

Introduction on API

Coinoah provides users with a full set of simple yet powerful development tools, designed to enable users to efficiently integrate Coinoah trading functions into their applications.

The Coinoah interface is the basis for accessing services. Upon creating an account on the Coinoah website, the developer could customize APIs by setting privileges on different levels and conducting automatic transactions or withdrawals on the APIs.

The APIs allow developers to:

- Obtain Coinoah currency to currency market price
- Obtain Coinoah market depth
- Obtain Coinoah candlestick data
- Obtain Coinoah historical order information
- Obtain Coinoah user information
- Order transaction
- Obtain pending order information based on order number
- Obtain pending order information based on trading pair
- Request withdrawal

Read this interface document as a reference after interface access is obtained.

If there are any issues, feel free to contact our API technical support at support@coinoah.com

Interface Invocation Pattern Specifications

COINOAH provides REST interface invocation patterns.

REST API

Representational State Transfer (short for "REST"), is now the most popular internet software infrastructure. With its clear infrastructure, compliance to standards, easy comprehensibility, and convenience, REST is applied on more and more websites.

Technical advantages of REST are as follows:

- In RESTful infrastructure, each URL represents one type of resource;
- Certain presentation layer of the resource is transferred between the client-side and the server;
- By calling four HTTP orders, the client-side is able to operate on the resource on the server, therefore realize the "representational state transfer" .

API Overview

REST API Document PUBLIC

1. Obtain Coinoah currency to currency market price

URL

<https://coinoah.com/exchangeApi/api/ticker>

#REQUEST

GET

<https://coinoah.com/exchangeApi/api/ticker>

#Response

```
{
  "code": "100200",
  "data": {
    "date": 1540366089711,
    "ticker": [
      {
        "symbol": "BTC_TRX",
        "high": "0.00000365400000000000",
        "vol": "340972524.11940000000000000000",
        "last": 0.000003624,
        "low": "0.00000356400000000000",
        "change": 0,
        "buy": "0.00000305500000000000",
        "sell": "0.00000298500000000000"
      },
      Object{...}
    ]
  },
  "msg": "成功"
}
```

2. Obtain Coinoah market depth

URL

https://www.coinoah.com/trade/trade?symbol=ETH_EKT

Request parameters

Parameters name	Type	Required	Description
symbol	String	Yes	Trading pair (add in the URL above)

COINOAH TEAM

#Request

GET

https://www.coinoah.com/trade/trade?symbol=ETH_EKT

#Response

```
{
  "sell": [
    [
      0.00002736,
      1071.36299794
    ],
    [
      0.00002738,
      0.19978522
    ],
    [
      0.00002742,
      75.9087355
    ]
  ],
  "buy": [
    [
      0.00002618,
      16.33033728
    ],
    [
      0.00002611,
      605.26813127
    ],
    [
      0.0000261,
      499.81630013
    ]
  ]
}
```



3. Obtain Coinoah candlestick data

URL

https://www.coinoah.com/dataload/kline-query/pages?startTime=1522206827&endTime=1523502887&pageSize=300&symbol=BTC_EOS&kline=15m

Request parameters

Parameters name	Type	Required	Description
startTime	long	Yes	Start time (add in the URL

COINOAH TEAM

			above)
endTime	long	Yes	End time (add in the URL above)
pageSize	int	Yes	300 data point (add in the URL above)
symbol	String	Yes	Trading pair (add in the URL above)
kline	String	Yes	`1m` `5m` `15m` `30m` `1h` `1h` (add in the URL above)

#Request

GET

https://www.coinoah.com/dataload/kline-query/pages?startTime=1522206827&endTime=1523502887&pageSize=300&symbol=BTC_EOS&kline=15m

#Response

```
{
  "code": "100200",
  "data": {
    "pages": {
      "list": [
        {
          "amount": 1228.7905,
          "close": 0.0012868,
          "klineType": "MIN15",
          "lowPrice": 0.0012703,
          "maxPrice": 0.0012931,
          "open": 0.0012917,
          "symbol": "BTC_EOS",
          "time": 1523502000
        },
        {
          "amount": 1957.1155,
          "close": 0.0012917,
          "klineType": "MIN15",
          "lowPrice": 0.0012498,
          "maxPrice": 0.0012925,
          "open": 0.0012519,
          "symbol": "BTC_EOS",
          "time": 1523501100
        }
      ],
      "pageNum": 1,
      "pageSize": 2,
    }
  }
}
```

```

        "pages": 1,
        "size": 2,
        "startIndex": 0,
        "total": 2
    }
},
"msg": "成功"
}

```

4. Historical Order information

URL https://www.coinoah.com/trade/info?symbol=BTC_EOS

Request parameters

Parameters name	Type	Required	Description
symbol	String	Yes	Trading pair (add in the URL above)

#Request

GET

https://www.coinoah.com/trade/info?symbol=BTC_EOS

#Response

```

{
  "24high": "0.00089970000000000000",
  "24low": "0.00087968000000000000",
  "24Total": "386060.17280000000000000000",
  "order": [{
    "time": "1539087652",
    "type": "SELL",
    "num": 263.4154,
    "price": 0.00088498
  }, {
    "time": "1539087611",
    "type": "SELL",
    "num": 355.9577,
    "price": 0.00088468
  }, {
    "time": "1539087585",
    "type": "BUY",
    "num": 24.8504,
    "price": 0.00088408
  }
}

```

```

    }, {
      "time": "1539087548",
      "type": "SELL",
      "num": 314.8985,
      "price": 0.00088478
    }
    .....
  }

```

REST API Document PRIVATE

Request Interactions

REST URL

<https://www.coinoah.com/exchangeApi>

Request Interactions

Parameters name	Type	Required	Description
businessNo	String	Yes	Api account
nonceStr	String	Yes	32digit random number
timestamp	long	Yes	timestamp (second)
data	String	Yes	JSON
sign	String	Yes	Please see "sign request parameters"

- Submit

Convert the encapsulated request parameters to `JSON format` and submit them to the server via POST.

- Response

The server first performs parameter security verification on the user request data, and after verification, responds the data to the client in the `JSON format` according to the business logic.

- Response

Parameters name	Type	Required	Description
code	String	Yes	Response code
msg	String	Yes	Response message
data	String	Yes	JSON format

- Sign request parameters

COINOAH TEAM

- Add `data` to the map as a key-value pair
- Add `timestamp` to the map
- Add `apiSecret` to the map
- Sort the map in ascending order; and stitch it into a URL string
- MD5 encryption of the spliced string; conversion to HEX uppercase string

1. Obtain Coinoah user information

URL

<https://www.coinoah.com/exchangeApi/api/asset>

#REQUEST

POST

```
{
  "businessNo": "2",
  "nonceStr": "19371335969612288099907154904052",
  "timestamp": "1539093137",
  "data": {},
  "sign": "C3EC666666B15528CE9E56AUUUUCE8A5"
}
```

Response

Parameters name	Type	Required	Description
assetCode	String	Yes	Asset Code
amountAvailable	String	Yes	Amount Available
amountLock	String	Yes	Amount Lock
amountLoan	String	Yes	Loan Amount

#RESPONSE

```
{
  "code": "100200",
  "data": [{
    "accountKind": "MASTER",
    "accountNo": "999629309752299521",
    "amountAvailable": 79,
    "amountLoan": 0E-20,
    "amountLock": 18,
    "assetCode": "AE",
    "brokerId": 10003,
    "id": 311,
    "uid": 10,
    "updateDate": 1539088288000,
  }
]
```

```

        "version": 2508686
    }, {
        "accountKind": "MASTER",
        "accountNo": "1004212491164262403",
        "amountAvailable": 17,
        "amountLoan": 0E-20,
        "amountLock": 11,
        "assetCode": "AVH",
        "brokerId": 10003,
        "id": 7444,
        "uid": 10,
        "updateDate": 1538286825000,
        "version": 4027119
    }],
    "msg": "成功"
}

```

2. Order transaction

URL

<https://www.coinoah.com/exchangeApi/api/matchOrder>

Request parameters

Parameters name	Type	Required	Description
outOrderNo	String	Yes	Order number (unique)
symbol	String	Yes	Trading pair
tradeCoinFlag	String	Yes	`FIXED` Limit price matching
tradeCoinType	String	Yes	`BUY` `SELL`
price	BigDecimal	Yes	Price
amount	BigDecimal	Yes	Amount

#REQUEST

POST

```

{
    "businessNo": "2",
    "nonceStr": "19371335969612288099907154904052",
    "timestamp": "1539093137",
    "data": {
        "outOrderNo": "36fffffe-b767-4094-8588-ae39f21b68e1",
        "symbol": "ETH_EKT",
        "tradeCoinFlag": "FIXED",
        "tradeCoinType": "BUY",
        "price": 0.00006713,

```



```

        "amount": 3.3
    },
    "sign": "C3EC666666B15528CE9E56AUUUUCE8A5"
}

```

3. Obtain pending order information based on order number

URL

<https://www.coinoah.com/exchangeApi/api/orderquery>

Request parameters

Parameters name	Type	Required	Description
outTradeNo	String	Yes	Order number

#REQUEST

POST

```

{
    "businessNo": "06",
    "nonceStr": "35880572999956722922710963171763",
    "timestamp": "1539142758",
    "data": {
        "outTradeNo": "455bc3ae-82d9-412b-804b-dc7895656514"
    },
    "sign": "90551D38FERTYU46378606C8CA736F33"
}

```

Response

Parameters name	Description
createTime	Creat time
tradedNumber	Traded amount
numberOver	Amount left
orderNo	Order number
payTransactionNo	Transaction number
tradeCoinType	`BUY` `SELL`
tradeCoinFlag	`FIXED`limit price matching
symbol	Trading pair
price	price
number	number
matchedMoney	Amount matched
moneyOver	Amount left ocer
tradeCoinStatus	Transaction order status

COINOAH TEAM

#RESPONSE

```
{
  "code": "100200",
  "data": {
    "createTime": 1539142752000,
    "matchedMoney": 0E-20,
    "money": null,
    "moneyOver": 0E-20,
    "number": 20.00000000000000000000,
    "numberOver": 20.00000000000000000000,
    "orderNo": "455bc3ae-82d9-412b-804b-dc7895656514",
    "payTransactionNo": "104956568446216",
    "price": 0.0005800000000000000000,
    "symbol": "ETH_CMT",
    "tradeCoinFlag": "FIXED",
    "tradeCoinStatus": "PROCESSING",
    "tradeCoinType": "SELL",
    "tradedNumber": 0E-20
  },
  "msg": "成功"
}
```

4. Obtain pending order information based on trading pair (200 data points)

URL

<https://www.coinoah.com/exchangeApi/api/matchOrder/process>

Request parameters

Parameters name	Type	Required	Description
symbol	String	Yes	Trading pair

#REQUEST

POST

```
{
  "businessNo": "1000300001",
  "nonceStr": "57048294028053518195749838529385",
  "timestamp": "1539143077",
  "data": {
    "symbol": "ETH_CMT"
  },
  "sign": "25DAD7891CD9EA094EBBAFB290D768D5"
}
```

Response

Parameters name	Description
createTime	Creat time
tradedNumber	Traded amount
numberOver	Amount left
orderNo	Order number
payTransactionNo	Transaction number
tradeCoinType	`BUY` `SELL`
tradeCoinFlag	`FIXED`limit price matching
symbol	Trading pair
price	price
number	number
matchedMoney	Amount matched
moneyOver	Amount left ocer
tradeCoinStatus	Transaction order status

#RESPONSE

```
{
  "code": "100200",
  "data": [{
    "createTime": 1539140733000,
    "matchedMoney": 0E-20,
    "money": null,
    "moneyOver": 0E-20,
    "number": 392.744900000000000000000000,
    "numberOver": 392.7449000000000000000000,
    "orderNo": "28111147-23d0-ffff-abf8-ff2867f16871",
    "payTransactionNo": "936",
    "price": 0.0005801200000000000000,
    "symbol": "ETH_CMT",
    "tradeCoinFlag": "FIXED",
    "tradeCoinStatus": "PROCESSING",
    "tradeCoinType": "SELL",
    "tradedNumber": 0E-20
  }],
  "msg": "成功"
}
```

5. Request withdrawal

URL

<https://www.coinoah.com/exchangeApi/api/cancel>

Request parameters

COINOAH TEAM

Parameters name	Type	Required	Description
outTradeNo	String	Yes	Order number

#REQUEST

POST

```
{
  "businessNo": "1000300001",
  "nonceStr": "53092405069999013009975712249140",
  "timestamp": "1539143361",
  "data": {
    "outTradeNo": "a6ab84d9-59c8-4fd6-820d-booppa3ffd0b"
  },
  "sign": "52EBBD8418COPOPD00223094B720657A"
}
```

Response

Parameters name	Description
100200	Success
100104	Invalid request
100103	Service not available
100100	Incoming field validation error
102100	Invalid trading pair
102101	Repeated order number
101102	Order not available

COINOAH