

CoinHarbour Peatio User API v2

API and SDK Documentation

Version: 3.1.0

API for CoinHarbour Peatio application.

Api

deleteApiV2PeatioAccountBeneficiariesId

Delete beneficiary

DELETE

```
/api/v2/peatio/account/beneficiaries/{id}
```

Usage and SDK Samples

[Curl](#) [Java](#) [Android](#) [Obj-C](#) [JavaScript](#) [C#](#) [PHP](#) [Perl](#) [Python](#)

```
curl -X DELETE \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/beneficiaries/{id}?otp="
```

Parameters

Path parameters

Name	Description
id*	Integer (int32) <i>Beneficiary Identifier in Database</i> Required

Query parameters

Name	Description
otp*	Integer (int32) <i>OTP to perform action</i> Required

Responses

Status: 204 - Delete beneficiary

getApiV2PeatioAccountBalances

Get list of user accounts

GET

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
-H "Accept: application/json" \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/balances?limit=&page=&account_type=&nonzero=&search=&search[currency_code]=
=&search[currency_name]=""
```

Parameters

Query parameters

Name	Description
limit	Integer (int32) <i>Limit the number of returned paginations. Defaults to 100.</i>
page	Integer (int32) <i>Specify the page of paginated results.</i>
account_type	String <i>Accounts type.</i>
nonzero	<i>Filter non zero balances.</i>
search	
search[currency_code]	String
search[currency_name]	String

Responses

Status: 200 - Get list of user accounts

Schema

```
▼ [ ]
  ▼ { }
    Get list of user accounts

    currency:      ▼ [] string
                    Currency code.

    balance:       ▼ [] number (double)
                    Account balance.

    locked:        ▼ [] number (double)
                    Account locked funds.

    account_type:  ▼ [] string
                    Account type.

    deposit_address: ▼ { }
      currencies:   ▼ [ ]
                    Currencies codes.

                    string
      ]
      blockchain_key: ▼ [] string
                    Unique key to identify blockchain.

      address:       ▼ [] string
                    Payment address.
```

```

state:      ▼ [] string
            Payment address state.

}

deposit_addresses: ▼ [ [] ]
                    User deposit addresses

                    ▼ { [] }
                    currencies: ▼ [ [] ]
                        Currencies codes.

                        string
                    ]

blockchain_key: ▼ [] string
                Unique key to identify blockchain.

address:      ▼ [] string
            Payment address.

state:        ▼ [] string
            Payment address state.

}

]

}
]

```

getApiV2PeatioAccountBalancesCurrency

Get user account by currency

GET

```
/api/v2/peatio/account/balances/{currency}
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
-H "Accept: application/json" \
"//www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/balances/{currency}"
```

Parameters

Path parameters

Name	Description
currency*	String <i>The currency code.</i> Required

Responses

Status: 200 - Get user account by currency

```

▼ { []
  Get list of user accounts

  currency:      ▼ [] string
                  Currency code.

  balance:       ▼ [] number (double)
                  Account balance.

  locked:        ▼ [] number (double)
                  Account locked funds.

  account_type:  ▼ [] string
                  Account type.

  deposit_address: ▼ { []
    currencies:   ▼ [ ]
                  Currencies codes.

                  string
                ]

    blockchain_key: ▼ [] string
                  Unique key to identify blockchain.

    address:       ▼ [] string
                  Payment address.

    state:         ▼ [] string
                  Payment address state.

  }

  deposit_addresses: ▼ [ ]
  User deposit addresses

  ▼ { []
    currencies:   ▼ [ ]
                  Currencies codes.

                  string
                ]

    blockchain_key: ▼ [] string
                  Unique key to identify blockchain.

    address:       ▼ [] string
                  Payment address.

    state:         ▼ [] string
                  Payment address state.

  }

]
}

```

getApiV2PeatioAccountBeneficiaries

Get list of user beneficiaries

GET

/api/v2/peatio/account/beneficiaries

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET  
-H "Accept: application/json"  
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/beneficiaries?limit=&page=&cy=&blockchain_key=&state="
```

Parameters

Query parameters

Name	Description
limit	Integer (int32) <i>Limit the number of returned paginations. Defaults to 100.</i>
page	Integer (int32) <i>Specify the page of paginated results.</i>
currency	String <i>Beneficiary currency code.</i>
blockchain_key	String <i>Blockchain key of the requested beneficiary</i>
state	String <i>Defines either beneficiary active - user can use it to withdraw money or pending - requires beneficiary activation with pin.</i>

Responses

Status: 200 - Get list of user beneficiaries

Schema

```
▼ [ ]  
  ▼ { []  
    Get list of user beneficiaries  
      id:          ▼ [] integer (int32)  
                    Beneficiary Identifier in Database  
      blockchain_key: ▼ [] string  
                    Unique key to identify blockchain.  
      protocol:     ▼ [] string  
                    Unique key to identify blockchain.  
      blockchain_name: ▼ [] string  
                    Blockchain name.  
      currency:      ▼ [] string  
                    Beneficiary currency code.  
      uid:           ▼ [] string  
                    Beneficiary owner  
      name:          ▼ [] string  
                    Human rememberable name which refer beneficiary.
```

```

description:    ▼ [] string
                Human rememberable description of beneficiary.

state:          ▼ [] string
                Defines either beneficiary active - user can use it to withdraw money or pending - requires
                beneficiary activation with pin.

withdrawals:    ▼ [] integer (int32)
                Withdrawals count for given beneficiary

sent_at:        ▼ [] string
                Time when last pin was sent

}

]

```

getApiV2PeatioAccountBeneficiariesId

Get beneficiary by ID

GET

```
/api/v2/peatio/account/beneficiaries/{id}
```

Usage and SDK Samples

[Curl](#) [Java](#) [Android](#) [Obj-C](#) [JavaScript](#) [C#](#) [PHP](#) [Perl](#) [Python](#)

```
curl -X GET \
-H "Accept: application/json"
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/beneficiaries/{id}"
```

Parameters

Path parameters

Name	Description
id*	Integer (int32) <i>Beneficiary Identifier in Database</i> Required

Responses

Status: 200 - Get beneficiary by ID

[Schema](#)

```

▼ { []
  Get list of user beneficiaries

  id:          ▼ [] integer (int32)
                Beneficiary Identifier in Database

  blockchain_key: ▼ [] string
                Unique key to identify blockchain.

```

```

protocol:      ▼ [] string
               Unique key to identify blockchain.

blockchain_name: ▼ [] string
                  Blockchain name.

currency:      ▼ [] string
               Beneficiary currency code.

uid:           ▼ [] string
               Beneficiary owner

name:          ▼ [] string
               Human rememberable name which refer beneficiary.

description:   ▼ [] string
               Human rememberable description of beneficiary.

state:         ▼ [] string
               Defines either beneficiary active - user can use it to withdraw money or pending - requires beneficiary activation with pin.

withdrawals:   ▼ [] integer (int32)
               Withdrawals count for given beneficiary

sent_at:       ▼ [] string
               Time when last pin was sent

}

```

getApiV2PeatioAccountDepositAddressCurrency

Returns deposit address for account you want to deposit to by currency. The address may be blank because address generation process is still in progress. If this case you should try again later.

GET

```
/api/v2/peatio/account/deposit_address/{currency}
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
-H "Accept: application/json" \
"//www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/deposit_address/{currency}?blockchain_key=&address_format="
```

Parameters

Path parameters

Name	Description
currency*	String The account you want to deposit to. Required

Query parameters

Name	Description
blockchain_key	String <i>Blockchain key of the requested deposit address</i>
address_format	String <i>Address format legacy/cash</i>

Responses

Status: 200 - Returns deposit address for account you want to deposit to by currency. The address may be blank because address generation process is still in progress. If this case you should try again later.

Schema

```
▼ { □
  Get your deposits history.

  id:      ▼ □ integer (int32)
           Unique deposit id.

  currency: ▼ □ string
           Deposit currency id.

  blockchain_key: ▼ □ string
           Unique key to identify blockchain.

  protocol: ▼ □ string
           Blockchain protocol

  warning: ▼ □ string
           Blockchain warning

  amount: ▼ □ number (double)
           Deposit amount.

  fee: ▼ □ number (double)
           Deposit fee.

  txid: ▼ □ string
           Deposit transaction id.

  confirmations: ▼ □ integer (int32)
           Number of deposit confirmations.

  state: ▼ □ string
           Deposit state.

  transfer_type: ▼ □ string
           Deposit transfer type

  created_at: ▼ □ string
           The datetime when deposit was created.

  completed_at: ▼ □ string
           The datetime when deposit was completed..

  tid: ▼ □ string
           The shared transaction ID

}
```

getApiV2PeatioAccountDeposits

Get your deposits history.

GET

/api/v2/peatio/account/deposits

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
-H "Accept: application/json" \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/deposits?currency=&state=&txid=&blockchain_key=&time_from=&time_to=&limit=&page="
```

Parameters

Query parameters

Name	Description
currency	String <i>Currency code</i>
state	String <i>Filter deposits by states.</i>
txid	String <i>Deposit transaction id.</i>
blockchain_key	String <i>Blockchain key of the requested deposit</i>
time_from	Integer (int32) <i>An integer represents the seconds elapsed since Unix epoch.</i>
time_to	Integer (int32) <i>An integer represents the seconds elapsed since Unix epoch.</i>
limit	Integer (int32) <i>Number of deposits per page (defaults to 100, maximum is 100).</i>
page	Integer (int32) <i>Page number (defaults to 1).</i>

Responses

Status: 200 - Get your deposits history.

Schema

```
▼ [ ]  
  ▼ { []  
    Get your deposits history.  
      id:      ▼ [] integer (int32)  
                Unique deposit id.  
      currency: ▼ [] string  
                Deposit currency id.  
      blockchain_key: ▼ [] string  
                    Unique key to identify blockchain.
```

```

protocol:      ▼ [] string
               Blockchain protocol

warning:       ▼ [] string
               Blockchain warning

amount:        ▼ [] number (double)
               Deposit amount.

fee:           ▼ [] number (double)
               Deposit fee.

txid:          ▼ [] string
               Deposit transaction id.

confirmations: ▼ [] integer (int32)
               Number of deposit confirmations.

state:         ▼ [] string
               Deposit state.

transfer_type: ▼ [] string
               Deposit transfer type

created_at:    ▼ [] string
               The datetime when deposit was created.

completed_at:  ▼ [] string
               The datetime when deposit was completed..

tid:           ▼ [] string
               The shared transaction ID

```

```

}
]
```

getApiV2PeatioAccountDepositsTxid

Get details of specific deposit.

GET

```
/api/v2/peatio/account/deposits/{txid}
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
-H "Accept: application/json" \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/deposits/{txid}"
```

Parameters

Path parameters

Name	Description
------	-------------

txid*	String <i>Deposit transaction id</i> Required
-------	-----------------------------------------------------

Responses

Status: 200 - Get details of specific deposit.

Schema

```

▼ { []
  Get your deposits history.

  id:          ▼ [] integer (int32)
                Unique deposit id.

  currency:    ▼ [] string
                Deposit currency id.

  blockchain_key: ▼ [] string
                Unique key to identify blockchain.

  protocol:    ▼ [] string
                Blockchain protocol

  warning:     ▼ [] string
                Blockchain warning

  amount:       ▼ [] number (double)
                Deposit amount.

  fee:          ▼ [] number (double)
                Deposit fee.

  txid:         ▼ [] string
                Deposit transaction id.

  confirmations: ▼ [] integer (int32)
                Number of deposit confirmations.

  state:        ▼ [] string
                Deposit state.

  transfer_type: ▼ [] string
                Deposit transfer type

  created_at:   ▼ [] string
                The datetime when deposit was created.

  completed_at: ▼ [] string
                The datetime when deposit was completed..

  tid:          ▼ [] string
                The shared transaction ID
}

```

List your internal transfers as paginated collection.

GET

/api/v2/peatio/account/internal_transfers

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
-H "Accept: application/json" \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/internal_transfers?currency=&state=&sender="
```

Parameters

Query parameters

Name	Description
currency	String <i>Currency code.</i>
state	String <i>The state to filter by.</i>
sender	String

Responses

Status: 200 - List your internal transfers as paginated collection.

Schema

```
▼ [ ]  
  ▼ { }  
    List your internal transfers as paginated collection.  
  
    currency:      ▼ [] string  
                  The currency code.  
  
    sender_username: ▼ [] string  
                  The internal transfer sender.  
  
    receiver_username: ▼ [] string  
                  The internal transfer receiver.  
  
    sender_uid:      ▼ [] string  
                  The internal transfer sender.  
  
    receiver_uid:    ▼ [] string  
                  The internal transfer receiver.  
  
    direction:       ▼ [] string  
                  The internal transfer direction (incoming or outgoing internal transfer).  
  
    amount:          ▼ [] number (double)  
                  Internal transfer Amount.  
  
    status:          ▼ [] string  
                  The internal transfer state.  
  
    created_at:      ▼ [] string  
                  The datetimes for the internal transfer.
```

```
    updated_at: ▼ [] string  
        The datetimes for the internal transfer.  
    }  
]
```

getApiV2PeatioAccountMembersMe

Returns current member

GET

```
/api/v2/peatio/account/members/me
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET\  
-H "Accept: application/json"\  
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/members/me"
```

Parameters

Responses

Status: 200 - Returns current member

Schema

```
▼ { []  
  Returns current member  
  uid: ▼ [] string  
      Member UID.  
  email: ▼ [] string  
      Member email.  
  group: ▼ [] string  
      Member's group.  
  beneficiaries_whitelisting: ▼ [] string  
      Member's beneficiaries whitelisting.  
}
```

getApiV2PeatioAccountStatsPnl

Get assets pnl calculated into one currency

GET

```
/api/v2/peatio/account/stats/pnl
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/stats/pnl?pnl_currency="
```

Parameters

Query parameters

Name	Description
pnl_currency	String <i>Currency code in which the PnL is calculated</i>

Responses

Status: 200 - Get assets pnl calculated into one currency

getApiV2PeatioAccountTransactions

Get your transactions history.

GET

```
/api/v2/peatio/account/transactions
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/transactions?currency=&order_by=&time_from=&time_to=&deposit_state=&withdraw_state=&txid=&limit=&page="
```

Parameters

Query parameters

Name	Description
currency	String <i>Currency code</i>
order_by	String <i>Sorting order</i>
time_from	Integer (int32) <i>An integer represents the seconds elapsed since Unix epoch.</i>
time_to	Integer (int32) <i>An integer represents the seconds elapsed since Unix epoch.</i>

deposit_state	String <i>Filter deposits by states.</i>
withdraw_state	String <i>Filter withdraws by states.</i>
txid	String <i>Transaction id.</i>
limit	Integer (int32) <i>Limit the number of returned transactions. Default to 100.</i>
page	Integer (int32) <i>Specify the page of paginated results.</i>

Responses

Status: 200 - Get your transactions history.

getApiV2PeatioAccountWithdraws

List your withdraws as paginated collection.

GET

```
/api/v2/peatio/account/withdraws
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
-H "Accept: application/json"
"//www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/withdraws?currency=&blockchain_key=&limit=&state=&rid=&time_from=&time_to=&
page="
```

Parameters

Query parameters

Name	Description
currency	String <i>Currency code.</i>
blockchain_key	String <i>Blockchain key of the requested withdrawal</i>
limit	Integer (int32) <i>Number of withdraws per page (defaults to 100, maximum is 100).</i>
state	String <i>Filter withdrawals by states.</i>
rid	String <i>Wallet address on the Blockchain.</i>
time_from	Integer (int32) <i>An integer represents the seconds elapsed since Unix epoch.</i>
time_to	Integer (int32) <i>An integer represents the seconds elapsed since Unix epoch.</i>
page	Integer (int32) <i>Page number (defaults to 1).</i>

Responses

Status: 200 - List your withdraws as paginated collection.

Schema

▼ []

▼ { []

List your withdraws as paginated collection.

id: ▼ [] integer (int32)

The withdrawal id.

currency: ▼ [] string
The currency code.

type: ▼ [] string
The withdrawal type

blockchain_key: ▼ [] string
Unique key to identify blockchain.

amount: ▼ [] string
The withdrawal amount

fee: ▼ [] number (double)
The exchange fee.

blockchain_txid: ▼ [] string
The withdrawal transaction id.

rid: ▼ [] string
The beneficiary ID or wallet address on the Blockchain.

protocol: ▼ [] string
Blockchain protocol

state: ▼ [] string
The withdrawal state.

confirmations: ▼ [] integer (int32)
Number of confirmations.

note: ▼ [] string
Withdraw note.

transfer_type: ▼ [] string
Withdraw transfer type

created_at: ▼ [] string
The datetimes for the withdrawal.

updated_at: ▼ [] string
The datetimes for the withdrawal.

done_at: ▼ [] string
The datetime when withdraw was completed

}

]

getApiV2PeatioAccountWithdrawsSums

Returns withdrawal sums for last 24 hours and 1 month

GET

```
/api/v2/peatio/account/withdraws/sums
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/withdraws/sums"
```

Parameters

Responses

Status: 200 - Returns withdrawal sums for last 24 hours and 1 month

getApiV2PeatioCoingeckoHistoricalTrades

Get recent trades on market

GET

```
/api/v2/peatio/coingecko/historical_trades
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/coingecko/historical_trades?ticker_id=&type=&limit=&start_time=&end_time="
```

Parameters

Query parameters

Name	Description
ticker_id*	String <i>A pair such as "LTC_BTC"</i> Required
type	String <i>To indicate nature of trade - buy/sell</i>
limit	Integer (int32) <i>Number of historical trades to retrieve from time of query. [0, 200, 500...]. 0 returns full history</i>
start_time	Integer (int32)
end_time	Integer (int32)

Responses

Status: 200 - Get recent trades on market

getApiV2PeatioCoingeckoOrderbook

Get depth or specified market

GET

```
/api/v2/peatio/coingecko/orderbook
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/coingecko/orderbook?ticker_id=&depth="
```

Parameters

Query parameters

Name	Description
ticker_id*	String <i>A pair such as "LTC_BTC"</i> Required
depth	Integer (int32) <i>Orders depth quantity: [0, 100, 200, 500...]</i>

Responses

Status: 200 - Get depth or specified market

getApiV2PeatioCoingeckoPairs

Get list of all available trading pairs

GET

```
/api/v2/peatio/coingecko/pairs
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/coingecko/pairs"
```

Parameters

Responses

Status: 200 - Get list of all available trading pairs

getApiV2PeatioCoingeckoTickers

Get list of all available trading pairs

GET

```
/api/v2/peatio/coingecko/tickers
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
-H "Accept: application/json"
"//www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/coingecko/tickers"
```

Parameters

Responses

Status: 200 - Get list of all available trading pairs

Schema

▼ { [] }

Get list of all available trading pairs

at: ▼ [] integer (int32)

Timestamp of ticker

ticker: ▼ { [] }

low: ▼ [] number (double)

The lowest trade price during last 24 hours (0.0 if no trades executed during last 24 hours)

high: ▼ [] number (double)

The highest trade price during last 24 hours (0.0 if no trades executed during last 24 hours)

open: ▼ [] number (double)

Price of the first trade executed 24 hours ago or less

last: ▼ [] number (double)

The last executed trade price

volume: ▼ [] number (double)

Total volume of trades executed during last 24 hours

amount: ▼ [] number (double)

Total amount of trades executed during last 24 hours

```

vol:           ▼ [] number (double)
               Alias to volume

avg_price:    ▼ [] number (double)
               Average price more precisely VWAP is calculated by adding up the total traded for
               every transaction(price multiplied by the number of shares traded) and then dividing by
               the total shares traded

price_change_percent: ▼ [] string
               Price change in the next format +3.19%.Price change is calculated using next formula
               (last - open) / open * 100%

at:           ▼ [] integer (int32)
               Timestamp of ticker

}

}

```

getApiV2PeatioCoinmarketcapAssets

Details on crypto currencies available on the exchange

GET

/api/v2/peatio/coinmarketcap/assets

Usage and SDK Samples

[Curl](#) [Java](#) [Android](#) [Obj-C](#) [JavaScript](#) [C#](#) [PHP](#) [Perl](#) [Python](#)

```
curl -X GET \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/coinmarketcap/assets"
```

Parameters

Responses

Status: 200 - Details on crypto currencies available on the exchange

getApiV2PeatioCoinmarketcapOrderbookMarketPair

Get depth or specified market

GET

/api/v2/peatio/coinmarketcap/orderbook/{market_pair}

Usage and SDK Samples

```
curl -X GET \
"//www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/coinmarketcap/orderbook/{market_pair}?depth="
```

Parameters

Path parameters

Name	Description
market_pair*	String <i>A pair such as "LTC_BTC"</i> Required

Query parameters

Name	Description
depth	Integer (int32) <i>Orders depth quantity: [0,5,10,20,50,100,500]</i>

Responses

Status: 200 - Get depth or specified market

getApiV2PeatioCoinmarketcapSummary

Overview of market data for all tickers and all market pairs on the exchange

GET

```
/api/v2/peatio/coinmarketcap/summary
```

Usage and SDK Samples

```
curl -X GET \
"//www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/coinmarketcap/summary"
```

Parameters

Responses

Status: 200 - Overview of market data for all tickers and all market pairs on the exchange

getApiV2PeatioCoinmarketcapTicker

Get 24-hour pricing and volume summary for each market pair

GET

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET\  
-H "Accept: application/json"\  
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/coinmarketcap/ticker"
```

Parameters

Responses

Status: 200 - Get 24-hour pricing and volume summary for each market pair

Schema

```

▼ { [] }
Get list of all available trading pairs

at: ▼ [] integer (int32)
Timestamp of ticker

ticker: ▼ { [] }

low: ▼ [] number (double)
The lowest trade price during last 24 hours (0.0 if no trades executed during last 24 hours)

high: ▼ [] number (double)
The highest trade price during last 24 hours (0.0 if no trades executed during last 24 hours)

open: ▼ [] number (double)
Price of the first trade executed 24 hours ago or less

last: ▼ [] number (double)
The last executed trade price

volume: ▼ [] number (double)
Total volume of trades executed during last 24 hours

amount: ▼ [] number (double)
Total amount of trades executed during last 24 hours

vol: ▼ [] number (double)
Alias to volume

avg_price: ▼ [] number (double)
Average price more precisely VWAP is calculated by adding up the total traded for every transaction(price multiplied by the number of shares traded) and then dividing by the total shares traded

price_change_percent: ▼ [] string
Price change in the next format +3.19%. Price change is calculated using next formula (last - open) / open * 100%

at: ▼ [] integer (int32)
Timestamp of ticker
}

}

```

getApiV2PeatioCoinmarketcapTradesMarketPair

Get recent trades on market

GET

```
/api/v2/peatio/coinmarketcap/trades/{market_pair}
```

Usage and SDK Samples

```
curl -X GET  
-H "Accept: application/json"  
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/coinmarketcap/trades/{market_pair}"
```

Parameters

Path parameters

Name	Description
market_pair*	String <i>A pair such as "LTC_BTC"</i> Required

Responses

Status: 200 - Get recent trades on market

[Schema](#)

▼ { []

Get recent trades on market

id: ▼ [] string
Trade ID.

price: ▼ [] number (double)
Trade price.

amount: ▼ [] number (double)
Trade amount.

total: ▼ [] number (double)
*Trade total (Amount * Price).*

fee_currency: ▼ [] number (double)
Currency user's fees were charged in.

fee: ▼ [] number (double)
Percentage of fee user was charged for performed trade.

fee_amount: ▼ [] number (double)
Amount of fee user was charged for performed trade.

market: ▼ [] string
Trade market id.

market_type: ▼ [] string
Market type.

created_at: ▼ [] string
Trade create time in iso8601 format.

taker_type: ▼ [] string
Trade taker order type (sell or buy).

side: ▼ [] string
Trade side.

order_id: ▼ [] integer (int32)
Order id.

}

getApiV2PeatioMarketOrders

Get your orders, result is paginated.

GET

/api/v2/peatio/market/orders

Usage and SDK Samples

[Curl](#) [Java](#) [Android](#) [Obj-C](#) [JavaScript](#) [C#](#) [PHP](#) [Perl](#) [Python](#)

```

curl -X GET \
-H "Accept: application/json"
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/market/orders?market=&market_type=&base_unit="e_unit=&state=&limit=&page=&order_by
=&ord_type=&type=&time_from=&time_to="

```

Parameters

Query parameters

Name	Description
market	String
market_type	String
base_unit	String
quote_unit	String
state	String <i>Filter order by state.</i>
limit	Integer (int32) <i>Limit the number of returned orders, default to 100.</i>
page	Integer (int32) <i>Specify the page of paginated results.</i>
order_by	String <i>If set, returned orders will be sorted in specific order, default to "desc".</i>
ord_type	String <i>Filter order by ord_type.</i>
type	String <i>Filter order by type.</i>
time_from	Integer (int32) <i>An integer represents the seconds elapsed since Unix epoch.If set, only orders created after the time will be returned.</i>
time_to	Integer (int32) <i>An integer represents the seconds elapsed since Unix epoch.If set, only orders created before the time will be returned.</i>

Responses

Status: 200 - Get your orders, result is paginated.

Schema

```

▼ [ ]
  ▼ { }
    Get your orders, result is paginated.

    id:          ▼ [] integer (int32)
                  Unique order id.

    uuid:        ▼ [] string
                  Unique order UUID.

    side:        ▼ [] string
                  Either 'sell' or 'buy'.

    ord_type:    ▼ [] string
                  Type of order, either 'limit' or 'market'.

    price:       ▼ [] number (double)
                  Price for each unit. e.g. If you want to sell/buy 1 btc at 3000 usd, the price is '3000.0'

```

avg_price:	▼ [] number (double) <i>Average execution price, average of price in trades.</i>
state:	▼ [] string <i>One of 'wait', 'done', or 'cancel'. An order in 'wait' is an active order, waiting fulfillment; a 'done' order is an order fulfilled; 'cancel' means the order has been canceled.</i>
market:	▼ [] string <i>The market in which the order is placed, e.g. 'btcusd'. All available markets can be found at /api/v2/markets.</i>
market_type:	▼ [] string <i>Market type.</i>
created_at:	▼ [] string <i>Order create time in iso8601 format.</i>
updated_at:	▼ [] string <i>Order updated time in iso8601 format.</i>
origin_volume:	▼ [] number (double) <i>The amount user want to sell/buy. An order could be partially executed, e.g. an order sell 5 btc can be matched with a buy 3 btc order, left 2 btc to be sold; in this case the order's volume would be '5.0', its remaining_volume would be '2.0', its executed volume is '3.0'.</i>
remaining_volume:	▼ [] number (double) <i>The remaining volume, see 'volume'.</i>
executed_volume:	▼ [] number (double) <i>The executed volume, see 'volume'.</i>
maker_fee:	▼ [] number (double) <i>Fee for maker.</i>
taker_fee:	▼ [] number (double) <i>Fee for taker.</i>
trades_count:	▼ [] integer (int32) <i>Count of trades.</i>
trades:	▼ [] <i>Trades wiht this order.</i>
	▼ { [] <i>Get recent trades on market</i>
id:	▼ [] string <i>Trade ID.</i>
price:	▼ [] number (double) <i>Trade price.</i>
amount:	▼ [] number (double) <i>Trade amount.</i>
total:	▼ [] number (double) <i>Trade total (Amount * Price).</i>
fee_currency:	▼ [] number (double) <i>Currency user's fees were charged in.</i>
fee:	▼ [] number (double) <i>Percentage of fee user was charged for performed trade.</i>
fee_amount:	▼ [] number (double) <i>Amount of fee user was charged for performed trade.</i>

```

    market:     ▼ [] string
                Trade market id.

    market_type: ▼ [] string
                  Market type.

    created_at:  ▼ [] string
                  Trade create time in iso8601 format.

    taker_type:  ▼ [] string
                  Trade taker order type (sell or buy).

    side:        ▼ [] string
                  Trade side.

    order_id:    ▼ [] integer (int32)
                  Order id.

}
]
}
]
```

getApiV2PeatioMarketOrdersId

Get information of specified order.

GET

```
/api/v2/peatio/market/orders/{id}
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
-H "Accept: application/json" \
"//www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/market/orders/{id}"
```

Parameters

Path parameters

Name	Description
id*	String Required

Responses

Status: 200 - Get information of specified order.

Schema

```
▼ { [] }
Get your orders, result is paginated.
```

id:	▼ [] integer (int32) <i>Unique order id.</i>
uuid:	▼ [] string <i>Unique order UUID.</i>
side:	▼ [] string <i>Either 'sell' or 'buy'.</i>
ord_type:	▼ [] string <i>Type of order, either 'limit' or 'market'.</i>
price:	▼ [] number (double) <i>Price for each unit. e.g. If you want to sell/buy 1 btc at 3000 usd, the price is '3000.0'</i>
avg_price:	▼ [] number (double) <i>Average execution price, average of price in trades.</i>
state:	▼ [] string <i>One of 'wait', 'done', or 'cancel'. An order in 'wait' is an active order, waiting fulfillment; a 'done' order is an order fulfilled; 'cancel' means the order has been canceled.</i>
market:	▼ [] string <i>The market in which the order is placed, e.g. 'btcusd'. All available markets can be found at /api/v2/markets.</i>
market_type:	▼ [] string <i>Market type.</i>
created_at:	▼ [] string <i>Order create time in iso8601 format.</i>
updated_at:	▼ [] string <i>Order updated time in iso8601 format.</i>
origin_volume:	▼ [] number (double) <i>The amount user want to sell/buy. An order could be partially executed, e.g. an order sell 5 btc can be matched with a buy 3 btc order, left 2 btc to be sold; in this case the order's volume would be '5.0', its remaining_volume would be '2.0', its executed volume is '3.0'.</i>
remaining_volume:	▼ [] number (double) <i>The remaining volume, see 'volume'.</i>
executed_volume:	▼ [] number (double) <i>The executed volume, see 'volume'.</i>
maker_fee:	▼ [] number (double) <i>Fee for maker.</i>
taker_fee:	▼ [] number (double) <i>Fee for taker.</i>
trades_count:	▼ [] integer (int32) <i>Count of trades.</i>
trades:	▼ [[] <i>Trades wiht this order.</i>
	▼ { [] <i>Get recent trades on market</i>
id:	▼ [] string <i>Trade ID.</i>
price:	▼ [] number (double) <i>Trade price.</i>

```

amount:      ▼ [] number (double)
              Trade amount.

total:       ▼ [] number (double)
              Trade total (Amount * Price).

fee_currency: ▼ [] number (double)
              Currency user's fees were charged in.

fee:         ▼ [] number (double)
              Percentage of fee user was charged for performed trade.

fee_amount:  ▼ [] number (double)
              Amount of fee user was charged for performed trade.

market:      ▼ [] string
              Trade market id.

market_type: ▼ [] string
              Market type.

created_at:  ▼ [] string
              Trade create time in iso8601 format.

taker_type:  ▼ [] string
              Trade taker order type (sell or buy).

side:        ▼ [] string
              Trade side.

order_id:    ▼ [] integer (int32)
              Order id.

}

]
}

```

getApiV2PeatioMarketTrades

Get your executed trades. Trades are sorted in reverse creation order.

GET

/api/v2/peatio/market/trades

Usage and SDK Samples

[Curl](#) [Java](#) [Android](#) [Obj-C](#) [JavaScript](#) [C#](#) [PHP](#) [Perl](#) [Python](#)

```

curl -X GET \
-H "Accept: application/json" \
//www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/market/trades?market=&market_type=&limit=&page=&type=&time_from=&time_to=&order_b
y="

```

Parameters

Query parameters

Name	Description
market	String
market_type	String
limit	Integer (int32) <i>Limit the number of returned trades. Default to 100.</i>
page	Integer (int32) <i>Specify the page of paginated results.</i>
type	String <i>To indicate nature of trade - buy/sell</i>
time_from	Integer (int32) <i>An integer represents the seconds elapsed since Unix epoch.If set, only trades executed after the time will be returned.</i>
time_to	Integer (int32) <i>An integer represents the seconds elapsed since Unix epoch.If set, only trades executed before the time will be returned.</i>
order_by	String <i>If set, returned trades will be sorted in specific order, default to 'desc'.</i>

Responses

Status: 200 - Get your executed trades. Trades are sorted in reverse creation order.

Schema

▼ []
▼ { []
Get recent trades on market

id: ▼ [] string
Trade ID.

price: ▼ [] number (double)
Trade price.

amount: ▼ [] number (double)
Trade amount.

total: ▼ [] number (double)
*Trade total (Amount * Price).*

fee_currency: ▼ [] number (double)
Currency user's fees were charged in.

fee: ▼ [] number (double)
Percentage of fee user was charged for performed trade.

fee_amount: ▼ [] number (double)
Amount of fee user was charged for performed trade.

market: ▼ [] string
Trade market id.

market_type: ▼ [] string
Market type.

created_at: ▼ [] string
Trade create time in iso8601 format.

taker_type: ▼ [] string

Trade taker order type (sell or buy).

side: ▼ string

Trade side.

order_id: ▼ integer (int32)

Order id.

}

]

getApiV2PeatioPublicConfig

GET

```
/api/v2/peatio/public/config
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET\\\n"//www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/public/config"
```

Parameters

Responses

Status: 200 - get Config(s)

getApiV2PeatioPublicCurrencies

Get list of currencies

GET

```
/api/v2/peatio/public/currencies
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET\\\n-H "Accept: application/json"\\\n"//www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/public/currencies?limit=&page=&type=&search=&search[code]=&search[name]+"
```

Parameters

Query parameters

Name	Description
limit	Integer (int32) <i>Limit the number of returned paginations. Defaults to 100.</i>
page	Integer (int32) <i>Specify the page of paginated results.</i>
type	String <i>Currency type</i>
search	
search[code]	String <i>Search by currency code using SQL LIKE</i>
search[name]	String <i>Search by currency name using SQL LIKE</i>

Responses

Status: 200 - Get list of currencies

Schema

```

▼ [ ]
  ▼ { }
    Get a currency

    id:           ▼ [] string
                  Currency code.

    name:          ▼ [] string
                  Currency name

    description:   ▼ [] string
                  Currency description

    homepage:      ▼ [] string
                  Currency homepage

    parent_id:     ▼ [] string
                  Currency parent id

    price:          ▼ [] string
                  Currency current price

    explorer_transaction: ▼ [] string
                          Currency transaction exprorer url template
                          example: https://testnet.blockchain.info/tx/

    explorer_address:  ▼ [] string
                        Currency address exprorer url template
                        example: https://testnet.blockchain.info/address/

    type:           ▼ [] string
                  Currency type

    deposit_enabled: ▼ [] string
                      Currency deposit possibility status (true/false).

    withdrawal_enabled: ▼ [] string
                         Currency withdrawal possibility status (true/false).

    deposit_fee:    ▼ [] string
                  Currency deposit fee
  
```

```

min_deposit_amount: ▼ [] string
                    Minimal deposit amount

withdraw_fee:      ▼ [] string
                    Currency withdraw fee

min_withdraw_amount: ▼ [] string
                     Minimal withdraw amount

base_factor:       ▼ [] string
                    Currency base factor

precision:         ▼ [] string
                    Currency precision

position:          ▼ [] string
                    Position used for defining currencies order

icon_url:          ▼ [] string
                    Currency icon

example: https://upload.wikimedia.org/wikipedia/commons/0/05/Ethereum_logo_2014.svg

min_confirmations: ▼ [] string
                    Number of confirmations required for confirming deposit or withdrawal

}

]

```

getApiV2PeatioPublicCurrenciesId

Get a currency

GET

```
/api/v2/peatio/public/currencies/{id}
```

Usage and SDK Samples

[Curl](#) [Java](#) [Android](#) [Obj-C](#) [JavaScript](#) [C#](#) [PHP](#) [Perl](#) [Python](#)

```
curl -X GET \
-H "Accept: application/json"\n"//www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/public/currencies/{id}"
```

Parameters

Path parameters

Name	Description
id*	String Currency code. Required

Responses

Status: 200 - Get a currency

▼ { [] }

Get a currency

id:	▼ [] string <i>Currency code.</i>
name:	▼ [] string <i>Currency name</i>
description:	▼ [] string <i>Currency description</i>
homepage:	▼ [] string <i>Currency homepage</i>
parent_id:	▼ [] string <i>Currency parent id</i>
price:	▼ [] string <i>Currency current price</i>
explorer_transaction:	▼ [] string <i>Currency transaction explorer url template</i> example: https://testnet.blockchain.info/tx/
explorer_address:	▼ [] string <i>Currency address explorer url template</i> example: https://testnet.blockchain.info/address/
type:	▼ [] string <i>Currency type</i>
deposit_enabled:	▼ [] string <i>Currency deposit possibility status (true/false).</i>
withdrawal_enabled:	▼ [] string <i>Currency withdrawal possibility status (true/false).</i>
deposit_fee:	▼ [] string <i>Currency deposit fee</i>
min_deposit_amount:	▼ [] string <i>Minimal deposit amount</i>
withdraw_fee:	▼ [] string <i>Currency withdraw fee</i>
min_withdraw_amount:	▼ [] string <i>Minimal withdraw amount</i>
base_factor:	▼ [] string <i>Currency base factor</i>
precision:	▼ [] string <i>Currency precision</i>
position:	▼ [] string <i>Position used for defining currencies order</i>
icon_url:	▼ [] string <i>Currency icon</i> example: https://upload.wikimedia.org/wikipedia/commons/0/05/Ethereum_logo_2014.svg

```
min_confirmations: ▼ [] string  
Number of confirmations required for confirming deposit or withdrawal  
}
```

getApiV2PeatioPublicHealthAlive

Get application liveness status

GET

```
/api/v2/peatio/public/health/alive
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET\  
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/public/health/alive"
```

Parameters

Responses

Status: 200 - Get application liveness status

getApiV2PeatioPublicHealthReady

Get application readiness status

GET

```
/api/v2/peatio/public/health/ready
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET\  
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/public/health/ready"
```

Parameters

Responses

Status: 200 - Get application readiness status

getApiV2PeatioPublicMarkets

Get all available markets.

GET

```
/api/v2/peatio/public/markets
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET  
-H "Accept: application/json"\n"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/public/markets?limit=&page=&ordering=&order_by=&base_unit="e_unit=&type=&search=&search[base_code]=&search[quote_code]=&search[base_name]=&search[quote_name]=""
```

Parameters

Query parameters

Name	Description
limit	Integer (int32) <i>Limit the number of returned paginations. Defaults to 100.</i>
page	Integer (int32) <i>Specify the page of paginated results.</i>
ordering	String <i>If set, returned values will be sorted in specific order, defaults to 'asc'.</i>
order_by	String <i>Name of the field, which result will be ordered by.</i>
base_unit	String <i>Strict filter for base unit</i>
quote_unit	String <i>Strict filter for quote unit</i>
type	String <i>Strict filter for market type</i>
search	
search[base_code]	String <i>Search base currency code using LIKE</i>
search[quote_code]	String <i>Search quote currency code using LIKE</i>
search[base_name]	String <i>Search base currency name using LIKE</i>
search[quote_name]	String <i>Search quote currency name using LIKE</i>

Responses

Status: 200 - Get all available markets.

Schema

```
▼ [ ]  
  ▼ { []  
    Get all available markets.
```

```

symbol:      ▼ [] string
             Unique market ticker symbol. It's always in the form of xxxyyy, where xxx is the base currency code, yyy is the quote currency code, e.g. 'btcusd'. All available markets can be found at /api/v2/markets.

name:        ▼ [] string
             Market name.

type:        ▼ [] string
             Market type.

base_unit:   ▼ [] string
             Market Base unit.

quote_unit:  ▼ [] string
             Market Quote unit.

min_price:   ▼ [] number (double)
             Minimum order price.

max_price:   ▼ [] number (double)
             Maximum order price.

min_amount:  ▼ [] number (double)
             Minimum order amount.

amount_precision: ▼ [] number (double)
                  Precision for order amount.

price_precision: ▼ [] number (double)
                  Precision for order price.

state:       ▼ [] string
             Market state defines if user can see/trade on current market.

}

]

```

getApiV2PeatioPublicMarketsMarketDepth

Get depth or specified market. Both asks and bids are sorted from highest price to lowest.

GET

```
/api/v2/peatio/public/markets/{market}/depth
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
"//www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/public/markets/{market}/depth?limit="
```

Parameters

Path parameters

Name	Description
market*	String Required

Query parameters

Name	Description
limit	Integer (int32) <i>Limit the number of returned price levels. Default to 300.</i>

Responses

Status: 200 - Get depth or specified market. Both asks and bids are sorted from highest price to lowest.

getApiV2PeatioPublicMarketsMarketKLine

Get OHLC(k line) of specific market.

GET

```
/api/v2/peatio/public/markets/{market}/k-line
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/public/markets/{market}/k-line?period=&time_from=&time_to=&limit="
```

Parameters

Path parameters

Name	Description
market*	String Required

Query parameters

Name	Description
period	Integer (int32) <i>Time period of K line, default to 1. You can choose between 1, 5, 15, 30, 60, 120, 240, 360, 720, 1440, 4320, 10080</i>
time_from	Integer (int32) <i>An integer represents the seconds elapsed since Unix epoch. If set, only k-line data after that time will be returned.</i>
time_to	Integer (int32) <i>An integer represents the seconds elapsed since Unix epoch. If set, only k-line data till that time will be returned.</i>
limit	Integer (int32) <i>Limit the number of returned data points default to 30. Ignored if time_from and time_to are given.</i>

Responses

Status: 200 - Get OHLC(k line) of specific market.

getApiV2PeatioPublicMarketsMarketOrderBook

Get the order book of specified market.

GET

```
/api/v2/peatio/public/markets/{market}/order-book
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET  
-H "Accept: application/json"\n"//www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/public/markets/{market}/order-book?asks_limit=&bids_limit="
```

Parameters

Path parameters

Name	Description
market*	String Required

Query parameters

Name	Description
asks_limit	Integer (int32) <i>Limit the number of returned sell orders. Default to 20.</i>
bids_limit	Integer (int32) <i>Limit the number of returned buy orders. Default to 20.</i>

Responses

Status: 200 - Get the order book of specified market.

Schema

```
▼ [ ]  
  ▼ { []  
    Get the order book of specified market.  
  
    asks: ▼ [ ]  
      Asks in orderbook  
  
      ▼ { []  
        Get your orders, result is paginated.  
  
        id: ▼ [ ] integer (int32)  
          Unique order id.  
  
        uuid: ▼ [ ] string  
          Unique order UUID.  
  
        side: ▼ [ ] string  
          Either 'sell' or 'buy'.
```

ord_type:	▼ [] string <i>Type of order, either 'limit' or 'market'.</i>
price:	▼ [] number (double) <i>Price for each unit. e.g.If you want to sell/buy 1 btc at 3000 usd, the price is '3000.0'</i>
avg_price:	▼ [] number (double) <i>Average execution price, average of price in trades.</i>
state:	▼ [] string <i>One of 'wait', 'done', or 'cancel'.An order in 'wait' is an active order, waiting fulfillment;a 'done' order is an order fulfilled;'cancel' means the order has been canceled.</i>
market:	▼ [] string <i>The market in which the order is placed, e.g. 'btcusd'.All available markets can be found at /api/v2/markets.</i>
market_type:	▼ [] string <i>Market type.</i>
created_at:	▼ [] string <i>Order create time in iso8601 format.</i>
updated_at:	▼ [] string <i>Order updated time in iso8601 format.</i>
origin_volume:	▼ [] number (double) <i>The amount user want to sell/buy.An order could be partially executed,e.g. an order sell 5 btc can be matched with a buy 3 btc order, left 2 btc to be sold; in this case the order's volume would be '5.0', its remaining_volume would be '2.0', its executed volume is '3.0'.</i>
remaining_volume:	▼ [] number (double) <i>The remaining volume, see 'volume'.</i>
executed_volume:	▼ [] number (double) <i>The executed volume, see 'volume'.</i>
maker_fee:	▼ [] number (double) <i>Fee for maker.</i>
taker_fee:	▼ [] number (double) <i>Fee for taker.</i>
trades_count:	▼ [] integer (int32) <i>Count of trades.</i>
trades:	▼ [[] <i>Trades wiht this order.</i> ▼ { [] <i>Get recent trades on market</i> id: ▼ [] string <i>Trade ID.</i> price: ▼ [] number (double) <i>Trade price.</i> amount: ▼ [] number (double) <i>Trade amount.</i> total: ▼ [] number (double) <i>Trade total (Amount * Price).</i> fee_currency: ▼ [] number (double)

Currency user's fees were charged in.

fee: ▶ [] number (double)
Percentage of fee user was charged for performed trade.

fee_amount: ▶ [] number (double)
Amount of fee user was charged for performed trade.

market: ▶ [] string
Trade market id.

market_type: ▶ [] string
Market type.

created_at: ▶ [] string
Trade create time in iso8601 format.

taker_type: ▶ [] string
Trade taker order type (sell or buy).

side: ▶ [] string
Trade side.

order_id: ▶ [] integer (int32)
Order id.

}

]

}

]

bids: ▶ [[]
Bids in orderbook

▼ { []
Get your orders, result is paginated.

id: ▶ [] integer (int32)
Unique order id.

uuid: ▶ [] string
Unique order UUID.

side: ▶ [] string
Either 'sell' or 'buy'.

ord_type: ▶ [] string
Type of order, either 'limit' or 'market'.

price: ▶ [] number (double)
Price for each unit. e.g. If you want to sell/buy 1 btc at 3000 usd, the price is '3000.0'

avg_price: ▶ [] number (double)
Average execution price, average of price in trades.

state: ▶ [] string
One of 'wait', 'done', or 'cancel'. An order in 'wait' is an active order, waiting fulfillment; a 'done' order is an order fulfilled; 'cancel' means the order has been canceled.

market: ▶ [] string
The market in which the order is placed, e.g. 'btcusd'. All available markets can be found at /api/v2/markets.

market_type: ▶ [] string
Market type.

created_at:	▼ [] string <i>Order create time in iso8601 format.</i>
updated_at:	▼ [] string <i>Order updated time in iso8601 format.</i>
origin_volume:	▼ [] number (double) <i>The amount user want to sell/buy.An order could be partially executed,e.g. an order sell 5 btc can be matched with a buy 3 btc order, left 2 btc to be sold; in this case the order's volume would be '5.0',its remaining_volume would be '2.0', its executed volume is '3.0'.</i>
remaining_volume:	▼ [] number (double) <i>The remaining volume, see 'volume'.</i>
executed_volume:	▼ [] number (double) <i>The executed volume, see 'volume'.</i>
maker_fee:	▼ [] number (double) <i>Fee for maker.</i>
taker_fee:	▼ [] number (double) <i>Fee for taker.</i>
trades_count:	▼ [] integer (int32) <i>Count of trades.</i>
trades:	▼ [[] <i>Trades with this order.</i> ▼ { [] <i>Get recent trades on market</i> id: ▼ [] string <i>Trade ID.</i> price: ▼ [] number (double) <i>Trade price.</i> amount: ▼ [] number (double) <i>Trade amount.</i> total: ▼ [] number (double) <i>Trade total (Amount * Price).</i> fee_currency: ▼ [] number (double) <i>Currency user's fees were charged in.</i> fee: ▼ [] number (double) <i>Percentage of fee user was charged for performed trade.</i> fee_amount: ▼ [] number (double) <i>Amount of fee user was charged for performed trade.</i> market: ▼ [] string <i>Trade market id.</i> market_type: ▼ [] string <i>Market type.</i> created_at: ▼ [] string <i>Trade create time in iso8601 format.</i> taker_type: ▼ [] string <i>Trade taker order type (sell or buy).</i>

```

    side:      ▼ [] string
               Trade side.

    order_id: ▼ [] integer (int32)
               Order id.

}
]
}
]

```

getApiV2PeatioPublicMarketsMarketTickers

Get ticker of specific market.

GET

/api/v2/peatio/public/markets/{market}/tickers

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```

curl -X GET \
-H "Accept: application/json" \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/public/markets/{market}/tickers"

```

Parameters

Path parameters

Name	Description
market*	String Required

Responses

Status: 200 - Get ticker of specific market.

Schema

```

▼ { []
  Get list of all available trading pairs

  at:   ▼ [] integer (int32)
        Timestamp of ticker

  ticker: ▼ { []
    low:          ▼ [] number (double)
                  The lowest trade price during last 24 hours (0.0 if no trades executed during last 24 hours)

    high:         ▼ [] number (double)

```

```

    The highest trade price during last 24 hours (0.0 if no trades executed during last 24
    hours)

open: ▾ [] number (double)
      Price of the first trade executed 24 hours ago or less

last: ▾ [] number (double)
      The last executed trade price

volume: ▾ [] number (double)
        Total volume of trades executed during last 24 hours

amount: ▾ [] number (double)
        Total amount of trades executed during last 24 hours

vol: ▾ [] number (double)
     Alias to volume

avg_price: ▾ [] number (double)
            Average price more precisely VWAP is calculated by adding up the total traded for
            every transaction(price multiplied by the number of shares traded) and then dividing by
            the total shares traded

price_change_percent: ▾ [] string
            Price change in the next format +3.19%.Price change is calculated using next formula
            (last - open) / open * 100%

at: ▾ [] integer (int32)
     Timestamp of ticker

}

}

```

getApiV2PeatioPublicMarketsMarketTrades

Get recent trades on market, each trade is included only once. Trades are sorted in reverse creation order.

GET

```
/api/v2/peatio/public/markets/{market}/trades
```

Usage and SDK Samples

[Curl](#) [Java](#) [Android](#) [Obj-C](#) [JavaScript](#) [C#](#) [PHP](#) [Perl](#) [Python](#)

```
curl -X GET \
-H "Accept: application/json" \
"//www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/public/markets/{market}/trades?limit=xtamp=&order_by="
```

Parameters

Path parameters

Name	Description
market*	String Required

Query parameters

Name	Description
limit	Integer (int32) <i>Limit the number of returned trades. Default to 100.</i>
timestamp	Integer (int32) <i>An integer represents the seconds elapsed since Unix epoch. If set, only trades executed before the time will be returned.</i>
order_by	String <i>If set, returned trades will be sorted in specific order, default to 'desc'.</i>

Responses

Status: 200 - Get recent trades on market, each trade is included only once. Trades are sorted in reverse creation order.

Schema

```

▼ [ ]
  ▼ { []
    Get recent trades on market

    id:      ▼ [] string
             Trade ID.

    price:   ▼ [] number (double)
             Trade price.

    amount:  ▼ [] number (double)
             Trade amount.

    total:   ▼ [] number (double)
             Trade total (Amount * Price).

    fee_currency: ▼ [] number (double)
                  Currency user's fees were charged in.

    fee:     ▼ [] number (double)
             Percentage of fee user was charged for performed trade.

    fee_amount: ▼ [] number (double)
                 Amount of fee user was charged for performed trade.

    market:   ▼ [] string
              Trade market id.

    market_type: ▼ [] string
                  Market type.

    created_at: ▼ [] string
                 Trade create time in iso8601 format.

    taker_type: ▼ [] string
                 Trade taker order type (sell or buy).

    side:     ▼ [] string
              Trade side.

    order_id:  ▼ [] integer (int32)
               Order id.

  }
]

```

getApiV2PeatioPublicMarketsTickers

Get ticker of all markets (For response doc see /:market/tickers/ response).

GET

/api/v2/peatio/public/markets/tickers

Usage and SDK Samples

```
curl -X GET \
-H "Accept: application/json" \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/public/markets/tickers"
```

Parameters

Responses

Status: 200 - Get ticker of all markets (For response doc see `/:market/tickers/` response).

Schema

```
▼ { [] }
  Get list of all available trading pairs

  at:    ▼ [] integer (int32)
         Timestamp of ticker

  ticker: ▼ { [] }
    low:          ▼ [] number (double)
                  The lowest trade price during last 24 hours (0.0 if no trades executed during last 24 hours)

    high:         ▼ [] number (double)
                  The highest trade price during last 24 hours (0.0 if no trades executed during last 24 hours)

    open:         ▼ [] number (double)
                  Price of the first trade executed 24 hours ago or less

    last:         ▼ [] number (double)
                  The last executed trade price

    volume:       ▼ [] number (double)
                  Total volume of trades executed during last 24 hours

    amount:       ▼ [] number (double)
                  Total amount of trades executed during last 24 hours

    vol:          ▼ [] number (double)
                  Alias to volume

    avg_price:   ▼ [] number (double)
                  Average price more precisely VWAP is calculated by adding up the total traded for every transaction(price multiplied by the number of shares traded) and then dividing by the total shares traded

    price_change_percent: ▼ [] string
                          Price change in the next format +3.19%.Price change is calculated using next formula (last - open) / open * 100%

    at:           ▼ [] integer (int32)
                  Timestamp of ticker

  }
```

getApiV2PeatioPublicMemberLevels

Returns hash of minimum levels and the privileges they provide.

GET

```
/api/v2/peatio/public/member-levels
```

Usage and SDK Samples

[Curl](#) [Java](#) [Android](#) [Obj-C](#) [JavaScript](#) [C#](#) [PHP](#) [Perl](#) [Python](#)

```
curl -X GET \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/public/member-levels"
```

Parameters

Responses

Status: 200 - Returns hash of minimum levels and the privileges they provide.

getApiV2PeatioPublicTimestamp

Get server current time, in seconds since Unix epoch.

GET

```
/api/v2/peatio/public/timestamp
```

Usage and SDK Samples

[Curl](#) [Java](#) [Android](#) [Obj-C](#) [JavaScript](#) [C#](#) [PHP](#) [Perl](#) [Python](#)

```
curl -X GET \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/public/timestamp"
```

Parameters

Responses

Status: 200 - Get server current time, in seconds since Unix epoch.

getApiV2PeatioPublicTradingFees

Returns trading_fees table as paginated collection

GET

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
-H "Accept: application/json" \
"//www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/public/trading_fees?group=&market_id=&market_type=&limit=&page=&ordering=&order_by=
="
```

Parameters

Query parameters

Name	Description
group	String <i>Member group for define maker/taker fee.</i>
market_id	String <i>Market id for define maker/taker fee.</i>
market_type	String
limit	Integer (int32) <i>Limit the number of returned paginations. Defaults to 100.</i>
page	Integer (int32) <i>Specify the page of paginated results.</i>
ordering	String <i>If set, returned values will be sorted in specific order, defaults to 'desc'.</i>
order_by	String <i>Name of the field, which result will be ordered by.</i>

Responses

Status: 200 - Returns trading_fees table as paginated collection

Schema

```
▼ [ ]
  ▼ { }
    Returns trading_fees table as paginated collection

    id:      ▼ [] integer (int32)
             Unique trading fee table identifier in database.

    group:   ▼ [] string
             Member group for define maker/taker fee.

    market_id: ▼ [] string
             Market id for define maker/taker fee.

    market_type: ▼ [] string
             Market type.

    maker:    ▼ [] number (double)
             Market maker fee.

    taker:    ▼ [] number (double)
             Market taker fee.

    created_at: ▼ [] string
             Trading fee table created time in iso8601 format.
```

```
updated_at: ▾ [] string
  Trading fee table updated time in iso8601 format.

}
]
```

getApiV2PeatioPublicVersion

Get running Peatio version and build details.

GET

```
/api/v2/peatio/public/version
```

Usage and SDK Samples

[Curl](#) [Java](#) [Android](#) [Obj-C](#) [JavaScript](#) [C#](#) [PHP](#) [Perl](#) [Python](#)

```
curl -X GET\
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/public/version"
```

Parameters

Responses

Status: 200 - Get running Peatio version and build details.

getApiV2PeatioPublicWithdrawLimits

Returns withdraw limits table as paginated collection

GET

```
/api/v2/peatio/public/withdraw_limits
```

Usage and SDK Samples

[Curl](#) [Java](#) [Android](#) [Obj-C](#) [JavaScript](#) [C#](#) [PHP](#) [Perl](#) [Python](#)

```
curl -X GET\
-H "Accept: application/json"\
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/public/withdraw_limits?group=&kyc_level=&limit=&page=&ordering=&order_by="
```

Parameters

Query parameters

Name	Description
------	-------------

group	String <i>Member group for define withdraw limits.</i>
kyc_level	String <i>KYC level for define withdraw limits.</i>
limit	Integer (int32) <i>Limit the number of returned paginations. Defaults to 100.</i>
page	Integer (int32) <i>Specify the page of paginated results.</i>
ordering	String <i>If set, returned values will be sorted in specific order, defaults to 'desc'.</i>
order_by	String <i>Name of the field, which result will be ordered by.</i>

Responses

Status: 200 - Returns withdraw limits table as paginated collection

Schema

```

▼ [ ]
  ▼ { }
    Returns withdraw limits table as paginated collection

    id:      ▼ [] integer (int32)
             Unique withdraw limit table identifier in database.

    group:   ▼ [] string
             Member group for define withdraw limits.

    kyc_level: ▼ [] string
             KYC level for define withdraw limits.

    limit_24_hour: ▼ [] number (double)
                  24 hours withdraw limit.

    limit_1_month: ▼ [] number (double)
                  1 month withdraw limit.

    created_at:  ▼ [] string
                 Withdraw limit table created time in iso8601 format.

    updated_at:  ▼ [] string
                 Withdraw limit table updated time in iso8601 format.

  }

]

```

patchApiV2PeatioAccountBeneficiariesIdActivate

Activates beneficiary with pin

PATCH

/api/v2/peatio/account/beneficiaries/{id}/activate

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X PATCH \
-H "Accept: application/json"
-H "Content-Type: application/json"
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/beneficiaries/{id}/activate"
```

Parameters

Path parameters

Name	Description
id*	Integer (int32) <i>Beneficiary Identifier in Database</i> Required

Body parameters

Name	Description
body *	▼ { [] } Required: pin pin: ▼ [] integer (int32) <i>Pin code for beneficiary activation</i> }

Responses

Status: 200 - Activates beneficiary with pin

Schema

```
▼ { [] }
Get list of user beneficiaries

id:      ▼ [] integer (int32)
Beneficiary Identifier in Database

blockchain_key: ▼ [] string
Unique key to identify blockchain.

protocol:    ▼ [] string
Unique key to identify blockchain.

blockchain_name: ▼ [] string
Blockchain name.

currency:     ▼ [] string
Beneficiary currency code.

uid:          ▼ [] string
Beneficiary owner

name:         ▼ [] string
Human rememberable name which refer beneficiary.

description:  ▼ [] string
Human rememberable description of beneficiary.
```

```

state: ▼ [] string
      Defines either beneficiary active - user can use it to withdraw money or pending - requires beneficiary activation with pin.

withdrawals: ▼ [] integer (int32)
      Withdrawals count for given beneficiary

sent_at: ▼ [] string
      Time when last pin was sent

}

```

patchApiV2PeatioAccountBeneficiariesIdResendPin

Resend beneficiary pin

PATCH

```
/api/v2/peatio/account/beneficiaries/{id}/resend_pin
```

Usage and SDK Samples

[Curl](#) [Java](#) [Android](#) [Obj-C](#) [JavaScript](#) [C#](#) [PHP](#) [Perl](#) [Python](#)

```
curl -X PATCH \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/beneficiaries/{id}/resend_pin"
```

Parameters

Path parameters

Name	Description
id*	Integer (int32) <i>Beneficiary Identifier in Database</i> Required

Responses

Status: 200 - Resend beneficiary pin

postApiV2PeatioAccountBeneficiaries

Create new beneficiary

POST

```
/api/v2/peatio/account/beneficiaries
```

Usage and SDK Samples

```
curl -X POST \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/beneficiaries"
```

Parameters

Body parameters

Name	Description
body *	<p>▼ { [] }</p> <p>Required: currency,data,name,otp</p> <p>currency: ▼ [] string <i>Beneficiary currency code.</i></p> <p>Enum: aud , eth , usdt , AUD , ETH , USDT</p> <p>blockchain_key: ▼ [] string <i>Blockchain key of the requested beneficiary</i></p> <p>Enum: aud-monoova , ethereum , fiat</p> <p>name: ▼ [] string <i>Human rememberable name which refer beneficiary.</i></p> <p>description: ▼ [] string <i>Human rememberable name which refer beneficiary.</i></p> <p>data: ▼ [] undefined <i>Beneficiary data in JSON format</i></p> <p>otp: ▼ [] integer (int32) <i>OTP to perform action</i></p> <p>}</p>

Responses

Status: 201 - Create new beneficiary

Schema

▼ { [] }

Get list of user beneficiaries

id: ▼ [] integer (int32)
Beneficiary Identifier in Database

blockchain_key: ▼ [] string
Unique key to identify blockchain.

protocol: ▼ [] string
Unique key to identify blockchain.

blockchain_name: ▼ [] string
Blockchain name.

currency: ▼ [] string
Beneficiary currency code.

uid: ▼ [] string
Beneficiary owner

```

name:      ▼ [] string
          Human rememberable name which refer beneficiary.

description: ▼ [] string
          Human rememberable description of beneficiary.

state:      ▼ [] string
          Defines either beneficiary active - user can use it to withdraw money or pending - requires beneficiary activation with pin.

withdrawals: ▼ [] integer (int32)
          Withdrawals count for given beneficiary

sent_at:    ▼ [] string
          Time when last pin was sent

}

```

postApiV2PeatioAccountInternalTransfers

Creates internal transfer.

POST

/api/v2/peatio/account/internal_transfers

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```

curl -X POST \
-H "Content-Type: application/json" \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/internal_transfers"

```

Parameters

Body parameters

Name	Description

body *	<p>▼ { []</p> <p>Required: amount,currency,otp,username_or_uid</p>
	<p>currency: ▼ [] string <i>The currency code.</i></p>
	<p>Enum: aave , ape , aud , audt , axs , bat , bch , bcha , bico , bnb , bnt , bsv , btc , busd , cel , cho , chr , comp , dai , dash , dmf , doge , eos , etc , eth , ftm , gmt , ht , ldo , leo , link , looks , lrc , ltc , nah , nmr , omg , powr , qnt , rmdc , shib , shr , snx , sxp , trst , uma , uni , uos , usdc , usdt , xlm , xmr , xrp , yfi , zec , zil , zrx , AAVE , APE , AUD , AUDT , AXS , BAT , BCH , BCHA , BICO , BNB , BNT , BSV , BTC , BUSD , CEL , CHO , CHR , COMP , DAI , DASH , DMF , DOGE , EOS , ETC , ETH , FTM , GMT , HT , LDO , LEO , LINK , LOOKS , LRC , LTC , NAH , NMR , OMG , POWR , QNT , RMDC , SHIB , SHR , SNX , SXP , TRST , UMA , UNI , UOS , USDC , USDT , XLM , XMR , XRP , YFI , ZEC , ZIL , ZRX</p>
	<p>amount: ▼ [] number (double) <i>The amount to transfer.</i></p>
	<p>otp: ▼ [] integer (int32) <i>OTP to perform action</i></p>
	<p>username_or_uid: ▼ [] string <i>Receiver uid or username.</i></p>
	}

Responses

Status: 201 - Creates internal transfer.

postApiV2PeatioAccountMembersBeneficiariesWhitelisting

Enable/Disable beneficiaries whitelisting for the specific user

POST

```
/api/v2/peatio/account/members/beneficiaries_whitelisting
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X POST \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
"//www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/members/beneficiaries_whitelisting"
```

Parameters

Body parameters

Name	Description

body *	<p>▼ { []</p> <p>Required: otp,state</p> <p>otp: ▼ [] integer (int32) <i>OTP to perform action</i></p> <p>state: ▼ [] undefined <i>The state of user beneficiaries whitelisting.</i></p> <p>}</p>
--------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Responses

Status: 201 - Enable/Disable beneficiaries whitelisting for the specific user

Schema

▼ { []	Returns current member
uid:	▼ [] string Member UID.
email:	▼ [] string Member email.
group:	▼ [] string Member's group.
beneficiaries_whitelisting:	▼ [] string Member's beneficiaries whitelisting.
}	

postApiV2PeatioAccountWithdraws

Creates new withdrawal to active beneficiary.

POST

/api/v2/peatio/account/withdraws

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X POST \
-H "Content-Type: application/json" \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/withdraws"
```

Parameters

Body parameters

Name	Description

body *

```

▼ { }

Required: amount,blockchain_key,currency,otp
otp: ▼ [] integer (int32)
      OTP to perform action

beneficiary_id: ▼ [] integer (int32)
      ID of Active Beneficiary belonging to user.

rid: ▼ [] string
      Wallet address on the Blockchain.

currency: ▼ [] string
      The currency code.

Enum: aave, ape, aud, audt, axs, bat, bch, bcha, bico, bnb, bnt, bsv, btc,
busd, cel, cho, chr, comp, dai, dash, dmf, doge, eos, etc, eth, ftm, gmt, ht,
ldo, leo, link, looks, lrc, ltc, nah, nmr, omg, powr, qnt, rmdc, shib, shr,
snx, sxp, trst, uma, uni, uos, usdc, usdt, xlm, xmr, xrp, yfi, zec, zil, zrx,
AAVE, APE, AUD, AUDT, AXS, BAT, BCH, BCHA, BICO, BNB, BNT, BSV, BTC, BUSD,
CEL, CHO, CHR, COMP, DAI, DASH, DMF, DOGE, EOS, ETC, ETH, FTM, GMT, HT, LDO,
LEO, LINK, LOOKS, LRC, LTC, NAH, NMR, OMG, POWR, QNT, RMDC, SHIB, SHR, SNX,
SXP, TRST, UMA, UNI, UOS, USDC, USDT, XLM, XMR, XRP, YFI, ZEC, ZIL, ZRX

amount: ▼ [] number (double)
      The amount to withdraw.

note: ▼ [] string
      Optional user metadata to be applied to the transaction. Used to tag transactions with
memorable comments.

blockchain_key: ▼ [] string
      Blockchain key of the requested withdraw

Enum: aud-monoova, binance-smart-chain, bitcoin, bitcoin-abc, bitcoincash-node,
bitcoinsv, dash, dogecoin, electrum, eth-kovan, eth-testnet, eth-testnet., ethereum,
ethereum-classic, fiat, heco-chain, litecoin, monero, opendax-cloud, ripple, straya,
tron, zcash
}

```

Responses

Status: 201 - Creates new withdrawal to active beneficiary.

postApiV2PeatioMarketOrders

Create a Sell/Buy order.

POST

/api/v2/peatio/market/orders

Usage and SDK Samples

Curl	Java	Android	Obj-C	JavaScript	C#	PHP	Perl	Python
------	------	---------	-------	------------	----	-----	------	--------

```

curl -X POST \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/market/orders"

```

Parameters

Body parameters

Name	Description
body *	<ul style="list-style-type: none"> ▼ { [] } <ul style="list-style-type: none"> Required: market,price,side,volume market: string <ul style="list-style-type: none"> Enum: aaveaud , aaveusdt , apeaud , apeusdt , axsaud , axsusdt , bataud , batUSD , bicoaud , bicousdt , bnbaud , bnbusdt , bntaud , bntusdt , btcaud , btcusdt , celaud , celusdt , choaud , chousdt , chraud , chrusdt , compaud , compusdt , daiaud , daiusdt , dashaud , dashusdt , dmfaud , dmfusdt , eosaud , eosusdt , etcaud , etcusdt , ethaud , ethusdt , ftmaud , ftmusdt , htaud , htsdt , ldoaud , ldousdt , leoaud , leousdt , linkaud , linkusdt , looksaud , looksusdt , lrcaud , lrcusdt , ltcraud , ltcusdt , nmraud , nmrusdt , omgaud , omgusdt , powraud , powrusdt , qntaud , qntusdt , shibaud , shibusdt , shraud , shrusdt , snxaud , snxusdt , sxpaud , sxpusdt , trstaud , trstusdt , umaaud , umausdt , uniaud , uniusdt , uosaud , uosusdt , usdtaud , xrpaud , xrpusdt , yfiaud , yfiusdt , zilaud , zilusdt , zrxaud , zrxusdt , busdusdt , busdaud , usdcdaud , usdcusdt side: string <ul style="list-style-type: none"> Enum: sell , buy volume: number (double) ord_type: string <ul style="list-style-type: none"> Default: limit Enum: market , limit price: number (double) }

Responses

Status: 201 - Create a Sell/Buy order.

Schema

▼ { [] }

Get your orders, result is paginated.

id: ▼ [] integer (int32)
Unique order id.

uuid: ▼ [] string
Unique order UUID.

side: ▼ [] string
Either 'sell' or 'buy'.

ord_type: ▼ [] string
Type of order, either 'limit' or 'market'.

price: ▼ [] number (double)
Price for each unit. e.g. If you want to sell/buy 1 btc at 3000 usd, the price is '3000.0'

avg_price: ▼ [] number (double)
Average execution price, average of price in trades.

state:	▼ [] string
	<i>One of 'wait', 'done', or 'cancel'. An order in 'wait' is an active order, waiting fulfillment; a 'done' order is an order fulfilled; 'cancel' means the order has been canceled.</i>
market:	▼ [] string
	<i>The market in which the order is placed, e.g. 'btcusd'. All available markets can be found at /api/v2/markets.</i>
market_type:	▼ [] string
	<i>Market type.</i>
created_at:	▼ [] string
	<i>Order create time in iso8601 format.</i>
updated_at:	▼ [] string
	<i>Order updated time in iso8601 format.</i>
origin_volume:	▼ [] number (double)
	<i>The amount user want to sell/buy. An order could be partially executed, e.g. an order sell 5 btc can be matched with a buy 3 btc order, left 2 btc to be sold; in this case the order's volume would be '5.0', its remaining_volume would be '2.0', its executed volume is '3.0'.</i>
remaining_volume:	▼ [] number (double)
	<i>The remaining volume, see 'volume'.</i>
executed_volume:	▼ [] number (double)
	<i>The executed volume, see 'volume'.</i>
maker_fee:	▼ [] number (double)
	<i>Fee for maker.</i>
taker_fee:	▼ [] number (double)
	<i>Fee for taker.</i>
trades_count:	▼ [] integer (int32)
	<i>Count of trades.</i>
trades:	▼ [[]]
	<i>Trades with this order.</i>
	▼ { [] }
	<i>Get recent trades on market</i>
id:	▼ [] string
	<i>Trade ID.</i>
price:	▼ [] number (double)
	<i>Trade price.</i>
amount:	▼ [] number (double)
	<i>Trade amount.</i>
total:	▼ [] number (double)
	<i>Trade total (Amount * Price).</i>
fee_currency:	▼ [] number (double)
	<i>Currency user's fees were charged in.</i>
fee:	▼ [] number (double)
	<i>Percentage of fee user was charged for performed trade.</i>
fee_amount:	▼ [] number (double)
	<i>Amount of fee user was charged for performed trade.</i>
market:	▼ [] string
	<i>Trade market id.</i>

```

market_type: ▾ [] string
  Market type.

created_at: ▾ [] string
  Trade create time in iso8601 format.

taker_type: ▾ [] string
  Trade taker order type (sell or buy).

side: ▾ [] string
  Trade side.

order_id: ▾ [] integer (int32)
  Order id.

}

]
}

```

postApiV2PeatioMarketOrdersCancel

Cancel all my orders.

POST

/api/v2/peatio/market/orders/cancel

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```

curl -X POST \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
"//www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/market/orders/cancel"

```

Parameters

Body parameters

Name	Description
------	-------------

body

```
▼ { []  
  market: string  
    Enum: aaveaud , aaveusdt , apeaud , apeusdt , axsaud , axsusdt , bataud , batusd ,  
    bicoaud , bicousdt , bnbaud , bnbusdt , bntaud , bntusdt , btcaud , btcusdt , celaud , celusdt ,  
    choaud , chousdt , chraud , chrusdt , compaud , compusdt , daiaud , daiusdt , dashaud ,  
    dashusdt , dmfaud , dmfusdt , eosaud , eosusdt , etcaud , etcusdt , ethaud , ethusdt , ftmaud ,  
    ftmusdt , htaud , htusdt , ldoaud , ldousdt , leoaud , leousdt , linkaud , linkusdt , looksaud ,  
    looksusdt , lrcaud , lrcusdt , ltcraud , ltcusdt , nmraud , nmrusdt , omgaud , omgusd , powraud ,  
    powrusdt , qntaud , qntusdt , shibaud , shibusdt , shraud , shrusdt , snxaud , snxusdt ,  
    sxpaud , sxpusdt , trstaud , trstusdt , umaaud , umausdt , uniaud , uniusdt , uosaud , uosusdt ,  
    usdtaud , xrpaud , xrpusdt , yfiaud , yfiusdt , zilaud , zilusd , zrxaud , zrxusdt , busdusdt ,  
    busdaud , usdcaud , usdcusdt  
  market_type: string  
    Enum: spot , qe  
  side: ▼ [] string  
    If present, only sell orders (asks) or buy orders (bids) will be cancelled.  
    Enum: sell , buy  
}
```

Responses

Status: 201 - Cancel all my orders.

Schema

▼ { []

Get your orders, result is paginated.

id: ▼ [] integer (int32)
Unique order id.

uuid: ▼ [] string
Unique order UUID.

side: ▼ [] string
Either 'sell' or 'buy'.

ord_type: ▼ [] string
Type of order, either 'limit' or 'market'.

price: ▼ [] number (double)
Price for each unit. e.g. If you want to sell/buy 1 btc at 3000 usd, the price is '3000.0'

avg_price: ▼ [] number (double)
Average execution price, average of price in trades.

state: ▼ [] string
One of 'wait', 'done', or 'cancel'. An order in 'wait' is an active order, waiting fulfillment; a 'done' order is an order fulfilled; 'cancel' means the order has been canceled.

market: ▼ [] string
The market in which the order is placed, e.g. 'btcusd'. All available markets can be found at /api/v2/markets.

market_type: ▼ [] string
Market type.

created_at: ▼ [] string
Order create time in iso8601 format.

updated_at: ▼ [] string

Order updated time in iso8601 format.

origin_volume: ▼ [] number (double)

The amount user want to sell/buy. An order could be partially executed, e.g. an order sell 5 btc can be matched with a buy 3 btc order, left 2 btc to be sold; in this case the order's volume would be '5.0', its remaining_volume would be '2.0', its executed volume is '3.0'.

remaining_volume: ▼ [] number (double)

The remaining volume, see 'volume'.

executed_volume: ▼ [] number (double)

The executed volume, see 'volume'.

maker_fee: ▼ [] number (double)

Fee for maker.

taker_fee: ▼ [] number (double)

Fee for taker.

trades_count: ▼ [] integer (int32)

Count of trades.

trades: ▼ []

Trades wiht this order.

 ▼ { [] }

Get recent trades on market

 id: ▼ [] string

Trade ID.

 price: ▼ [] number (double)

Trade price.

 amount: ▼ [] number (double)

Trade amount.

 total: ▼ [] number (double)

*Trade total (Amount * Price).*

 fee_currency: ▼ [] number (double)

Currency user's fees were charged in.

 fee: ▼ [] number (double)

Percentage of fee user was charged for performed trade.

 fee_amount: ▼ [] number (double)

Amount of fee user was charged for performed trade.

 market: ▼ [] string

Trade market id.

 market_type: ▼ [] string

Market type.

 created_at: ▼ [] string

Trade create time in iso8601 format.

 taker_type: ▼ [] string

Trade taker order type (sell or buy).

 side: ▼ [] string

Trade side.

 order_id: ▼ [] integer (int32)

Order id.

```
        }
    ]
}
```

postApiV2PeatioMarketOrdersIdCancel

Cancel an order.

POST

```
/api/v2/peatio/market/orders/{id}/cancel
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X POST \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/market/orders/{id}/cancel"
```

Parameters

Path parameters

Name	Description
id*	String Required

Responses

Status: 201 - Cancel an order.

postApiV2PeatioPublicWebhooksAdapterEvent

Webhook controller

POST

```
/api/v2/peatio/public/webhooks/{adapter}/{event}
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X POST \
"/www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/public/webhooks/{adapter}/{event}"
```

Parameters

Path parameters

Name	Description
adapter*	String <i>Name of adapter for process webhook</i> Required
event*	String <i>Name of event can be deposit or withdraw</i> Required

Responses

Status: 201 - Webhook controller

putApiV2PeatioAccountBeneficiariesId

Update beneficiary

PUT

```
/api/v2/peatio/account/beneficiaries/{id}
```

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X PUT \
-H "Content-Type: application/json" \
"//www.coinharbour.com.au/api/v2/peatio/api/v2/peatio/account/beneficiaries/{id}"
```

Parameters

Path parameters

Name	Description
id*	Integer (int32) <i>Beneficiary Identifier in Database</i> Required

Body parameters

Name	Description
body *	<pre>▼ { [] }</pre> <p>Required: otp,state</p> <p>state: <pre>▼ [] string</pre> <i>Beneficiary state</i></p> <p>Enum: <code>active</code>, <code>disabled</code></p> <p>otp: <pre>▼ [] integer (int32)</pre> <i>OTP to perform action</i></p> <pre>}</pre>

Responses

Status: 200 - Update beneficiary

Suggestions, contact, support and error reporting;

Information URL: <https://www.coinharbour.com.au> (<https://www.coinharbour.com.au>)

Contact Info: support@coinharbour.com.au (support@coinharbour.com.au)

All rights reserved

<https://github.com/openware/peatio/blob/master/LICENSE.md>