**Fex api Instruction Manual**

**V1.0**

[1 Document introduction 2](#_Toc500411259)

[1.1 Special declaration 2](#_Toc500411260)

[1.2 Statement 2](#_Toc500411261)

[2 Network communication protocol 2](#_Toc500411262)

[2.1 Application protocol 2](#_Toc500411263)

[2.2 Data encapsulation format 2](#_Toc500411264)

[3 Interface content 2](#_Toc500411265)

[3.1 Contract backup 2](#_Toc500411266)

[3.1.1 Request path 2](#_Toc500411267)

[3.1.2 Description 2](#_Toc500411268)

[3.1.3 Interface parameters 2](#_Toc500411269)

[3.1.4 Response parameters 3](#_Toc500411270)

# Introduction

## Special declaration

## Statement

This document is an Instruction Manual of installing API

# Network communication protocol

## Application protocol

1. HTTP
2. content-type:"application/json;charset=utf-8"

## Data Encapsulation Format

1. JSON

## Statement on the data standard used in the API

1. Interface is written in java language
2. Use POST request

**3.Interface encryption and Signature instructions**

**3.1 Interface parameter signature**

(1) Pass parameters into Map set: (Different interfaces have different parameters, the following parameters are for reference)

Map<String,String> map = **new** HashMap<String,String>();

map.put("userid", "115006");

map.put("time", t.toString());

map.put("accesskey","eba14683-9f8e-41c9-a23e-b2fdc5cbe4b2");

(2) Generate a signature string:

Generate a Sign signature string with the above encapsulated map set:

String ss = SignUtil.signTopRequestSort(map, secret, "UTF-8");

The secret string is the api private key obtained by FEX platform

Note: generating the Sign signature requires to sort the keys in the map in ascending alphabetical order.

Sorting the order, and use “=””&” to separate the key and value.

Sorting Map order and generate a string such as follow:accesskey=7a7afd85-ffc7-417c-a99e-34da1473fb21&time=211323232323250&userid=21150

Ordered code example:

**public** **static** String formatUrlMap(Map<String, Object> paramMap, **boolean** urlEncode, **boolean** keyToLower) {

String buff = "";

Map<String, Object> tmpMap = paramMap;

**try** {

List<Map.Entry<String, Object>> infoIds = **new** ArrayList<Map.Entry<String, Object>>(tmpMap.entrySet());

// Sorting the all passing-in parameters by ASCII code of field name from A-Z (Dictionary order)

Collections.*sort*(infoIds, **new** Comparator<Map.Entry<String, Object>>() {

@Override

**public** **int** compare(Map.Entry<String, Object> o1, Map.Entry<String, Object> o2) {

**return** (o1.getKey()).toString().compareTo(o2.getKey());

}

});

// Construct a URL key-value pair format

StringBuilder buf = **new** StringBuilder();

**for** (Map.Entry<String, Object> item : infoIds) {

**if** (StringUtil.*isNotBlank*(item.getKey())) {

String key = item.getKey();

Object val = item.getValue();

**if** (urlEncode) {

//val = URLEncoder.encode(val, "utf-8");

}

**if** (keyToLower) {

buf.append(key.toLowerCase() + "=" + val);

}

**else** {

buf.append(key + "=" + val);

}

buf.append("&");

}

}

buff = buf.toString();

**if** (buff.isEmpty() == **false**) {

buff = buff.substring(0, buff.length() - 1);

}

} **catch** (Exception e) {

**return** **null**;

}

**return** buff;

}

Example of generating a signature string code:

**public** **static** String signTopRequestSort(Map<String, Object> params, String secret, String signMethod)

**throws** IOException {

// First: obtain an ordered string

String query = *formatUrlMap*(params, **false**, **false**);

// Second: use MD5/HMAC to encrypt

**byte**[] bytes = **new** **byte**[0];

**if** (Constants.***CHARSET\_UTF8***.equals(signMethod)) {

bytes = *encryptHMAC*(query, secret);

} **else** {

query = query+secret;

**try** {

bytes = *encryptMD5*(query);

} **catch** (NoSuchAlgorithmException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

// Third: Converts binary to uppercase hexadecimal

**return** *byte2hex*(bytes);

}

(3).Put the signature into map set,

Map<String,Object> map = **new** HashMap<String,Object>();

map.put("userid", "115006");

map.put("time","1554359015819");

map.put("accesskey","eba14683-9f8e-41c9-a23e-b2fdc5cbe4b2");

map.put("sign","70FA08C1ACC9C42A18E43C6453AB7039");

And convert map into json string.

{"accesskey":"eba14683-9f8e-41c9-a23e-b2fdc5cbe4b2","sign":"70FA08C1ACC9C42A18E43C6453AB7039","time":"1554362491357","userid":"115006"}

Use BASE 64 to encrypt

String sc = encode(JSON.toJSONString(json).getBytes());

Follow 3.2 to encapsulate the generated encrypted string into json format and then pass the parameters.

**3.2 The interface parameter and interface return value is encrypted/decrypted with BASE64**

Interface parameter use base64 to encrypt, and encapsulate the encrypted string as following Json format:

{"param":"eyJhY2Nlc3NrZXkiOiJlYmExNDY4My05ZjhlLTQxYzktYTIzZS1iMmZkYzVjYmU0YjIiLCJzaWduIjoiOERFNDlBREQ1RjM0RjI4MDhDRUIzRTkzOUZBMEQ4N0YiLCJ0aW1lIjoiMTU1NDM1MDYyNzQzNyIsInVzZXJpZCI6IjExNTAwNiJ9"}

Use “param” string as a key

**3.3 Interface return value parameters**

Interface return parameter needs to do BASE64 encryption

# Interface content

## 4.1 User account asset info

### 4.1.1 Request path

/api/getUserCoinInfo

### 4.1.2 Description

Get user asset info, user available asset, and restrict asset

### 4.1.3 Interface parameter

The result of Interface parameter encapsulation is as follow:

{"param":"eyJhY2Nlc3NrZXkiOiJlYmExND9"}

Parameter details before interface encapsulation:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Data type | Parameter example |
| 1 | userid | Userid | Y | String | String number, for exmaple “115006”,(Convert numbers into a string) |
| 2 | time | Current time stamp (Milli-second) | Y | String | Convert time stamp into string and then pass the parameter, for example:  "1554359015819" |
| 3. | accesskey | Api Private key | Y | String | The obtained API private key value |
| 4. | sign | The above 1,2,3 three parameters generate a SIGN signature string | Y | String | For example :"sign":"9DDFF496CBAAFE73C137201A613EA4E7" |

The interface parameter encapsulation result is as follow:

{"param":"eyJhY2Nlc3NrZXkiOiJlYmExND9"}

### 4.1.4 Response result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Instruction |
| 1 | code | Return status code | Y | 0000：Successful Others：Failed |
| 2 | msg | Return message | N | Calling interface failed, return to fail message |
| 3 | data | Return data | N | Calling interface success, return to success message |

1.Response parameter statements

Use base64 decryption to get response success result, example:

[{"cointypename":"sc","frzzcoinnum":"0","transacoinnum":"10000","userid":"115006"},{"cointypename":"iost","frzzcoinnum":"0","transacoinnum":"10000","userid":"115006"}]

2.Decryption success parameter statement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Field type | Statement |
| 1 | cointypename | Token name | String |  |
| 2 | frzzcoinnum | Restricted asset | String |  |
| 3 | transacoinnum | Available balance | String |  |
| 4 | userid | Userid | String |  |

3.An example of response success result：

{

"code": "1000",

"data": "W3siY29pbnR5cGVuYW1lIjoic2MiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6Imlvc3QiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6ImV0aCIsImZyenpjb2lubnVtIjoiMCIsInRyYW5zYWNvaW5udW0iOiIxMDAwMCIsInVzZXJpZC"

}

BASE64 encrypt the data

4.Response failed result example:

{

"code" : "1002",

"msg" : "Parameter invalid"

}

## 4.2 User order record

### 4.2.1 Request path

/api/getUserEntrustInfo

### 4.2.2 Description

Get user current order, history order, history trading record, info

### 4.2.3 Interface parameter

Parameter details before interface encapsulation:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Data type | Parameter example |
| 1 | userid | Userid | Y | String | String numbers, for example "115006" (Convert numbers into character string) |
| 2 | time | Current time timestamp (Millisecond) | Y | String | Convert time stamp into string and pass the parameter  ex: "1554359015819" |
| 3. | accesskey | Api private key | Y | String | Obtained api private key value |
| 4 | page | Current page | Y | String | Current page number, ex: "1" |
| 5 | pageSize | Page size | Y | String | Number of data per page, ex: “10” |
| 6 | startTime | Check start time | N | String | Format yyyy-mm-dd hh:mm:ss string, ex： "2019-03-23 18:14:22"  Type is 1,this parameter is invalid |
| 7 | endTime | Check end time | N | String | Format yyyy-mm-dd hh:mm:ss string  ex： "2019-03-23 18:14:22"  Type is 1, this parameter is invalid |
| 8 | symbolname | Trading pair name | N | String | Format example： "btcusdt" |
| 9 | type | Type | y | String | Type 1. Current order 2. History order 3. history trading record, ex: "1" |
| 10 | sign | SIGN signature string generated from above parameters | Y | String | EX："sign":"9DDFF496CBAAFE73C137201A613EA4E7" |

The result of interface parameter encapsulation:{"param":"eyJhY2Nlc3NrZXkiOiJlYmExND9"}

### 4.2.4 Response result

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | N | Field | Name | Require to fill | Instruction |
| 1 |  | code | Return status code | Y | 0000：Successful Others：Failed |
| 2 |  | msg | Return message | N | Calling interface failed, return to fail message |
| 3 |  | data | Return data | N | Calling interface success, return to success message |

1. Response parameter statement

Use base64 decryption to get response success result:

1. Decryption success parameter statement

Type1,2 Parameter statement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Instruction |
| 1 | count | Data amount | Integer | 85 |
| 2 | list | Data set | List |  |
| 3 | coinnum | Order number | String | "10000" |
| 4 | orderPrice | Order price | String | "0.00012" |
| 5 | ordertime | Order time | Time stamp | 1553618183000 |
| 6 | status | Status | Integer | Status 0. Unfilled 1. Partial deal 2. All complete 3.Withdraw order 4. Partial deal completed |
| 7 | symbolname | Trading pair name | String | ethbtc |
| 8 | transaNum | Trading amount | String | "0.0132" |
| 9 | transaPrice | Trading price | String | "0.0132" |
| 10 | transatotalamount | Total trading amount | String | "0.2132" |
| 11 | transafee | Trading fee | String | "0.002" |

Type 3. Parameter statement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Instruction |
| 1 | count | Data amount | Integer | 85 |
| 2 | list | Data set | List |  |
| 3 | symbolname | Trading pair name | String | "ethbtc" |
| 4 | transafee | fee | String | "0.00012" |
| 5 | transanum | Trading amount | String | "0.20012" |
| 6 | transaprice | Trading price | String | "0.20012" |
| 7 | transatime | Trading time | Time stamp | 1554325933000 |

3.Response success result example:

{

"code": "1000",

"data": "W3siY29pbnR5cGVuYW1lIjoic2MiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6Imlvc3QiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6ImV0aCIsImZyenpjb2lubnVtIjoiMCIsInRyYW5zYWNvaW5udW0iOiIxMDAwMCIsInVzZXJpZC"

}

BASE64 decrypt data

4.Response failed success example:

{

"code" : "1002",

"msg" : "Parameter invalid"

}

## 4.3 Financial record

### 4.3.1Request path

/api/getRolecoinchange

### 4.3.2Description

Get user financial record

### 4.3.3 Interface parameter

Parameter details before interface encapsulation:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Data type | Parameter example |
| 1 | userid | User id | Y | String | String numbers, for example "115006" (Convert numbers into character string) |
| 2 | time | Current time stamp (Milli-second) | Y | String | Convert time stamp into string and pass the parameter  ex: "1554359015819" |
| 3. | accesskey | API private key | Y | String | Obtained api private key value |
| 4 | page | Current page | Y | String | Current page number, ex: "1" |
| 5 | pageSize | Page size | Y | String | Data amount on each page, for example: "10" |
| 6 | startTime | Check start time | N | String | Format yyyy-mm-dd hh:mm:ss string  Example： "2019-03-23 18:14:22" |
| 7 | endTime | Check end time | N | String | Format yyyy-mm-dd hh:mm:ss string  Example： "2019-03-23 18:14:22" |
| 8 | coinname | Token name | N | String | Format： "usdt" |
| 9 | sign | SIGN signature string generated from the above parameters | Y | String | Example："sign":"9DDFF496CBAAFE73C137201A613EA4E7" |

The result of interface parameter encapsulation:

{"param":"eyJhY2Nlc3NrZXkiOiJlYmExND9"}

### 4.3.4Response result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Instruction |
| 1 | code | Return status code | Y | 0000：Successful Others：Failed |
| 2 | msg | Return message | N | Calling interface failed, return to fail message |
| 3 | data | Return data | N | Calling interface success, return to success message |

1. Response parameter statement

Use base64 decryption to get response success result:

2. Decryption success parameter statement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Instruction |
| 1 | count | Amount | Integer | 25 |
| 2 | list | Data set | List |  |
| 2 | changeCoin | Change amount | String | "2" |
| 3 | changetype | Change type 0、Sell，1 Buy | Integer | 0 |
| 4 | coinname | Token name | String | usdt |
| 5 | endCoin | Amount after changed | String |  |
| 6 | orderNum | Order amount | String |  |
| 7 | orderPrice | Order price | String |  |
| 8 | proCoin | Amount before changed | String |  |
| 9 | time | Record creation time | Time stamp |  |
| 10 | userid | User id |  |  |

3.Response success result example:

{

"code": "1000",

"data": "W3siY29pbnR5cGVuYW1lIjoic2MiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6Imlvc3QiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6ImV0aCIsImZyenpjb2lubnVtIjoiMCIsInRyYW5zYWNvaW5udW0iOiIxMDAwMCIsInVzZXJpZC"

}

BASE64 decrypt data

4.Response failed success example:

{

"code" : "1002",

"msg" : "Parameter invalid"

}

## 4.4 Last 500 trading record of a user

### 4.4.1Request path

/api/getTrade

### 4.4.2Description

Get last 500 trading record of a user

### 4.4.3 Interface parameter

Parameter details before interface encapsulation:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Data type | Parameter example |
| 1 | userid | User id | Y | String | String numbers, for example "115006" (Convert numbers into character string) |
| 2 | time | Current time stamp (Milli-second) | Y | String | Convert time stamp into string and pass the parameter  ex: "1554359015819" |
| 3. | accesskey | API private key | Y | String | Obtained api private key value |
| 4 | symbol | Trading pair name | Y | String | godusdt |
| 5. | sign | SIGN signature string generated from the above parameters | Y | String | 如："sign":"9DDFF496CBAAFE73C137201A613EA4E7" |

The result of interface parameter encapsulation:

{"param":"eyJhY2Nlc3NrZXkiOiJlYmExND9"}

### 4.4.4Response result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Instruction |
| 1 | code | Return status code | Y | 0000：Successful Others：Failed |
| 2 | msg | Return message | N | Calling interface failed, return to fail message |
| 3 | data | Return data | N | Calling interface success, return to success message |

1. Response parameter statement

Use base64 decryption to get response success result:

[{"cointypename":"sc","frzzcoinnum":"0","transacoinnum":"10000","userid":"115006"},{"cointypename":"iost","frzzcoinnum":"0","transacoinnum":"10000","userid":"115006"}]

2. Decryption success parameter statement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Instruction |
| 1 | symbolname | Trading pair name | String | "ethbtc" |
| 2 | transafee | Trading fee | String | "0.00012" |
| 3 | transanum | Trading amount | String | "0.20012" |
| 4 | transaprice | Trading price | String | "0.20012" |
| 5 | transatime | Trading time | Time stamp | 1554325933000 |

3.Response success result example:

{

"code": "1000",

"data": "W3siY29pbnR5cGVuYW1lIjoic2MiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6Imlvc3QiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6ImV0aCIsImZyenpjb2lubnVtIjoiMCIsInRyYW5zYWNvaW5udW0iOiIxMDAwMCIsInVzZXJpZC"

}

BASE64 decrypt data

4.Response failed success example:

{

"code" : "1002",

"msg" : "Parameter invalid"

}

## 4.5 User visit Buy and sell order of a trading pair

### 4.5.1Request path

/api/getSymbolPrice

### 4.5.2Description

Get trading pair buy and sell order

### 4.5.3 Interface parameter

Parameter details before interface encapsulation:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Data type | Parameter example |
| 1 | userid | User id | Y | String | String numbers, for example "115006" (Convert numbers into character string) |
| 2 | time | Current time stamp (Milli-second) | Y | String | Convert time stamp into string and pass the parameter  ex: "1554359015819" |
| 3. | accesskey | API private key | Y | String | Obtained api private key value |
| 4 | symbol | Trading pair name | Y | String | usdtbtc |
| 5. | sign | SIGN signature string generated from the above parameters | Y | String | example："sign":"9DDFF496CBAAFE73C137201A613EA4E7" |

The result of interface parameter encapsulation:

{"param":"eyJhY2Nlc3NrZXkiOiJlYmExND9"}

### 4.5.4Response result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Instruction |
| 1 | code | Return status code | Y | 0000：Successful Others：Failed |
| 2 | msg | Return message | N | Calling interface failed, return to fail message |
| 3 | data | Return data | N | Calling interface success, return to success message |

1. Response parameter statement

Use base64 decryption to get response success result:

[{"cointypename":"sc","frzzcoinnum":"0","transacoinnum":"10000","userid":"115006"},{"cointypename":"iost","frzzcoinnum":"0","transacoinnum":"10000","userid":"115006"}]

2. Decryption success parameter statement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Instruction |
| 1 |  |  |  |  |

3.Response success result example:

{

"code": "1000",

"data": "W3siY29pbnR5cGVuYW1lIjoic2MiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6Imlvc3QiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6ImV0aCIsImZyenpjb2lubnVtIjoiMCIsInRyYW5zYWNvaW5udW0iOiIxMDAwMCIsInVzZXJpZC"

}

BASE64 decrypt data

4.Response failed success example:

{

"code" : "1002",

"msg" : "Parameter invalid"

}

## 4.6 Buy at limited price

### 4.6.1Request path

/api/buy

### 4.6.2Description

Buy at limited price

### 4.6.3 Interface parameter

Parameter details before interface encapsulation:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Data type | Parameter example |
| 1 | userid | User id | Y | String | String numbers, for example "115006" (Convert numbers into character string) |
| 2 | time | Current time stamp (Milli-second) | Y | String | Convert time stamp into string and pass the parameter  ex: "1554359015819" |
| 3. | accesskey | API private key | Y | String | Obtained api private key value |
| 4 | price | Buy price | Y | Double | Buy price Example:12.023 |
| 5 | num | Buy amount | Y | Double | Buy amount example:12.023 |
| 6 | symbol | Trading pair name | Y | String | Example: btcusdt |
| 7 | sign | SIGN signature string generated from the above parameters | Y | String | Example："sign":"9DDFF496CBAAFE73C137201A613EA4E7" |

The result of interface parameter encapsulation:

{"param":"eyJhY2Nlc3NrZXkiOiJlYmExND9"}

### 4.6.4Response result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Instruction |
| 1 | code | Return status code | Y | 0000：Successful Others：Failed |
| 2 | msg | Return message | N | Calling interface failed, return to fail message |
| 3 | data | Return data | N | Calling interface success, return to success message |

1. Response parameter statement

Use base64 decryption to get response success result:

2. Decryption success parameter statement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Instruction |
| 1 | symbol | Trading pair name | String |  |
| 2 | orderhash | Order hash | String |  |
| 3 | time | Current time stamp | Long |  |

3.Response success result example:

{

"code": "1000",

"data": "W3siY29pbnR5cGVuYW1lIjoic2MiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6Imlvc3QiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6ImV0aCIsImZyenpjb2lubnVtIjoiMCIsInRyYW5zYWNvaW5udW0iOiIxMDAwMCIsInVzZXJpZC"

}

BASE64 decrypt data

4.Response failed success example:

{

"code" : "1002",

"msg" : "Parameter invalid"

}

## 4.7 Sell at limited price

### 4.7.1Request path

/api/sell

### 4.7.2Description

Sell at limited price

### 4.7.3 Interface parameter

Parameter details before interface encapsulation:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Data type | Parameter example |
| 1 | userid | User id | Y | String | String numbers, for example "115006" (Convert numbers into character string) |
| 2 | time | Current time stamp (Milli-second) | Y | String | Convert time stamp into string and pass the parameter  ex: "1554359015819" |
| 3. | accesskey | API private key | Y | String | Obtained api private key value |
| 4 | price | Sell price | Y | Double | Sell price example:12.023 |
| 5 | num | Sell amount | Y | Double | Sell amount example:12.023 |
| 6 | symbol | Trading pair name | Y | String | Example：btcusdt |
| 7 | sign | SIGN signature string generated from the above parameters | Y | String | Example："sign":"9DDFF496CBAAFE73C137201A613EA4E7" |

The result of interface parameter encapsulation:

{"param":"eyJhY2Nlc3NrZXkiOiJlYmExND9"}

### 4.7.4Response result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Instruction |
| 1 | code | Return status code | Y | 0000：Successful Others：Failed |
| 2 | msg | Return message | N | Calling interface failed, return to fail message |
| 3 | data | Return data | N | Calling interface success, return to success message |

1. Response parameter statement

Use base64 decryption to get response success result:

2. Decryption success parameter statement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Instruction |
| 1 | symbol | Trading pair name | String |  |
| 2 | orderhash | Order hash | String |  |
| 3 | time | Current time stamp | Long |  |

3.Response success result example:

{

"code": "1000",

"data": "W3siY29pbnR5cGVuYW1lIjoic2MiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6Imlvc3QiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6ImV0aCIsImZyenpjb2lubnVtIjoiMCIsInRyYW5zYWNvaW5udW0iOiIxMDAwMCIsInVzZXJpZC"

}

BASE64 decrypt data

4.Response failed success example:

{

"code" : "1002",

"msg" : "Parameter invalid"

}

## 4.8 Cancel order

### 4.8.1Request path

/api/cancleOrder

### 4.8.2Description

Cancel order

### 4.8.3 Interface parameter

Parameter details before interface encapsulation:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Data type | Parameter example |
| 1 | userid | User id | Y | String | String numbers, for example "115006" (Convert numbers into character string) |
| 2 | time | Current time stamp (Milli-second) | Y | String | Convert time stamp into string and pass the parameter  ex: "1554359015819" |
| 3. | accesskey | API private key | Y | String | Obtained api private key value |
| 4 | orderHash | Order hash value | Y | String | Order hash example：1802824f75cf74e90d48db8f479ef917703e3d |
| 5 | symbol | Trading pair name | Y | String | example：btcusdt |
| 6 | sign | SIGN signature string generated from the above parameters | Y | String | example："sign":"9DDFF496CBAAFE73C137201A613EA4E7" |

The result of interface parameter encapsulation:

{"param":"eyJhY2Nlc3NrZXkiOiJlYmExND9"}

### 4.8.4Response result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Instruction |
| 1 | code | Return status code | Y | 0000：Successful Others：Failed |
| 2 | msg | Return message | N | Calling interface failed, return to fail message |
| 3 | data | Return data | N | Calling interface success, return to success message |

1. Response parameter statement

Use base64 decryption to get response success result:

2. Decryption success parameter statement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Instruction |
| 1 | symbol | Trading pair name | String |  |
| 2 | orderHash | Order hash | String |  |
| 3 | time | Current time stamp | Long |  |

3.Response success result example:

{

"code": "1000",

"data": "W3siY29pbnR5cGVuYW1lIjoic2MiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6Imlvc3QiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6ImV0aCIsImZyenpjb2lubnVtIjoiMCIsInRyYW5zYWNvaW5udW0iOiIxMDAwMCIsInVzZXJpZC"

}

BASE64 decrypt data

4.Response failed success example:

{

"code" : "1002",

"msg" : "Parameter invalid"

}

## 4.9 Buy at market price

### 4.9.1Request path

/api/currentbuy

### 4.9.2Description

Buy at market price

### 4.9.3 Interface parameter

Parameter details before interface encapsulation:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Data type | Parameter example |
| 1 | userid | User id | Y | String | String numbers, for example "115006" (Convert numbers into character string) |
| 2 | time | Current time stamp (Milli-second) | Y | String | Convert time stamp into string and pass the parameter  ex: "1554359015819" |
| 3. | accesskey | API private key | Y | String | Obtained api private key value |
| 4 | transaNum | Trading amount | Y | Double | The trading amount of buying at a market price example：12.23 |
| 5 | symbol | Trading pair name | Y | String | Example：btcusdt |
| 6 | sign | SIGN signature string generated from the above parameters | Y | String | Example："sign":"9DDFF496CBAAFE73C137201A613EA4E7" |

The result of interface parameter encapsulation:

{"param":"eyJhY2Nlc3NrZXkiOiJlYmExND9"}

### 4.9.4Response result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Instruction |
| 1 | code | Return status code | Y | 0000：Successful Others：Failed |
| 2 | msg | Return message | N | Calling interface failed, return to fail message |
| 3 | data | Return data | N | Calling interface success, return to success message |

1. Response parameter statement

Use base64 decryption to get response success result:

2. Decryption success parameter statement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Instruction |
| 1 | symbol | Trading pair name | String |  |
| 2 | orderHash | Order hash | String |  |
| 3 | time | Current time stamp | Long |  |

3.Response success result example:

{

"code": "1000",

"data": "W3siY29pbnR5cGVuYW1lIjoic2MiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6Imlvc3QiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6ImV0aCIsImZyenpjb2lubnVtIjoiMCIsInRyYW5zYWNvaW5udW0iOiIxMDAwMCIsInVzZXJpZC"

}

BASE64 decrypt data

4.Response failed success example:

{

"code" : "1002",

"msg" : "Parameter invalid"

}

## 4.10 Sell at market price

### 4.10.1Request path

/api/currentsell

### 4.10.2Description

Sell at market price

### 4.10.3 Interface parameter

Parameter details before interface encapsulation:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Data type | Parameter example |
| 1 | userid | User id | Y | String | String numbers, for example "115006" (Convert numbers into character string) |
| 2 | time | Current time stamp (Milli-second) | Y | String | Convert time stamp into string and pass the parameter  ex: "1554359015819" |
| 3. | accesskey | API private key | Y | String | Obtained api private key value |
| 4 | num | Trading amount | Y | Double | The trading amount of selling at a market price example：12.23 |
| 5 | symbol | Trading pair name | Y | String | Example：btcusdt |
| 6 | sign | SIGN signature string generated from the above parameters | Y | String | Example："sign":"9DDFF496CBAAFE73C137201A613EA4E7" |

The result of interface parameter encapsulation:

{"param":"eyJhY2Nlc3NrZXkiOiJlYmExND9"}

### 4.9.4Response result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Instruction |
| 1 | code | Return status code | Y | 0000：Successful Others：Failed |
| 2 | msg | Return message | N | Calling interface failed, return to fail message |
| 3 | data | Return data | N | Calling interface success, return to success message |

1. Response parameter statement

Use base64 decryption to get response success result:

2. Decryption success parameter statement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Field | Name | Require to fill | Instruction |
| 1 | symbol | Trading pair name | String |  |
| 2 | orderHash | Order hash | String |  |
| 3 | Time | Current time stamp | Long |  |

3.Response success result example:

{

"code": "1000",

"data": "W3siY29pbnR5cGVuYW1lIjoic2MiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6Imlvc3QiLCJmcnp6Y29pbm51bSI6IjAiLCJ0cmFuc2Fjb2lubnVtIjoiMTAwMDAiLCJ1c2VyaWQiOiIxMTUwMDYifSx7ImNvaW50eXBlbmFtZSI6ImV0aCIsImZyenpjb2lubnVtIjoiMCIsInRyYW5zYWNvaW5udW0iOiIxMDAwMCIsInVzZXJpZC"

}

BASE64 decrypt data

4.Response failed success example:

{

"code" : "1002",

"msg" : "Parameter invalid"

}