Restful API Information

General Information

- All endpoints should add a request header named `X-EX-APIKEY` with `apiKey` as a value **except test connectivity endpoint**.
- All endpoints require an additional parameter `signature` to be sent **except test connectivity endpoint**.
- Endpoints use HMAC SHA256 signatures. The HMAC SHA256 signature is a keyed HMAC SHA256 operation. All params sorted by alphabet order except param `signature` concatenated with `&` as `totalParams`. Then use your `secretKey` as the key and `totalParams` as the value for the HMAC operation.
- The `signature` is **not case sensitive**.

For example, there is a post request with content-type by application/x-www-form-urlencoded, the request params are symbol=eth_btc, limit=100

1. All params should sorted by alphabet order and concatenated with & , then we get totalParams

```
limit=100&symbol=eth_btc
```

2. Use secretKey as the key and totalParams as the value for the HMAC operation.

```
[linux]$ echo -n "limit=100&symbol=eth_btc" | openssl dgst -sha256 -hmac
"4ece26928af3c7a30c91a40b0cdb450c"
(stdin)= 32f9b8ef255fc02fc0041a5a497e62cb8b327fdb716c8dd1e629b438a1785bef
signature=32f9b8ef255fc02fc0041a5a497e62cb8b327fdb716c8dd1e629b438a1785bef
```

3. Send request

```
curl -H "X-EX-APIKEY: 82b037dddd35453fb963e9dbd2d678b9" -X POST
"limit=100&symbol=eth_btc&signature=32f9b8ef255fc02fc0041a5a497e62cb8b327fdb716c8dd1e6
29b438a1785bef" https://apiServer/uri
```

1. Test Connectivity

- **Request**:

```
GET /api/isLive
```

- **Response:**

```
{
   "data": null,
   "msg": "ok",
   "code": 0
}
```

2. Get Account Balance

- **Request**:

```
GET /api/account/balance
```

If assetCode is not set, return balance of all assets otherwise return the balance of the specified assetCode

- **Request Params**:

Name	Type	Mandatory	Description
assetCode	string	no	such as eth, btc, ltc, assetCode must be lowercase

- **Response:**

```
    {
        "data": [
           {
               "asset": "eth",          // asset code
               "usable": 1000.0000,     // available amount
               "locked": 2.0000        // locked amount
           },
           {
               "asset": "btc",
               "usable": 200.0000,
               "locked": 0.05
           }
        ],
        "msg": "ok",
        "code": 0
    }
```

3. Ticker Details

- **Request**:

```
GET /api/ticker/get
```

- **Request Params**:

Name	Type	Mandatory	Description
symbol	string	yes	such as eth_btc, ltc_btc, symbol must be lowercase

- **Response:**

```
{
            "data": {
      "last":"0.00000048",
      "lowestAsk":"0.00000048",
      "highestBid":"0.00000047",
      "percentChange":"0.09090909",
      "baseVolume":"73.54205337",
      "quoteVolume":"155804874.92619288",
      "high24hr":"0.00000050",
      "low24hr":"0.00000044"
    },
            "msg": "ok",
            "code": 0
    }
```

4. Ticker List

- **Request**:

```
GET /api/ticker/list
```

- **Request Params**:

```
Empty
```

- **Response:**

```
    {
            "data":{
               "eth_btc":{
                  "last":"0.00000048",
                  "lowestAsk":"0.00000048",
                  "highestBid":"0.00000047",
                  "percentChange":"0.09090909",
                  "baseVolume":"73.54205337",
                  "quoteVolume":"155804874.92619288",
                  "high24hr":"0.00000050",
                  "low24hr":"0.00000044"
               },
               "caf_btc":{
                  "last":"0.00001944",
                  "lowestAsk":"0.00001965",
                  "highestBid":"0.00001944",
                  "percentChange":"0.01302761",
                  "baseVolume":"0.21144312",
                  "quoteVolume":"11015.01644333",
                  "high24hr":"0.00001972",
                  "low24hr":"0.00001886"
               }
            },
            "msg":"ok",
            "code":0
    }
```

5. New Order

- **Request**:

```
POST /api/order/create
Content-Type: application/x-www-form-urlencoded
```

- **Request Params**:

Name	Type	Mandatory	Description
symbol	string	yes	such as eth_btc, ltc_btc
side	string	yes	buy or sell
type	string	yes	limit
price	decimal	yes	-
quantity	decimal	yes	-

- **Response:**

```
    {
                "data": {
                   "symbol": "eth_btc",
                   "orderUuid": "0bdb5a8b-d6e5-4fbf-b133-3bcb34f782c8", // unique identifier
of the order
                   "orderTime": 1527237096525  // utc timestamp when the order created
                },
                "msg": "ok",
                "code": 0
         }
```

6. Cancel Order

- **Request**:

```
POST /api/order/cancel
Content-Type: application/x-www-form-urlencoded
```

- **Request Params**:

Name	Type	Mandatory	Description
symbol	string	yes	such as eth_btc
orderUuid	string	yes	unique identifier of the order

- **Response:**

```
    {
                "data": {
                   "symbol": "eth_btc",
                   "orderUuid": "0bdb5a8b-d6e5-4fbf-b133-3bcb34f782c8" // unique identifier
of the order
                },
                "msg": "ok",
                "code": 0
         }
```

7. Get Order Information

- **Request**:

```
GET /api/order/info
```

- **Request Params**:

Name	Type	Mandatory	Description
symbol	string	yes	such as eth_btc
orderUuid	string	yes	unique identifier of the order

- **Response:**

```
    {
                "data": {
                   "symbol": "eth_btc",
                   "orderUuid": "0bdb5a8b-d6e5-4fbf-b133-3bcb34f782c8", // unique identifier
of the order
                   "price": 0.7000,
                   "qty": 1.00,
                   "dealQty": 1.00,        // dealed quantity
                   "avgPrice": 1.00,       // average deal price
                   "status": "full_deal",  // order status
[new,fullDeal,partialDeal,canceling,canceled]
                   "type": "limit",
                   "side": "buy",
                   "orderTime": 1527237096525 // utc timestamp when the order created
                },
                "msg": "ok",
                "code": 0
         }
```

8. Get My All Orders

- **Request**:

  ```
  GET /api/order/myAll
  ```

  If startTime or endTime is specified, return orders with the orderTime between startTime and endTime ordered by orderTime asc.
  If startTime and endTime is not specified, return latest orders ordered by orderTime desc.

- **Request Params**:

Name	Type	Mandatory	Description
symbol	string	yes	such as eth_btc
startTime	long	no	utc timestamp
endTime	long	no	utc timestamp
limit	int	no	count of records, max: 300, default: 50

- **Response:**

```
    {
                "data":
                [
                   {
                       "symbol": "eth_btc",
                       "orderUuid": "0bdb5a8b-d6e5-4fbf-b133-3bcb34f782c8", // unique
identifier of the order
                       "price": 0.7000,
                       "qty": 1.00,
                       "dealQty": 1.00,        // dealed quantity
                       "avgPrice": 1.00,       // average deal price
                       "status": "full_deal",  // order status [new, full_deal,
partial_deal, canceling, canceled]
                       "type": "limit",
                       "side": "buy",
                       "orderTime": 1527237096525 // utc timestamp when the order created
                   },
                   {
                       ...
                   }
                ],
                "msg": "ok",
                "code": 0
         }
```

9. Get My Open Orders

- **Request**:

```
GET /api/order/myOpens
```

If startTime or endTime is specified, return orders with the orderTime between startTime and endTime ordered by orderTime asc.
If startTime and endTime is not specified, return latest orders ordered by orderTime desc.

Notice: The status of open orders contains [new, partial_deal, canceling].

- **Request Params**:

Name	Type	Mandatory	Description
symbol	string	yes	such as eth_btc
startTime	long	no	utc timestamp
endTime	long	no	utc timestamp
limit	int	no	count of records, max: 300, default: 50

- **Response:**

```
    {
                "data":
                [
                   {
                       "symbol": "eth_btc",
                       "orderUuid": "0bdb5a8b-d6e5-4fbf-b133-3bcb34f782c8", // unique
identifier of the order
                       "price": 0.7000,
                       "qty": 1.00,
                       "dealQty": 1.00,  // dealed quantity
                       "avgPrice": 1.00,       // average deal price
                       "status": "new",  // order status [new, partial_deal, canceling]
                       "type": "limit",
                       "side": "buy",
                       "orderTime": 1527237096525 // utc timestamp when the order created
                   },
                   {
                       ...
                   }
                ],
                "msg": "ok",
                "code": 0
         }
```

10. Get Order Book

- **Request**:

  ```
  GET /api/depth
  ```

  Notice: All exchanges share the same depth by the specified symbol in Expread.

- **Request Params**:

Name	Type	Mandatory	Description
symbol	string	yes	such as eth_btc
limit	int	no	max: 200, default: 200

- **Response:**

```
{
            "data": {
               "asks": [
                   ["0.07", "0.01"]   // [price, quantity]
               ],
               "bids": [
                   ["0.06", "0.01"]   // [price, quantity]
               ]
            },
            "msg": "ok",
            "code": 0
     }
```

11. Get Recent Trades

- **Request**:

  ```
  GET /api/trades/recent
  ```

  Trades are ordered by time desc.

Notice: A trade may generated by the ask from exchange A and the bid from exchange B In Expread.

- **Request Params**:

Name	Type	Mandatory	Description
symbol	string	yes	such as eth_btc
limit	int	no	max: 200, default: 200

- **Response:**

```
{
            "data": [
               {
                   "price": "0.07",        // deal price
                   "qty": "0.01",          // deal quantity
                   "time": 1527237096525   // Utc timestamp of deal time
               },
               {
                   ...
               }
            ],
            "msg": "ok",
            "code": 0
     }
```

12. Get Klines

- **Request**:

```
GET /api/klines
```

If startTime or endTime is specified, return klines with open time between startTime and endTime ordered by open time asc.
If startTime and endTime is not specified, return latest klines ordered by open time desc.

Notice: Klines are aggregated with transactions by the specified symbol from all exchanges in Expread.

- **Request Params**:

Name	Type	Mandatory	Description
symbol	string	yes	such as eth_btc
interval	string	yes	1m, 5m, 15m, 30m, 1h, 2h, 4h, 6h, 8h, 12h, 1d, 1w, 1M m: minute, h: hour, d: day, w: week, M: month
limit	int	no	max: 300, default: 100
startTime	long	no	utc timestamp
endTime	long	no	utc timestamp

- **Response:**

```
{
            "data": [
               [
                  1526971020000,  // open time
                  "700",          // open price
                  "720",          // high price
                  "680",          // low  price
                  "710",          // close price
                  "70000"         // volume
               ]
            ],
            "msg": "ok",
            "code": 0
     }
```

13. Get Crypto Recharge address

- **Request**:

```
GET /api/recharge/cryptoWalletInfo
```

- **Request Params**:

Name	Type	Mandatory	Description
assetCode	string	yes	such as eth

- **Response:**

```
   {
     "code": 0,
     "msg": "ok",
     "data": {
       "assetCode": "eth",
       "digitAssetAddress": "0x19b24750ac76bb4cc1a2e7f8379c54aff7918ca5" //recharge
address
     }
   }
```

14. Get Crypto Recharge Records

- **Request**:

```
GET /api/recharge/cryptoRechargeRecords
```

If startTime or endTime is specified, return records with create time between startTime and endTime ordered by create time desc.

- **Request Params**:

Name	Type	Mandatory	Description
assetCode	string	no	such as eth
limit	int	no	max: 300, default: 50
startTime	long	no	utc timestamp
endTime	long	no	utc timestamp

- **Response:**

```
   {
     "code": 0,
     "msg": "ok",
     "data": [
       {
         "txId": "0xbcc271cbc172aeea7a5c90664b93b3322211c25aadf1a8674544750810de6bf1",
//txId
         "blockTime": 1543314156,
         "valueOut": 0.01, // recharge amount
         "outAddress": "0x1606035dc4023346eef90dbc1b279096dc4e0f96", //recharge address
         "assetCode": "eth", // recharge crypto code
         "blockId": 6781629,
         "txIndex": 32,
         "confirmations": 1,
         "status": 2,    // status :1 unconfirmed,2confirmed
         "createTime": 1543318090000,
         "updateTime": 1545968385000
```

```
      },
      {
        ...
      }
    ]
  }
```

15. Get User Transactions

- **Request**:

```
GET /api/transaction/userTransactionList
```

If startTime or endTime is specified, return records with deal time between startTime and endTime ordered by create time desc.

- **Request Params**:

Name	Type	Mandatory	Description
symbol	string	yes	such as eth_btc
limit	int	no	max: 300, default: 50
side	String	no	buy or sell
orderUuid	String	no	order number
startTime	long	no	utc timestamp
endTime	long	no	utc timestamp

- **Response:**

```
  {
    "code": 0,
    "msg": "ok",
    "data": [
      {
        "txUuid": "a72e0533c7ee45868b5f04a61fb9b756",       //transaction number
        "orderUuid": "ff5ede34-a68d-4d95-9231-81eba5fe60f5", //order number
        "side": "sell",
        "tradeFee": 0.00002498,                             //transaction fee
        "dealTime": 1553466691389,                          // deal time
        "dealPrice": 0.03416,
        "dealQty": 0.731,
        "dealVolumePrice": 0.02497096                      //total
      },
      {
        ...
      }
    ]
  }
```

16. Get Crypto Withdrawal Records

- **Request**:

```
GET /api/withdrawal/cryptoWithdrawalRecords
```

If startTime or endTime is specified, return records with create time between startTime and endTime ordered by create time desc.

- **Request Params**:

Name	Type	Mandatory	Description
assetCode	string	no	such as eth
limit	int	no	max: 300, default: 50
recordId	long	no	record number
txId	String	no	txid
status	int	no	withdraw status :5 email pending,10 email failed,15pending ,20 pending2,25 denied， 30 withdrawing， 35 denied， 45 cancelled,50 cancelled2
startTime	long	no	utc timestamp
endTime	long	no	utc timestamp

- **Response:**

```
   {
     "code": 0,
     "msg": "ok",
     "data": [
       {
         "id": 31,
         "assetCode": "btc",
         "amount": 0.1,
         "fee": 0.002,
         "walletAddress":
"aaa0b9d8fec7d3cbada063dd395a337aabae584c78f218be69a8dbf088d78e4c",
         "status": 30,
         "failureReason": null,
         "createTime": 1554886257000,
         "txId": null,
         "withTime": null
       },
       {
         ...
       }
```

```
    ]
  }
```

17. Apply Crypto Withdrawal

- **Request**:

```
GET /api/withdrawal/cryptoWithdrawalRecords
```

- **Request Params**:

Name	Type	Mandatory	Description
assetCode	string	no	such as eth
walletAddress	string	no	withdrawing address
applyAmt	BigDecimal	no	withdrawal amount

- **Response:**

```
  {
    "code": 0,
    "msg": "ok",
    "data": 33   // recordId
  }
```

Websocket API Information

General Information

- The endpoint of the websocket is ws://server:port/ws.
- If client send ping frame, server will return pong frame.
- Client can send a message with "ping" instead of sending a ping frame if it can not send a ping frame.

1. Publish Order Book

- **Message:**
    - **subscribe**
      { "event": "subscribe", "channel": "sub_depth", "limit": 200, "symbol": "eth_btc" }
      symbol: such as eth_btc
    - **unsubscribe**
      { "event": "unsubscribe", "channel": "sub_depth", "symbol": "eth_btc" }

  Notice: All exchanges share the same depth by the specified symbol in Expread.

- **Response:**

```
{
            "symbol": "eth_btc",
            "channel": "sub_depth",
            "data": {
               "asks": [
                   ["0.07", "0.01"]   // [price, quantity]
               ],
               "bids": [
                   ["0.06", "0.01"]   // [price, quantity]
               ]
            }
     }
```

2. Publish Ticker

- **Message:**
  - **subscribe**
    { "event": "subscribe", "channel": "sub_ticker", "symbol": "eth_btc" }
    symbol: such as eth_btc
  - **unsubscribe**
    { "event": "unsubscribe", "channel": "sub_ticker", "symbol": "eth_btc" }
- **Response:**

```
{
            "symbol": "eth_btc",
            "channel": "sub_ticker",
            "data": {
               "askPrice": "0.07",
               "askQty": "0.01",
               "bidPrice": "0.06",
               "bidQty": "1.00"
            }
     }
```

3. Publish Trades

- **Message:**
  - **subscribe**
    { "event": "subscribe", "channel": "sub_trades", "symbol":"eth_btc" }
    symbol: such as eth_btc
  - **unsubscribe**
    { "event": "unsubscribe", "channel": "sub_trades", "symbol": "eth_btc" }

  Notice: A trade may generated by the ask from exchange A and the bid from exchange B In Expread
- **Response:**

```
{
            "symbol": "eth_btc",
            "channel": "sub_trades",
            "data": [
               [
                   0.07,                // price
                   1,                   // qty
                   1526970840000        // utc timestamp of deal time
               ]
            ]
     }
```

4. Publish User Trades

- **Message:**
    - **subscribe**
      { "event": "subscribe", "channel": "sub_user_trades", "symbol":"eth_btc", "apiKey": "82b037dddd35453fb963e9dbd2d678b9" }
      symbol: such as eth_btc
    - **unsubscribe**
      { "event": "unsubscribe", "channel": "sub_user_trades, "symbol": "eth_btc" }

  Notice: One socket connection can only subscribe one user trades. If you want to subscribe different user trades, please use different socket connections.

- **Response:**

```
{
            "symbol": "eth_btc",
            "channel": "sub_user_trades",
            "data": [
               [
                   0.07,                // price
                   1,                   // qty
                   0.07,                // volume
                   1526970840000        // utc timestamp of deal time
               ]
            ]
     }
```