
HitBTC Websocket Client Documentation

Release 1.0.0

Nils Diefenbach

Jan 24, 2018

Contents

| | | |
|----------|---|-----------|
| 1 | HitBTC Websocket API Client Code Reference | 3 |
| 2 | The Client Object | 5 |
| 3 | The Connector Object | 7 |
| 4 | The Websocket APP Object | 9 |
| 5 | Indices and tables | 11 |
| | Python Module Index | 13 |

Contents:

CHAPTER 1

HitBTC Websocket API Client Code Reference

HitBTC WSS API V2.0 Client.

The Client Object

```
class hitbtc.client.HitBTC (key=None, secret=None, raw=None, stdout_only=False, silent=False,  
                             url=None, **conn_ops)
```

HitBTC Websocket API Client class.

Programmed using the official API documentation as a reference.

Documentation can be found here: <https://api.hitbtc.com/?python#socket-api-reference>

```
cancel_order (custom_id=None, **params)
```

Cancel an order via Websocket.

Offical Endpoint Documentation: <https://api.hitbtc.com/?python#cancel-order>

```
credentials_given
```

Assert if credentials are complete.

```
login (key=None, secret=None, basic=None, custom_nonce=None)
```

Login using the WSS API.

Offical Endpoint Documentation: <https://api.hitbtc.com/?python#socket-session-authentication>

```
place_order (custom_id=None, **params)
```

Place a new order via Websocket.

Offical Endpoint Documentation: <https://api.hitbtc.com/?python#place-new-order>

```
recv (block=True, timeout=None)
```

Retrieve data from the connector queue.

```
replace_order (custom_id=None, **params)
```

Replace an existing order via Websocket.

Offical Endpoint Documentation: <https://api.hitbtc.com/?python#cancel-replace-orders>

```
request_active_orders (custom_id=None, **params)
```

Request your account's active orders.

This requires you to be logged-in! Call `login()` first!

Offical Endpoint Documentation: <https://api.hitbtc.com/?python#get-active-orders-2>

request_balance (*custom_id=None, **params*)

Request your account's balance.

This requires you to be logged-in! Call `login()` first.

Official Endpoint Documentation: <https://api.hitbtc.com/?python#get-trading-balance>

request_currencies (*custom_id=None, **params*)

Request currencies currently listed at HitBTC.

Official Endpoint Documentation: <https://api.hitbtc.com/?python#get-currencies>

request_symbols (*custom_id=None, **params*)

Request symbols currently traded at HitBTC.

Official Endpoint Documentation: <https://api.hitbtc.com/?python#get-symbols>

request_trades (*custom_id=None, **params*)

Request trades executed at HitBTC.

Official Endpoint Documentation: <https://api.hitbtc.com/?python#get-trades>

start ()

Start the websocket connection.

stop ()

Stop the websocket connection.

subscribe_book (*cancel=False, custom_id=None, **params*)

Request a stream for order book data.

Official Endpoint Documentation: <https://api.hitbtc.com/?python#subscribe-to-orderbook>

subscribe_candles (*cancel=False, custom_id=None, **params*)

Request a stream for candle data.

Official Endpoint Documentation: <https://api.hitbtc.com/?python#subscribe-to-candles>

subscribe_reports (*cancel=False, custom_id=None, **params*)

Request a stream of your account's order activities.

This requires you to be logged-in! Call `HitBTC.login()` first!

Official Endpoint Documentation: <https://api.hitbtc.com/?python#subscribe-to-reports>

subscribe_ticker (*cancel=False, custom_id=None, **params*)

Request a stream for ticker data.

Official Endpoint Documentation: <https://api.hitbtc.com/?python#subscribe-to-ticker>

subscribe_trades (*cancel=False, custom_id=None, **params*)

Request a stream for trade data.

Official Endpoint Documentation: <https://api.hitbtc.com/?python#subscribe-to-trades>

The Connector Object

```
class hitbtc.connector.HitBTCTConnector (url=None, raw=None, stdout_only=False,  
                                         silent=False, **conn_ops)
```

Class to pre-process HitBTC data, before putting it on the internal queue.

Data on the queue is available as a 3-item-tuple by default.

Response items on the queue are formatted as: ('Response', 'Success' or 'Failure', (request, response))

'Success' indicates a successful response and 'Failure' a failed one. `request` is the original payload sent by the client and `response` the related response object from the server.

Stream items on the queue are formatted as: (method, symbol, params)

You can disable extraction and handling by passing `raw=True` on instantiation. Note that this will also turn off recording of sent requests, as well all logging activity.

```
authenticate (key, secret, basic=False, custom_nonce=None)
```

Login to the HitBTC Websocket API using the given public and secret API keys.

```
echo (msg)
```

Print message to stdout if `silent` isn't True.

```
put (item, block=False, timeout=None)
```

Place the given item on the internal q.

```
send (method, custom_id=None, **params)
```

Send the given Payload to the API via the websocket connection.

Parameters

- **method** – JSONRPC method to call
- **custom_id** – custom ID to identify response messages relating to this request
- **kwargs** – payload parameters as key=value pairs

The Websocket APP Object

```
class hitbtc.wss.WebSocketConnector(url, timeout=None, q_maxsize=None, reconnect_interval=None, log_level=None)
```

Websocket Connection Thread.

Inspired heavily by ekulyk's PythonPusherClient Connection Class <https://github.com/ekulyk/PythonPusherClient/blob/master/pusherclient/connection.py>

Data received is available by calling WebSocketConnection.recv()

disconnect()

Disconnect from the websocket connection and joins the Thread.

pass_up(data, recv_at)

Pass data up to the client via the internal Queue().

Parameters

- **data** – data to be passed up
- **recv_at** – float, time of reception

Returns

reconnect()

Issue a reconnection by setting the reconnect_required event.

recv(block=True, timeout=None)

Wrap for self.q.get().

Parameters

- **block** – Whether or not to make the call to this method block
- **timeout** – Value in seconds which determines a timeout for get()

Returns

run()

Run the main method of thread.

send (*data*)

Send the given Payload to the API via the websocket connection.

Furthermore adds the sent payload to self.history.

Parameters *data* – data to be sent

Returns

stop ()

Wrap around disconnect().

class hitbtc.wss.**WebSocketConnectorThread** (*url*, *timeout=None*, *q_maxsize=None*, *re-connect_interval=None*, *log_level=None*,
***kwargs*)

Thread-based WebSocketConnector.

disconnect ()

Disconnect from the websocket and join thread.

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`

h

`hitbtc`, 3

A

authenticate() (hitbtc.connector.HitBTCCConnector method), 7

C

cancel_order() (hitbtc.client.HitBTC method), 5
credentials_given (hitbtc.client.HitBTC attribute), 5

D

disconnect() (hitbtc.wss.WebSocketConnector method), 9
disconnect() (hitbtc.wss.WebSocketConnectorThread method), 10

E

echo() (hitbtc.connector.HitBTCCConnector method), 7

H

HitBTC (class in hitbtc.client), 5
hitbtc (module), 3
HitBTCCConnector (class in hitbtc.connector), 7

L

login() (hitbtc.client.HitBTC method), 5

P

pass_up() (hitbtc.wss.WebSocketConnector method), 9
place_order() (hitbtc.client.HitBTC method), 5
put() (hitbtc.connector.HitBTCCConnector method), 7

R

reconnect() (hitbtc.wss.WebSocketConnector method), 9
recv() (hitbtc.client.HitBTC method), 5
recv() (hitbtc.wss.WebSocketConnector method), 9
replace_order() (hitbtc.client.HitBTC method), 5
request_active_orders() (hitbtc.client.HitBTC method), 5
request_balance() (hitbtc.client.HitBTC method), 5
request_currencies() (hitbtc.client.HitBTC method), 6
request_symbols() (hitbtc.client.HitBTC method), 6

request_trades() (hitbtc.client.HitBTC method), 6
run() (hitbtc.wss.WebSocketConnector method), 9

S

send() (hitbtc.connector.HitBTCCConnector method), 7
send() (hitbtc.wss.WebSocketConnector method), 9
start() (hitbtc.client.HitBTC method), 6
stop() (hitbtc.client.HitBTC method), 6
stop() (hitbtc.wss.WebSocketConnector method), 10
subscribe_book() (hitbtc.client.HitBTC method), 6
subscribe_candles() (hitbtc.client.HitBTC method), 6
subscribe_reports() (hitbtc.client.HitBTC method), 6
subscribe_ticker() (hitbtc.client.HitBTC method), 6
subscribe_trades() (hitbtc.client.HitBTC method), 6

W

WebSocketConnector (class in hitbtc.wss), 9
WebSocketConnectorThread (class in hitbtc.wss), 10