      "order_id": 10000591,
      "price": 500,
      "status": 0,
      "symbol": "btc_usd",
      "type": "sell"
    },
    {
      "amount": 0.2,
      "avg_price": 0,
      "create_date": 1417417957000,
      "deal_amount": 0,
      "order_id": 10000724,
      "price": 0.1,
      "status": 0,
      "symbol": "btc_usd",
      "type": "buy"
    }
  ]
}

```

↪ *filled, 2 = fully filled, 10 = cancelled*
average transaction price
order time
filled quantity
order id
entrustment price
0 = unfilled, 1 = partially_

Raises AllcoinResponseException, BinanceAPIException

get_orders (*symbol, order_status, order_ids*)

Get info about multiple orders

Parameters

- **symbol** (*str*) – required
- **order_status** (*int*) – 0 for unfilled orders; 1 for filled orders

- **order_ids** (*str*) – order IDs to fetch (multiple orders are separated by ‘;’, 50 orders at most are allowed per request)

```
order = client.get_orders('eth_btc', 1)
```

Returns API response

```
{
  "result": true,
  "orders": [
    {
      "amount": 0.1,
      "avg_price": 0,
      "create_date": 1418008467000,
      "deal_amount": 0,
      "order_id": 10000591,
      "price": 500,
      "status": 0,
      "symbol": "btc_usd",
      "type": "sell"
    },
    {
      "amount": 0.2,
      "avg_price": 0,
      "create_date": 1417417957000,
      "deal_amount": 0,
      "order_id": 10000724,
      "price": 0.1,
      "status": 0,
      "symbol": "btc_usd",
      "type": "buy"
    }
  ]
}
```

↔ filled, 2 = fully filled, 10 = cancelled

average transaction price
order time
filled quantity
order id
entrustment price
0 = unfilled, 1 = partially_

Raises AllcoinResponseException, BinanceAPIException

get_order_history (*symbol, order_status, page=1, limit=200*)

Get history of orders for a symbol

Parameters

- **symbol** (*str*) – required
- **order_status** (*int*) – 0 for unfilled orders; 1 for filled orders
- **page** (*int*) – page to fetch
- **limit** (*int*) – amount on each page

```
order = client.get_order_history('eth_btc', 1)
```

Returns API response

```
[
  {
```

```
    "current_page": 1,
    "orders": [
        {
            "amount": 0,
            "avg_price": 0,
            "create_date": 1405562100000,
            "deal_amount": 0,
            "order_id": 0,
            "price": 0,
            "status": 2,
            "symbol": "btc_usd",
            "type": "sell"
        }
    ],
    "page_length": 1,
    "result": true,
    "total": 3
}
```

Raises AllcoinResponseException, BinanceAPIException

exceptions module

exception allcoin.exceptions.AllcoinAPIException(*response*)

Bases: exceptions.Exception

CODES = {'10006': "User doesn't exist", '10007': 'Signature does not match', '10000'

__init__(*response*)

exception allcoin.exceptions.AllcoinRequestException(*message*)

Bases: exceptions.Exception

__init__(*message*)

4.2 Index

- genindex

a

`allcoin.client`, 20

`allcoin.exceptions`, 30

Symbols

`__init__()` (allcoin.client.Client method), 20
`__init__()` (allcoin.exceptions.AllcoinAPIException method), 30
`__init__()` (allcoin.exceptions.AllcoinRequestException method), 30

A

allcoin.client (module), 20
allcoin.exceptions (module), 30
AllcoinAPIException, 30
AllcoinRequestException, 30
API_URL (allcoin.client.Client attribute), 20
API_VERSION (allcoin.client.Client attribute), 20

B

`batch_orders()` (allcoin.client.Client method), 26

C

`cancel_order()` (allcoin.client.Client method), 26
Client (class in allcoin.client), 20
CODES (allcoin.exceptions.AllcoinAPIException attribute), 30
`create_buy_order()` (allcoin.client.Client method), 25
`create_order()` (allcoin.client.Client method), 24
`create_sell_order()` (allcoin.client.Client method), 25

G

`get_klines()` (allcoin.client.Client method), 23
`get_open_orders()` (allcoin.client.Client method), 28
`get_order()` (allcoin.client.Client method), 27
`get_order_book()` (allcoin.client.Client method), 21
`get_order_history()` (allcoin.client.Client method), 29
`get_orders()` (allcoin.client.Client method), 28
`get_ticker()` (allcoin.client.Client method), 20
`get_trade_history()` (allcoin.client.Client method), 22
`get_trades()` (allcoin.client.Client method), 22
`get_userinfo()` (allcoin.client.Client method), 24

O

ORDER_STATUS_CANCELLED (allcoin.client.Client attribute), 20
ORDER_STATUS_FILLED (allcoin.client.Client attribute), 20
ORDER_STATUS_PARTIALLY_FILLED (allcoin.client.Client attribute), 20
ORDER_STATUS_UNFILLED (allcoin.client.Client attribute), 20