

---

# **python-bigone Documentation**

*Release 0.2.0*

**Sam McHardy**

**Jun 27, 2018**



---

## Contents

---

<b>1</b>	<b>Features</b>	<b>3</b>
<b>2</b>	<b>Quick Start</b>	<b>5</b>
<b>3</b>	<b>Donate</b>	<b>7</b>
<b>4</b>	<b>Other Exchanges</b>	<b>9</b>
4.1	Contents . . . . .	9
4.2	Index . . . . .	20
	<b>Python Module Index</b>	<b>21</b>



This is an unofficial Python wrapper for the BigONE exchanges REST API v2. I am in no way affiliated with BigONE, use at your own risk.

**PyPi** <https://pypi.python.org/pypi/python-bigone>

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

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



# CHAPTER 1

---

## Features

---

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





## CHAPTER 2

---

### Quick Start

---

Register an account with BigONE.

Generate an API Key and store it.

```
pip install python-bigone
```

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

# get markets
markets = client.get_markets()

# get market order book
depth = client.get_order_book('ETH-BTC')

# get market trades
trades = client.get_market_trades('ETH-BTC')

# get your accounts
currencies = client.get_accounts()

# place a bid order
transaction = client.create_order('KCS-BTC', Client.SIDE_BID, '0.01', '1000')

# place an ask order
transaction = client.create_order('KCS-BTC', Client.SIDE_ASK, '0.01', '1000')

# get a list of your orders for a symbol
orders = client.get_orders('ETH-BTC')

# get a list of your trades for a symbol
orders = client.get_trades('ETH-BTC')

# get list of all withdrawals
```

(continues on next page)

(continued from previous page)

```
withdrawals = client.get_withdrawals()

# get list of all deposits
deposits = client.get_deposits()
```

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 [Allcoin](#) check out my [python-allucoin](#) library.

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

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

## 4.1 Contents

### 4.1.1 Getting Started

#### Installation

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

```
pip install python-bigone
```

#### Register on BigONE

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

#### Generate an API Key

To use signed account methods you are required to [create an API Key](#) and store it.

## Initialise the client

Pass your API Key and Secret

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

## API Rate Limit

No information

## 4.1.2 BigONE API

### client module

**class** bigone.client.**Client** (*api\_key, api\_secret*)

Bases: object

**API\_URL** = 'https://big.one/api/v2'

**SIDE\_BID** = 'BID'

**SIDE\_ASK** = 'ASK'

**\_\_init\_\_** (*api\_key, api\_secret*)

Big.One API Client constructor

<https://open.big.one/>

#### Parameters

- **api\_key** (*str*) – Api Key
- **api\_secret** – Api Secret

```
client = Client(api_key, api_secret)
```

**get\_accounts** ()

List accounts of current user

```
accounts = client.get_accounts()
```

**Returns** list of dicts

```
[
  {
    "type": "account",
    "user_id": "5c1b7700-c903-41f8-9b4c-d78be9b2685d",
    "account_id": "7eb4201b-9ae8-450d-819a-1c7dfd272e3d",
    "account_type": "BTC",
    "account_name": "Bitcoin",
    "account_logo_url": "https://storage.googleapis.com/big-one/coins/BTC.
↪svg",
    "public_key": "16zy7YGERPEygQitMFAQp4afPJBZeo2WK6",
    "system_account": "",
    "active_balance": "0.00000000",
```

(continues on next page)

(continued from previous page)

```

        "frozen_balance": "0.00000000",
        "estimated_btc_price": "1.00000000",
        "verification": "EMAIL_VERIFIED",
        "has_asset_pin": false
    },
    {
        "type": "account",
        "user_id": "5c1b7700-c903-41f8-9b4c-d78be9b2685d",
        "account_id": "3d7c2353-add3-4e03-8637-c78e4d2b17cb",
        "account_type": "ETH",
        "account_name": "Ether",
        "account_logo_url": "https://storage.googleapis.com/big-one/coins/ETH.
↔svg",
        "public_key": "0xae0fde180aae6e0916632dabfdd2f8733c8a03826",
        "system_account": "",
        "active_balance": "0.00000000",
        "frozen_balance": "0.00000000",
        "estimated_btc_price": "0.06500000",
        "verification": "EMAIL_VERIFIED",
        "has_asset_pin": false
    }
]

```

**Raises** BigoneRequestException, BigoneAPIException

**get\_account** (*currency*)

Get account for a currency

**Parameters** **currency** (*str*) – Name of currency

```
account = client.get_account('BTC')
```

**Returns** dict

```

{
    "type": "account",
    "user_id": "5c1b7700-c903-41f8-9b4c-d78be9b2685d",
    "account_id": "7eb4201b-9ae8-450d-819a-1c7dfd272e3d",
    "account_type": "BTC",
    "account_name": "Bitcoin",
    "account_logo_url": "https://storage.googleapis.com/big-one/coins/BTC.svg
↔",
    "public_key": "16zy7YGERPEygQitMFaqP4afPJBZeo2WK6",
    "system_account": "",
    "active_balance": "0.00000000",
    "frozen_balance": "0.00000000",
    "estimated_btc_price": "1.00000000",
    "verification": "EMAIL_VERIFIED",
    "has_asset_pin": false,
    "withdrawal": {
        "fee": "0.002",
        "fee_type": "BTC"
    },
    "deposits": [],
    "withdrawals": [],
}

```

(continues on next page)

(continued from previous page)

```
"recipients": []
}
```

**Raises** BigoneRequestException, BigoneAPIException

**get\_markets()**

List markets

[https://open.big.one/docs/api\\_market.html#all-markets](https://open.big.one/docs/api_market.html#all-markets)

```
markets = client.get_markets()
```

**Returns** list of dicts

```
[
  {
    "uuid": "d2185614-50c3-4588-b146-b8afe7534da6",
    "quoteScale": 8,
    "quoteAsset": {
      "uuid": "0df9c3c3-255a-46d7-ab82-dedae169fba9",
      "symbol": "BTC",
      "name": "Bitcoin"
    },
    "name": "BTG/BTC",
    "baseScale": 4,
    "baseAsset": {
      "uuid": "5df3b155-80f5-4f5a-87f6-a92950f0d0ff",
      "symbol": "BTG",
      "name": "Bitcoin Gold"
    }
  }
]
```

**Raises** BigoneRequestException, BigoneAPIException

**get\_tickers()**

List market tickers

[https://open.big.one/docs/api\\_tickers.html#tickers-of-all-market](https://open.big.one/docs/api_tickers.html#tickers-of-all-market)

```
markets = client.get_tickers()
```

**Returns** list of dicts

```
[
  {
    "volume": null,
    "open": "1.0000000000000000",
    "market_uuid": "ETH-EOS",
    "low": null,
    "high": null,
    "daily_change_perc": "0",
    "daily_change": "0E-16",
```

(continues on next page)



(continued from previous page)

```

    "close": "1.0000000000000000",
    "bid": {
        "price": "1.0000000000000000",
        "amount": "106.0000000000000000"
    },
    "ask": {
        "price": "45.0000000000000000",
        "amount": "4082.3283464000000000"
    }
}
]

```

**Raises** BigoneRequestException, BigoneAPIException

**get\_ticker** (*symbol*)

Get symbol market details

[https://open.big.one/docs/api\\_tickers.html#ticker-of-one-market](https://open.big.one/docs/api_tickers.html#ticker-of-one-market)

**Parameters** **symbol** (*str*) – Name of symbol

```

# using market ID
market = client.get_ticker('ETH-BTC')

# using market UUID
market = client.get_ticker('d2185614-50c3-4588-b146-b8afe7534da6')

```

**Returns** dict

```

{
    "volume": null,
    "open": "42.0000000000000000",
    "market_uuid": "BTC-EOS",
    "low": "42.0000000000000000",
    "high": null,
    "daily_change_perc": "0",
    "daily_change": "0E-16",
    "close": "42.0000000000000000",
    "bid": {
        "price": "42.0000000000000000",
        "amount": "3.3336371100000000"
    },
    "ask": {
        "price": "45.0000000000000000",
        "amount": "4082.3283464000000000"
    }
}

```

**Raises** BigoneRequestException, BigoneAPIException

**get\_order\_book** (*symbol*)

Get symbol market details

**Parameters** **symbol** (*str*) – Name of symbol

```
# Using market ID
book = client.get_order_book('ETH-BTC')

# Using market UUID
book = client.get_order_book('d2185614-50c3-4588-b146-b8afe7534da6')
```

**Returns** dict

```
{
  "market_uuid": "BTC-EOS",
  "bids": [
    {
      "price": "42",
      "order_count": 4,
      "amount": "23.33363711"
    }
  ],
  "asks": [
    {
      "price": "45",
      "order_count": 2,
      "amount": "4193.3283464"
    }
  ]
}
```

**Raises** BigoneRequestException, BigoneAPIException

**get\_market\_trades** (*symbol*, *after=None*, *before=None*, *first=None*, *last=None*)  
Get market trades - max 50

[https://open.big.one/docs/api\\_market\\_trade.html#trades-of-a-market](https://open.big.one/docs/api_market_trade.html#trades-of-a-market)

**Parameters**

- **symbol** (*str*) – Name of symbol
- **after** (*int*) – Return trades after this id
- **before** (*int*) – Return trades before this id
- **first** (*int*) – Slicing count
- **last** (*int*) – Slicing count

```
trades = client.get_market_trades('ETH-BTC')

# using after trade ID
trades = client.get_market_trades('ETH-BTC', after=1)

# using first slice value
trades = client.get_market_trades('ETH-BTC', first=20)
```

**Returns** list of dicts

```
{
  "edges": [
    {
      "node": {
        "taker_side": "BID",
        "price": "46.145000000000000000",
        "market_uuid": "BTC-EOS",
        "id": 1,
        "amount": "0.246548000000000000"
      },
      "cursor": "dGVzdGN1cmVzZQo="
    }
  ],
  "page_info": {
    "end_cursor": "dGVzdGN1cmVzZQo=",
    "start_cursor": "dGVzdGN1cmVzZQo=",
    "has_next_page": true,
    "has_previous_page": false
  }
}
```

**Raises** BigoneRequestException, BigoneAPIException

**create\_order** (*symbol, side, price, amount*)

Create a new order

#### Parameters

- **symbol** (*str*) – Name of symbol
- **side** (*str*) – side of order (BID or ASK)
- **price** (*str*) – Price as string
- **amount** (*str*) – Amount as string

```
order = client.create_order('ETH-BTC', 'BID', '1.0', '1.0')
```

**Returns** dict

```
{
  "id": 10,
  "market_uuid": "BTC-EOS",
  "price": "10.00",
  "amount": "10.00",
  "filled_amount": "9.0",
  "avg_deal_price": "12.0",
  "side": "ASK",
  "state": "FILLED"
}
```

**Raises** BigoneRequestException, BigoneAPIException

**get\_orders** (*symbol, after=None, before=None, first=None, last=None, side=None, state=None*)

Get a list of orders

#### Parameters

- **symbol** (*str*) – Name of symbol
- **after** (*int*) – Return trades after this id
- **before** (*int*) – Return trades before this id
- **first** (*int*) – Slicing count
- **last** (*int*) – Slicing count
- **side** (*str*) – Order Side ASK|BID
- **state** (*str*) – Order State CANCELED|FILLED|PENDING

```
orders = client.get_orders('ETH-BTC')
```

**Returns** dict

```
{
  "edges": [
    {
      "node": {
        "id": 10,
        "market_uuid": "d2185614-50c3-4588-b146-b8afe7534da6",
        "price": "10.00",
        "amount": "10.00",
        "filled_amount": "9.0",
        "avg_deal_price": "12.0",
        "side": "ASK",
        "state": "FILLED"
      },
      "cursor": "dGVzdGN1cmVzZQo="
    }
  ],
  "page_info": {
    "end_cursor": "dGVzdGN1cmVzZQo=",
    "start_cursor": "dGVzdGN1cmVzZQo=",
    "has_next_page": true,
    "has_previous_page": false
  }
}
```

**Raises** BigoneRequestException, BigoneAPIException

**get\_order** (*order\_id*)

Get an order

[https://open.big.one/docs/api\\_orders.html#get-one-order](https://open.big.one/docs/api_orders.html#get-one-order)

**Parameters** **order\_id** (*str*) – Id of order

```
orders = client.get_order('10')
```

**Returns** dict

```
{
  "id": 10,
  "market_uuid": "d2185614-50c3-4588-b146-b8afe7534da6",
```

(continues on next page)

(continued from previous page)

```

"price": "10.00",
"amount": "10.00",
"filled_amount": "9.0",
"avg_deal_price": "12.0",
"side": "ASK",
"state": "FILLED"
}

```

**Raises** BigoneRequestException, BigoneAPIException

**cancel\_order** (*order\_id*)

Cancel an order

[https://open.big.one/docs/api\\_orders.html#candle-order](https://open.big.one/docs/api_orders.html#candle-order)

**Parameters** *order\_id* (*str*) – Id of order

```
res = client.cancel_order('10')
```

**Returns** dict

```
{}
```

**Raises** BigoneRequestException, BigoneAPIException

**cancel\_orders** ()

Cancel all orders

[https://open.big.one/docs/api\\_orders.html#candle-all-orders](https://open.big.one/docs/api_orders.html#candle-all-orders)

```
res = client.cancel_orders()
```

**Returns** dict

```
{}
```

**Raises** BigoneRequestException, BigoneAPIException

**get\_trades** (*symbol=None, after=None, before=None, first=None, last=None*)

Get a list of your trades

**Parameters**

- **symbol** (*str*) – Name of symbol
- **before** (*int*) – Return trades before this id
- **first** (*int*) – Slicing count
- **last** (*int*) – Slicing count

```
trades = client.get_trades('ETH-BTC')
```

**Returns** dict

```
{
  "edges": [
    {
      "node": {
        "viewer_side": "ASK" // ASK, BID, SELF_TRADING
        "taker_side": "BID",
        "price": "46.1450000000000000",
        "market_uuid": "BTC-EOS",
        "id": 1,
        "amount": "0.2465480000000000"
      },
      "cursor": "dGVzdGN1cmVzZQo="
    }
  ],
  "page_info": {
    "end_cursor": "dGVzdGN1cmVzZQo=",
    "start_cursor": "dGVzdGN1cmVzZQo=",
    "has_next_page": true,
    "has_previous_page": false
  }
}
```

**Raises** BigoneRequestException, BigoneAPIException

**withdrawals** (*first=None, after=None*)

Get a list of withdrawals

[https://open.big.one/docs/api\\_withdrawal.html#get-withdrawals-of-user](https://open.big.one/docs/api_withdrawal.html#get-withdrawals-of-user)

**Parameters**

- **first** (*str*) – Slicing count
- **after** (*str*) – Return withdrawals after this value

```
withdrawal = client.withdrawals()
```

**Returns** dict

```
{
  "edges": [
    {
      "node": {
        "id": 10,
        "customer_id": "ETH",
        "asset_uuid": "ETH",
        "amount": "5",
        "state": "CONFIRMED",
        "note": "2018-03-15T16:13:45.610463Z",
        "txid":
        ↪ "0x4643bb6b393ac20a6175c713175734a72517c63d6f73a3ca90a15356f2e967da0",
        "completed_at": "2018-03-15T16:13:45.610463Z",
        "inserted_at": "2018-03-15T16:13:45.610463Z",
        "is_internal": true,
        "target_address":
        ↪ "0x4643bb6b393ac20a6175c713175734a72517c63d6f7"
      },

```

(continues on next page)

(continued from previous page)

```

        "cursor": "dGVzdGN1cmVzZQo="
    }
],
"page_info": {
    "end_cursor": "dGVzdGN1cmVzZQo=",
    "start_cursor": "dGVzdGN1cmVzZQo=",
    "has_next_page": true,
    "has_previous_page": false
}
}

```

**Raises** BigoneRequestException, BigoneAPIException

**get\_deposits** (*first=None, after=None*)

Get a list of deposits

[https://open.big.one/docs/api\\_deposit.html#deposit-of-user](https://open.big.one/docs/api_deposit.html#deposit-of-user)

#### Parameters

- **first** (*str*) – Slicing count
- **after** (*str*) – Return withdrawals after this value

```

# get all deposits
deposits = client.get_deposits()

# get BTC deposits
deposits = client.get_deposits('BTC')

```

**Returns** list of dicts

```

[
  {
    "type": "deposit",
    "deposit_id": "7b76603c-db87-4c4a-b36f-9981058d885e",
    "deposit_type": "ETH",
    "amount": "1.00000000",
    "confirmations": 2231,
    "state": "confirmed",
    "scanner_url": "https://etherscan.io/tx/
↪0x181a1d616b5b07d5a404636c527ca5432878664d4b85a20976bbd72c07e58abe",
    "created_at": "2017-09-06T07:53:41.297047856Z"
  }
]

```

**Raises** BigoneRequestException, BigoneAPIException

## exceptions module

**exception** bigone.exceptions.**BigoneAPIException** (*response*)

Bases: exceptions.Exception

Exception class to handle general API Exceptions

*code* values

*message* format

**\_\_init\_\_** (*response*)

x.**\_\_init\_\_**(...) initializes x; see help(type(x)) for signature

**exception** bigone.exceptions.**BigoneRequestException** (*message*)

Bases: exceptions.Exception

**\_\_init\_\_** (*message*)

x.**\_\_init\_\_**(...) initializes x; see help(type(x)) for signature

## 4.2 Index

- genindex



**b**

`bigone.client`, 10

`bigone.exceptions`, 19



## Symbols

`__init__()` (bigone.client.Client method), 10  
`__init__()` (bigone.exceptions.BigoneAPIException method), 20  
`__init__()` (bigone.exceptions.BigoneRequestException method), 20

## A

`API_URL` (bigone.client.Client attribute), 10

## B

`bigone.client` (module), 10  
`bigone.exceptions` (module), 19  
`BigoneAPIException`, 19  
`BigoneRequestException`, 20

## C

`cancel_order()` (bigone.client.Client method), 17  
`cancel_orders()` (bigone.client.Client method), 17  
`Client` (class in bigone.client), 10  
`create_order()` (bigone.client.Client method), 15

## G

`get_account()` (bigone.client.Client method), 11  
`get_accounts()` (bigone.client.Client method), 10  
`get_deposits()` (bigone.client.Client method), 19  
`get_market_trades()` (bigone.client.Client method), 14  
`get_markets()` (bigone.client.Client method), 12  
`get_order()` (bigone.client.Client method), 16  
`get_order_book()` (bigone.client.Client method), 13  
`get_orders()` (bigone.client.Client method), 15  
`get_ticker()` (bigone.client.Client method), 13  
`get_tickers()` (bigone.client.Client method), 12  
`get_trades()` (bigone.client.Client method), 17

## S

`SIDE_ASK` (bigone.client.Client attribute), 10  
`SIDE_BID` (bigone.client.Client attribute), 10

## W

`withdrawals()` (bigone.client.Client method), 18