

1. Integration Guidance

Integrate URLs

REST API

Market charts: `wss://kapi1.btloginc.com:9082`

Heartbeat API calls, sent once every 20 seconds, are required to maintain the connection for market charts.

Request Format

All API requests are sent in JSON format.

Return Format

All API returns are in JSON format. At the top level of the JSON format, there are several fields indicating the status and attributes of the request: "code" and "msg". The other fields are the content of the return. When the code equals 0 it indicates success; code not equal to 0 means failure. Msg is the specific error message, and data is the returned data.

2. Market Charts API

Login

Request JSON parameters:

```
{  
    "name": "86000402",                                // The account number field returned after  
    successful login to the trading server  
    "clienttype": 1,                                     // Type of clients  
    "key": "55117c2a-84cb-44b1-b179-24273a304c48"   // Key registered on the platform  
}
```

3 network packets returned (login result returned, dictionary, market open time data)

1. Login result returned

```
{  
    "code": 0,                                         //Successful or not  
    "data": {  
        "Result": 4,  
        "Time": "2023-01-15T10:00:00Z"  
    },  
    "msg": ""  
}
```

```

    "DataSourceState": 0,                                If the data source of the market server is
stable
    "LoginResult": 0,                                 Result of logging into the market server. 0: successful;
Not 0: failure
    "Reserve": 0,
    "token": "",
    "Seq": 1234,
    "Year": 2019,                                     //Time of the market server
    "Month": 10,                                       //Time of the market server
    "Day": 24,                                         //Time of the market server
    "Time": 1571910061                                //Time of the market server
},
    "msg": ""
}

```

2. Dictionary

```

{
  "code": 0,
  "data": {
    "Num": 151,                                         //Total number of products
    "DictInfo": [
      {"SecID": 3158070,                                //Product code
       "ShortName": "BCH/USDTW",                         //Name of product
       "Big": "10;11;12;",                             //Second board
       "Gb": "",                                       //Not in use
       "Digit": 2,                                      //Number of decimal places
       "Pips": 0,                                       //Not in use
       "Zone": 6,                                       //First board
       "Type": 8,                                       //Not in use
       "TimeType": 1,                                    //Not in use
       "CompanyType": 1,                                //Company type
       "CallOrPutFlag": 0,
       "New64CompanyType": 1,                            //New company type
       "XlsStartEndTimeRange": {
         {
           "StartTime": 0,                               //The data (in minutes) compared with the time 00:00.
           "EndTime": 1440                             //The data (in minutes) compared with the time 00:00.
           For example, if the data is 240, it means 240 minutes have passed from 00:00. If we divide 240
           minutes by 60 minutes (1 hour), the result is 4 hr, which equals 04:00. So it means the market opens
           at 04:00.
         },
         "EndTime": 1440                               //The data (in minutes) compared with the time 00:00.
         For example, if the data is 240, it means 240 minutes have passed from 00:00. If we divide 240
         minutes by 60 minutes (1 hour), the result is 4 hr, which equals 04:00. So it means the market
         closes at 04:00.
       },
       "DlsStartEndTimeRange": {
         {
           "StartTime": 0,                               //The data (in minutes) compared with the time 00:00. For
           "EndTime": 1440                             //The data (in minutes) compared with the time 00:00. For
           example, if the data is 240, it means 240 minutes have passed from 00:00. If we divide 240 minutes
           by 60 minutes (1 hour), the result is 4 hr, which equals 04:00. So it means the market opens at
           04:00.
         },
         "EndTime": 1440                               //The data (in minutes) compared with the time 00:00. For
         example, if the data is 240, it means 240 minutes have passed from 00:00. If we divide 240 minutes
         by 60 minutes (1 hour), the result is 4 hr, which equals 04:00. So it means the market closes at
         04:00.
       }
     ],
     "msg": ""
}

```

3. Market open time data

```

{
  "action": "All Products",
  "data": [
    {"weekday": 0,                                     //0: Sunday, 1: Monday, 2: Tuesday, 3:
Wednesday, 4: Thursday, 5: Friday, 6: Saturday
      "xlsOrDls": 0,                                  //0: Daylight saving time; 1: Standard time
      "szname": "AAPLUSDT",                           //Product name
      "schedule": [
        {"begin": 0,                                   // Start time 00:00
         "end": 1439}                                //The data (in minutes) compared with the
time 00:00. For example, if the data is 240, it means 240 minutes have passed from 00:00. If we
divide 240 minutes by 60 minutes (1 hour), the result is 4 hr, which equals 04:00. So it means the
market closes at 04:00.
      ]
    },
    {
  
```

```

        "weekday": 1,                      //Monday
        "xlsOrDls": 0,                     //0: Daylight saving time; 1: Standard time
        "szname": "AAPLUSDT",              //Product name
        "schedule": [
            {
                "begin": 0,                  // Start time 00:00
                "end": 1439                 //The data (in minutes) compared with the
time 00:00. For example, if the data is 240, it means 240 minutes have passed from 00:00. If we
divide 240 minutes by 60 minutes (1 hour), the result is 4 hr, which equals 04:00. So it means the
market closes at 04:00.
            }
        ],
        {
            "weekday": 3,                  //Wednesday
            "xlsOrDls": 0,                 //0: Daylight saving time; 1: Standard time
            "szname": "AAPLUSDT",          //Product name
            "schedule": [
                {
                    "begin": 0,                  // Start time 00:00
                    "end": 0                   //The data (in minutes) compared with the
time 00:00. For example, if the data is 240, it means 240 minutes have passed from 00:00. If we
divide 240 minutes by 60 minutes (1 hour), the result is 4 hr, which equals 04:00. So it means the
market closes at 04:00.
                }
            ]
        }
    ]
}

```

Request board data

Request JSON parameters:

```
{
    "action": "ReqRealPanel",
    "zone": [ 6, 3, 5, 1, 4, 7, 2 ],           // Request keyword
                                                // It can be one or more board data. The text in boards
are zones in dictionary.
    "seq": 821575647
}
```

Return data:

```
{
    "code": 0,                                // Returned keyword
    "action": "Panel",
    "msg": "",                               // Returned keyword
    "uid": "",                               // Returned keyword
    "data": {
        "Pos": 0,                            //Not in use
        "StockNum": 0,                       //Not in use
        "OptionNum": 0,                      //Not in use
        "IndexNum": 0,                      //Not in use
        "GoldNum": 238,                      // Total number of products
        "Order": 0,                          //Not in use
        "TypeID": 6,                         //Type ID
        "UpDown": 0,                         //Not in use
        "Panel": [
            {
                "CodeId": 3750197,             //Product code
                "YesterdayPrice": 0,          //Yesterday's closing price
                "CurPrice": 0,                //Current price
                "BuyPrice": 0,                //Buy price
                "SellPrice": 0,               //Sell price
                "OpenPrice": 0,               //Open price
                "HighPrice": 0,               //High price
                "LowPrice": 0,                //Low price
                "Change": 0,                 //Price change
                "BidHighPrice": 0,
                "BidLowPrice": 0,
                "AskHighPrice": 0,
                "FlagBid": 0,                //Price change retained on the server 1 up      0 eq
-1 down
                "FlagAsk": 0,                //Price change retained on the server 1 up      0 eq
-1 down
                "FlagCur": 0,                //Price change retained on the server 1 up      0 eq
-1 down
                "Flag": 0,                  //Time
                "Time": 0,                  //Time
                "IsOpen": 1                  //Is the market open
            },
            {
                "CodeId": 3750193,

```

```

        "YesterdayPrice": 0.03695,
        "CurPrice": 0.0364,
        "BuyPrice": 0.03643,
        "SellPrice": 0.03637,
        "OpenPrice": 0.03695,
        "HighPrice": 0.03701,
        "LowPrice": 0.03627,
        "Change": -0.00054999813,
        "BidHighPrice": 11673165,
        "BidLowPrice": 0,
        "AskHighPrice": 0,
        "FlagBid": 0,
        "FlagAsk": 0,
        "FlagCur": 0,
        "Flag": 0,
        "Time": 1693382154,
        "IsOpen": 1
    },
],
}
}

```

Subscribe to market data

Request JSON parameters (Deep's product IDs have to be included in the symbols)

```
{
  "action": "ReqSubcri",           //Request keyword
  "symbols": [
    "3223607",                  //Product code
    "3159350",                  //Product code
    "3355958",                  //Product code
    "3487030"                   //Product code
  ],
  "deep": "3223607"             //Product code in price level quotes
}
```

3 network packets returned (price level quotes, real-time quotes, transaction snapshot)

1. Price level quotes

```
{
  "action": "tickinfo_deep",          // Returned keyword
  "data": [
    {
      "Y": "12856",                // symbolID
      "A": ["316.04", "316.07", "316.20", "316.26", "316.29", "316.38", "316.43"], // priceAsk
      "B": ["314.78", "314.74", "314.69", "314.63", "314.59", "314.53", "314.47"], // priceBid
      "U": ["651.00", "451.00", "613.00", "684.00", "594.00", "684.00", "417.00"], // volumeAsk
      "M": ["1105.00", "467.00", "479.00", "638.00", "723.00", "752.00", "463.00"], // volumeBid
      "T": 1676960685,              // lastTime
      "V": "36.00",                 // lastVolume
      "C": "315.41",                // CurPrice
      "L": ["651.00", "1102.00", "1715.00", "2399.00", "2993.00", "3677.00", "4094.00"], // volAskTotal
      "I": ["1105.00", "1572.00", "2051.00", "2689.00", "3412.00", "4164.00", "4627.00"], // volBidTotal
      "R": ["0.11", "0.19", "0.30", "0.42", "0.53", "0.65", "0.72"], // volAskTotalPer
      "E": ["0.16", "0.23", "0.30", "0.40", "0.51", "0.62", "0.69"] // volBidTotalPer
    }
  ]
}
```

2. Real-time quotes

```
{
  "action": "tickinfo",           // Returned keyword
  "data": [
    {
      "Y": "3159350",            // symbolID
      "A": ["27421.52"],         // priceAsk
      "B": ["27421.02"],         // priceBid
      "T": 1693383923,          // lastTime
      "V": "1.34",                // lastVolume
      "C": "27421.27"           // CurPrice
    }
  ]
}
```

```
        }
    ]
}
```

3. Transaction snapshot

```
{
  "action": "dealticksnap",          // Returned keyword
  "data": [
    {
      "Y": 3223607,                // symbolID
      "ItemCount": 20,              // Data of 20 transactions
      "Items": [
        {
          "P": 27417.81,            //Price
          "V": 0.25,                //Volume
          "T": 1693383923          //Time
        },
        {
          "P": 27417.81,            //Price
          "V": 0.19,                //Volume
          "T": 1693383922          //Time
        },
      ]
    }
}
```

Request candlestick charts

Request JSON parameters

```
{
  "action": "ReqKline",           // Request keyword
  "code": 3355958,               // Product ID
  "interval": 35,                // Candlestick chart type
  "from": 1693366559,           // Start time
  "to": 1693384559,             // End time
}
```

Types of candlestick charts

K_LINE_TYPE_5MIN = 0x01,	// 5m	K_LINE_TYPE_4HOUR = 38,	// 4h
K_LINE_TYPE_15MIN = 0x02, // 15m		K_LINE_TYPE_30MIN = 0x03, // 30m	
K_LINE_TYPE_60MIN = 0x04, // 60m		K_LINE_TYPE_DAY = 0x05, // Daily	
K_LINE_TYPE_WEEK = 0x06, // Weekly		K_LINE_TYPE_MONTH = 0x07, //	
Monthly	K_LINE_TYPE_1MIN = 35, // 1m		K_LINE_TYPE_2HOUR = 37, //
	// 2h		

Returned data

```
{
  "code": 0,
  "action": "Kline",           // Returned keyword
  "msg": "",
  "uid": "tradeChart",
  "data": {
    "Seq": 0,
    "StartTime": 1693364160,
    "CodeId": 3355958,           //Product ID
    "Interval": 35,              //Types of candlestick charts
    "Count": 1,                  //Number of candlesticks
    "KdataInfo": [
      {
        "T": 1566721800,          //Time
        "O": 0.27365,             //Open price
        "C": 0.27355,             //Closing price
        "H": 0.27375,             //Highest price
        "L": 0.2735,               //Lowest price
        "M": 0,                    //Trading volume
        "A": 0
      }
    ]
}
```

```
}
```

Heartbeat API

Request JSON:

```
{
  "action": "KeepLive"
}
```

Returned data:

```
{
  "action": "keeplive"
}
```