
poloniex_aio Documentation

Release 0.1.1

Olivier Cazade

Dec 18, 2018

General:

1	Public API	1
2	Trading API	3
3	Indices and tables	13
	Python Module Index	15

Public API does not require any Poloniex account.

Note that making more than 6 calls per second to the public API, or repeatedly and needlessly fetching excessive amounts of data, can result in your IP being banned.

1.1 Ticker

`poloniex_aio.public.returnTicker(session)`

Returns the ticker for all markets.

Parameters `session` – Aiohttp client session object

1.2 24-hour volume

`poloniex_aio.public.return24hVolume(session)`

Returns the 24-hour volume for all markets, plus totals for primary currencies.

Parameters `session` – Aiohttp client session object

1.3 Order book

`poloniex_aio.public.returnOrderBook(session, *, currencyPair)`

Returns the order book for a given market, as well as a sequence number for use with the Push API and an indicator specifying whether the market is frozen.

Parameters

- `session` – Aiohttp client session object
- `currencyPair` (*String*) – Market identifier, example : BTC_ETH

1.4 Trade history

`poloniex_ao.public.returnTradeHistory (session, *, currencyPair, start=None, end=None)`

Returns the past 200 trades for a given market, or up to 50,000 trades between a range specified in UNIX timestamps by the “start” and “end” parameters.

Parameters

- **session** – Aiohttp client session object
- **currencyPair** (*String*) – Market identifier, example : BTC_ETH
- **start** (*String*) – UNIX timestamp. Every returned trade will have a timestamps greater or equal.
- **end** (*String*) – UNIX timestamp. Every returned trade will have a timestamps lower or equal.

1.5 Chart data

`poloniex_ao.public.returnChartData (session, *, currencyPair, start, end, period)`

Returns candlestick chart data.

Parameters

- **session** – Aiohttp client session object
- **currencyPair** (*String*) – Market identifier, example : BTC_ETH
- **start** (*String*) – UNIX timestamp. Every returned trade will have a timestamps greater or equal.
- **end** (*String*) – UNIX timestamp. Every returned trade will have a timestamps lower or equal.
- **period** (*String*) – candlestick period in seconds; valid values are 300, 900, 1800, 7200, 14400, and 86400

1.6 Currencies

`poloniex_ao.public.returnCurrencies (session)`

Parameters **session** – Aiohttp client session object

Returns information about currencies.

1.7 Loan orders

`poloniex_ao.public.returnLoanOrders (session, *, currency)`

Returns the list of loan offers and demands for a given currency, specified by the “currency” GET parameter.

Parameters

- **session** – Aiohttp client session object
- **currency** (*String*) – Currency identifier, example : BTC

To use the trading API, you will need to create an API key.

Please note that there is a default limit of 6 calls per second.

2.1 Balances

`poloniex_aio.trading.returnBalances (session, api_key, secret_key)`

Returns all of your available balances

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key

2.2 Complete balances

`poloniex_aio.trading.returnCompleteBalances (session, api_key, secret_key, *, account=*)`

Returns all of your balances, including available balance, balance on orders, and the estimated BTC value of your balance.

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key
- **account** (*String*) – By default, this call is limited to your exchange account; set this parameter to “all” to include your margin and lending accounts.

2.3 Deposit Addresses

`poloniex_ao.trading.returnDepositAddresses (session, api_key, secret_key)`

Returns all of your deposit addresses.

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key

2.4 Generate new address

`poloniex_ao.trading.generateNewAddress (session, api_key, secret_key, *, currency)`

Generates a new deposit address

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key
- **currency** (*String*) – The address will be generated for the currency specified here

2.5 Deposits and withdrawals history

`poloniex_ao.trading.returnDepositsWithdrawals (session, api_key, secret_key, *, start=None, end=None)`

Returns your deposit and withdrawal history within a range

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key
- **start** (*String*) – UNIX timestamp. Every returned value will have a timestamps greater or equal.
- **end** (*String*) – UNIX timestamp. Every returned value will have a timestamps lower or equal.

2.6 Open orders

`poloniex_ao.trading.returnOpenOrders (session, api_key, secret_key, *, currencyPair)`

Returns your open orders for a given market.

Parameters

- **session** – Aiohttp client session object

- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key
- **currencyPair** (*String*) – The currency pair. Set this to “all” to return open orders for all markets.

2.7 Trade History

`poloniex_aido.trading.returnTradeHistory` (*session*, *api_key*, *secret_key*, *, *currencyPair*, *start=None*, *end=None*, *limit=None*)

Returns your trade history for a given market.

If you do not specify a range, it will be limited to one day.

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key
- **currencyPair** (*String*) – The currency pair. Set this to “all” to return open orders for all markets.
- **start** (*String*) – UNIX timestamp. Every returned value will have a timestamps greater or equal.
- **end** (*String*) – UNIX timestamp. Every returned value will have a timestamps lower or equal.
- **limit** (*String*) – Maximum number of returned values. Default to 500, max 10 000

2.8 Trades involving an order

`poloniex_aido.trading.returnOrderTrades` (*session*, *api_key*, *secret_key*, *, *orderNumber*)

Returns all trades involving a given order

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key
- **orderNumber** (*String*) – The order id

2.9 Buy order

`poloniex_aido.trading.buy` (*session*, *api_key*, *secret_key*, *, *currencyPair*, *rate*, *amount*, *fillOrKill=None*, *immediateOrCancel=None*, *postOnly=None*)

Places a limit buy order in a given market.

Parameters

- **session** – Aiohttp client session object

- **api_key** (*String*) – the API key
- **secret_key** (*String*) – the API secret key
- **currencyPair** (*String*) – the currency pair
- **rate** (*String*) – the rate to buy
- **amount** (*String*) – the amount to buy
- **fillOrKill** (*String*) – a fill-or-kill order will either fill in its entirety or be completely aborted
- **immediateOrCancel** (*String*) – an immediate-or-cancel order can be partially or completely filled, but any portion of the order that cannot be filled immediately will be canceled rather than left on the order book.
- **postOnly** (*String*) – a post-only order will only be placed if no portion of it fills immediately; this guarantees you will never pay the taker fee on any part of the order that fills.

2.10 Sell order

`poloniex_ao.trading.sell(session, api_key, secret_key, *, currencyPair, rate, amount, fillOrKill=None, immediateOrCancel=None, postOnly=None)`

Places a sell order in a given market. Parameters and output are the same as for the buy method.

2.11 Cancer an order

`poloniex_ao.trading.cancelOrder(session, api_key, secret_key, *, orderNumber)`

Cancels an order you have placed in a given market.

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key
- **orderNumber** (*String*) – Order ID

2.12 Move an order

`poloniex_ao.trading.moveOrder(session, api_key, secret_key, *, orderNumber, rate, amount=None, immediateOrCancel=None, postOnly=None)`

Cancels an order and places a new one of the same type in a single atomic transaction, meaning either both operations will succeed or both will fail.

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key
- **orderNumber** (*String*) – The order id

- **rate** (*String*) – the new rate
- **amount** (*String*) – the new amount
- **immediateOrCancel** (*String*) – an immediate-or-cancel order can be partially or completely filled, but any portion of the order that cannot be filled immediately will be canceled rather than left on the order book.
- **postOnly** (*String*) – a post-only order will only be placed if no portion of it fills immediately; this guarantees you will never pay the taker fee on any part of the order that fills.

2.13 Withdraw

`poloniex_ao.trading.withdraw(session, api_key, secret_key, *, currency, amount, address, paymentId=None)`

Immediately places a withdrawal for a given currency

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key
- **currency** (*String*) – The currency to withdraw
- **amount** (*String*) – The amount to withdraw
- **address** (*String*) – The destination address
- **paymentId** (*String*) – XMR only

2.14 Fee info

`poloniex_ao.trading.returnFeeInfo(session, api_key, secret_key)`

If you are enrolled in the maker-taker fee schedule, returns your current trading fees and trailing 30-day volume in BTC.

This information is updated once every 24 hours.

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key

2.15 Balances sorted by account

`poloniex_ao.trading.returnAvailableAccountBalances(session, api_key, secret_key, *, account=None)`

Returns your balances sorted by account.

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key
- **account** (*String*) – if you wish to fetch only the balances of one account, set the account here.

2.16 Tradable balances

`poloniex_ao.trading.returnTradableBalances (session, api_key, secret_key)`

Returns your current tradable balances for each currency in each market for which margin trading is enabled.

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key

2.17 Transfers

`poloniex_ao.trading.transferBalance (session, api_key, secret_key, *, currency, amount, fromAccount, toAccount)`

Transfers funds from one account to another.

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key
- **fromAccount** (*String*) – Source account
- **toAccount** (*String*) – Destination account

2.18 Margin account summary

`poloniex_ao.trading.returnMarginAccountSummary (session, api_key, secret_key)`

Returns a summary of your entire margin account.

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key

2.19 Margin buy order

`poloniex_ao.trading.marginBuy` (*session*, *api_key*, *secret_key*, *, *currencyPair*, *rate*, *amount*, *lendingRate=None*)

Places a margin buy order in a given market.

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key
- **currencyPair** (*String*) – The currency pair
- **rate** (*String*) – The order rate.
- **amount** (*String*) – The order amount.
- **lendingRate** (*String*) – Maximum lending rate.

2.20 Margin sell order

`poloniex_ao.trading.marginSell` (*session*, *api_key*, *secret_key*, *, *currencyPair*, *rate*, *amount*, *lendingRate=None*)

Places a margin sell order in a given market. Parameters and output are the same as for the `marginBuy` method.

2.21 Margin position

`poloniex_ao.trading.getMarginPosition` (*session*, *api_key*, *secret_key*, *, *currencyPair*)

Returns information about your margin position in a given market.

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key
- **currencyPair** (*String*) – The currency pair. Set this to “all” if you wish to fetch all of your margin positions at once.

2.22 Close margin position

`poloniex_ao.trading.closeMarginPosition` (*session*, *api_key*, *secret_key*, *, *currencyPair*)

Closes your margin position in a given market.

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key
- **currencyPair** (*String*) – The currency pair.

2.23 Create a loan offer

`poloniex_aid.trading.createLoanOffer` (*session, api_key, secret_key, *, currency, amount, duration, autoRenew, lendingRate*)

Creates a loan offer for a given currency.

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key
- **currency** (*String*) – The currency to lend
- **amount** (*String*) – Amount of currency to lend
- **duration** (*String*) – Lending duration
- **autoRenew** (*String*) – Should the loan offer auto renew.
- **lendingRate** (*String*) – The lending rate

2.24 Cancel a loan offer

`poloniex_aid.trading.cancelLoanOffer` (*session, api_key, secret_key, *, orderNumber*)

Cancels a loan offer

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key
- **orderNumber** (*String*) – The loan offer ID

2.25 Open loan offers

`poloniex_aid.trading.returnOpenLoanOffers` (*session, api_key, secret_key*)

Returns your open loan offers for each currency.

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key

2.26 Active loans

`poloniex_aid.trading.returnActiveLoans` (*session, api_key, secret_key*)

Returns your active loans for each currency.

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key

2.27 Lending history

`poloniex_ao.trading.returnLendingHistory` (*session*, *api_key*, *secret_key*, *, *start*, *end*, *limit=None*)

Returns your lending history within a time range.

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key
- **start** (*String*) – UNIX timestamp. Every returned value will have a timestamps greater or equal.
- **end** (*String*) – UNIX timestamp. Every returned value will have a timestamps lower or equal.
- **limit** (*String*) – Maximum number of returned values.

2.28 Toggles the autoRenew setting

`poloniex_ao.trading.toggleAutoRenew` (*session*, *api_key*, *secret_key*, *, *orderNumber*)

Toggles the autoRenew setting on an active.

Parameters

- **session** – Aiohttp client session object
- **api_key** (*String*) – The API key
- **secret_key** (*String*) – The API secret key
- **orderNumber** (*String*) – The loan ID

CHAPTER 3

Indices and tables

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

p

`poloniex_aio.public`, [1](#)
`poloniex_aio.trading`, [3](#)

B

buy() (in module poloniex_aio.trading), 5

C

cancelLoanOffer() (in module poloniex_aio.trading), 10

cancelOrder() (in module poloniex_aio.trading), 6

closeMarginPosition() (in module poloniex_aio.trading), 9

createLoanOffer() (in module poloniex_aio.trading), 10

G

generateNewAddress() (in module poloniex_aio.trading), 4

getMarginPosition() (in module poloniex_aio.trading), 9

M

marginBuy() (in module poloniex_aio.trading), 9

marginSell() (in module poloniex_aio.trading), 9

moveOrder() (in module poloniex_aio.trading), 6

P

poloniex_aio.public (module), 1

poloniex_aio.trading (module), 3

R

return24hVolume() (in module poloniex_aio.public), 1

returnActiveLoans() (in module poloniex_aio.trading), 10

returnAvailableAccountBalances() (in module poloniex_aio.trading), 7

returnBalances() (in module poloniex_aio.trading), 3

returnChartData() (in module poloniex_aio.public), 2

returnCompleteBalances() (in module poloniex_aio.trading), 3

returnCurrencies() (in module poloniex_aio.public), 2

returnDepositAddresses() (in module poloniex_aio.trading), 4

returnDepositsWithdrawals() (in module poloniex_aio.trading), 4

returnFeeInfo() (in module poloniex_aio.trading), 7

returnLendingHistory() (in module poloniex_aio.trading), 11

returnLoanOrders() (in module poloniex_aio.public), 2

returnMarginAccountSummary() (in module poloniex_aio.trading), 8

returnOpenLoanOffers() (in module poloniex_aio.trading), 10

returnOpenOrders() (in module poloniex_aio.trading), 4

returnOrderBook() (in module poloniex_aio.public), 1

returnOrderTrades() (in module poloniex_aio.trading), 5

returnTicker() (in module poloniex_aio.public), 1

returnTradableBalances() (in module poloniex_aio.trading), 8

returnTradeHistory() (in module poloniex_aio.public), 2

returnTradeHistory() (in module poloniex_aio.trading), 5

S

sell() (in module poloniex_aio.trading), 6

T

toggleAutoRenew() (in module poloniex_aio.trading), 11

transferBalance() (in module poloniex_aio.trading), 8

W

withdraw() (in module poloniex_aio.trading), 7