**Welcome to python-binance v0.5.6**

This is an unofficial Python wrapper for the [Binance exchange REST API v1/3](https://github.com/binance-exchange/binance-official-api-docs). I am in no way affiliated with Binance, use at your own risk.

If you came here looking for the [Binance exchange](https://www.binance.com/?ref=10099792) to purchase cryptocurrencies, then [go here](https://www.binance.com/?ref=10099792). If you want to automate interactions with Binance stick around.

Source code

<https://github.com/sammchardy/python-binance>

Documentation

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

Binance API Telegram

<https://t.me/binance_api_english>

**Features**

* Implementation of General, Market Data and Account endpoints.
* Simple handling of authentication
* No need to generate timestamps yourself, the wrapper does it for you
* Response exception handling
* Websocket handling with reconnection and multiplexed connections
* Symbol Depth Cache
* Withdraw functionality
* Deposit addresses

**Quick Start**

[Register an account with Binance](https://www.binance.com/register.html?ref=10099792).

[Generate an API Key](https://www.binance.com/userCenter/createApi.html) and assign relevant permissions.

pip install python-binance

from binance.client import Client

client = Client(api\_key, api\_secret)

# get market depth

depth = client.get\_order\_book(symbol='BNBBTC')

# place market buy order

order = client.create\_order(

symbol='BNBBTC',

side=Client.SIDE\_BUY,

type=Client.ORDER\_TYPE\_MARKET,

quantity=100)

# get all symbol prices

prices = client.get\_all\_tickers()

# withdraw 100 ETH

# check docs for assumptions around withdrawals

from binance.exceptions import BinanceApiException, BinanceWithdrawException

try:

result = client.withdraw(

asset='ETH',

address='<eth\_address>',

amount=100)

except BinanceApiException as e:

print(e)

except BinanceWithdrawException as e:

print(e)

else:

print("Success")

# fetch list of withdrawals

withdraws = client.get\_withdraw\_history()

# fetch list of ETH withdrawals

eth\_withdraws = client.get\_withdraw\_history('ETH)

# get a deposit address

address = client.get\_deposit\_address('BTC)

# start trade websocket

def process\_message(msg):

print("message type: {}".format(msg['e']))

print(msg)

# do something

from binance.websockets import BinanceSocketManager

bm = BinanceSocketManager(client)

bm.start\_aggtrade\_socket(symbol='BNBBTC')

bm.start()

For more [check out the documentation](https://python-binance.readthedocs.io/en/latest/).

**Donate**

If this library helped you out feel free to donate.

* ETH: 0x79c9A113d63B6AC024c1AAae6f6419721C3611A3
* NEO: AVJB4ZgN7VgSUtArCt94y7ZYT6d5NDfpBo
* BTC: 1Dknp6L6oRZrHDECRedihPzx2sSfmvEBys

**Other Exchanges**

If you use [Quoinex](https://accounts.quoinex.com/sign-up?affiliate=PAxghztC67615) or [Qryptos](https://accounts.qryptos.com/sign-up?affiliate=PAxghztC67615) check out my [python-quoine](https://github.com/sammchardy/python-quoine) library.

If you use [Kucoin](https://www.kucoin.com/#/?r=E42cWB) check out my [python-kucoin](https://github.com/sammchardy/python-kucoin) library.

If you use [IDEX](https://idex.market) check out my [python-idex](https://github.com/sammchardy/python-idex) library.