

- [API](#)
- [Broker](#)
- [Best Practice](#)
- [Change Log](#)

API ▼

[中文](#)

- [Overview](#)
- [Trading Account](#)
- [Order Book Trading](#)
- [Block Trading](#)
- [Spread Trading](#)
- [Public Data](#)
- [Trading Statistics](#)
- [Funding Account](#)
- [Sub-account](#)
- [Financial Product](#)
- [Affiliate](#)
- [Status](#)
- [Announcement](#)
- [Error Code](#)

## Overview

Welcome to our API documentation. OKX provides REST and WebSocket APIs to suit your trading needs.

- For users who complete registration on [my.okx.com](https://my.okx.com), please visit <https://my.okx.com/docs-v5/en/> for the API documentation.
- For users who complete registration on [app.okx.com](https://app.okx.com), please visit <https://app.okx.com/docs-v5/en/> for the API documentation.

## API Resources and Support

### TUTORIALS

- Learn how to trade with API: Best practice to OKX's API
- Learn python spot trading step by step: Python Spot Trading Tutorial
- Learn python derivatives trading step by step: Python Derivatives Trading Tutorial

### PYTHON LIBRARIES

- Use Python SDK for easier integration: Python SDK
- Get access to our market maker python sample code Python market maker sample

### CUSTOMER SERVICE

- Please take 1 minute to help us improve: API Satisfaction Survey
- If you have any questions, please consult online customer service

# API key Creation

Please refer to my api page regarding API Key creation.

## GENERATING AN API KEY

Create an API key on the website before signing any requests. After creating an API key, keep the following information safe:

- API key
- Secret key
- Passphrase

The system returns randomly-generated API keys and SecretKeys. You will need to provide the Passphrase to access the API. We store the salted hash of your Passphrase for authentication. We cannot recover the Passphrase if you have lost it. You will need to create a new set of API key.

## API KEY PERMISSIONS

There are three permissions below that can be associated with an API key. One or more permission can be assigned to any key.

- **Read** : Can request and view account info such as bills and order history which need read permission
- **Trade** : Can place and cancel orders, funding transfer, make settings which need write permission
- **Withdraw** : Can make withdrawals

## API KEY SECURITY

To improve security, we strongly recommend clients linked the API key to IP addresses

- Each API key can bind up to 20 IP addresses, which support IPv4/IPv6 and network segment formats.

API keys that are not linked to an IP address and have 'trade' or 'withdraw' permissions will expire after 14 days of inactivity. (The API key of demo trading will not expire)

- Only when the user calls an API that requires API key authentication will it be considered as the API key is used.
- Calling an API that does not require API key authentication will not be considered used even if API key information is passed in.
- For websocket, only operation of logging in will be considered to have used the API key. Any operation through the connection after logging in (such as subscribing/placing an order) will not be considered to have used the API key. Please pay attention.

Users can get the usage records of the API key with **trade** or **withdraw** permissions but unlinked to any IP address through Security Center.

# REST Authentication

## MAKING REQUESTS

All private REST requests must contain the following headers:

- **OK-ACCESS-KEY** The API key as a String.
- **OK-ACCESS-SIGN** The Base64-encoded signature (see Signing Messages subsection for details).
- **OK-ACCESS-TIMESTAMP** The UTC timestamp of your request .e.g : 2020-12-08T09:08:57.715Z
- **OK-ACCESS-PASSPHRASE** The passphrase you specified when creating the API key.

Request bodies should have content type **application/json** and be in valid JSON format.

## SIGNATURE

### Signing Messages

The **OK-ACCESS-SIGN** header is generated as follows:

- Create a pre-hash string of timestamp + method + requestPath + body (where + represents String concatenation).

- Prepare the SecretKey.
- Sign the pre-hash string with the SecretKey using the HMAC SHA256.
- Encode the signature in the Base64 format.

Example: `sign= CryptoJS.enc.Base64.stringify(CryptoJS.HmacSHA256(timestamp + 'GET' + '/api/v5/account/balance?ccy=BTC', SecretKey))`

The `timestamp` value is the same as the `OK-ACCESS-TIMESTAMP` header with millisecond ISO format, e.g. `2020-12-08T09:08:57.715Z`.

The request method should be in UPPERCASE: e.g. `GET` and `POST`.

`requestPath` is the path of requesting an endpoint.

Example: `/api/v5/account/balance`

`body` refers to the String of the request body. It can be omitted if there is no request body (frequently the case for `GET` requests).

Example: `{"instId": "BTC-USDT", "lever": "5", "mgnMode": "isolated"}`

'GET' request parameters are counted as requestpath, not body

The SecretKey is generated when you create an API key.

Example: `22582BD0CFF14C41EDBF1AB98506286D`

## WebSocket

### OVERVIEW

WebSocket is a new HTML5 protocol that achieves full-duplex data transmission between the client and server, allowing data to be transferred effectively in both directions. A connection between the client and server can be established with just one handshake. The server will then be able to push data to the client according to preset rules. Its advantages include:

- The WebSocket request header size for data transmission between client and server is only 2 bytes.
- Either the client or server can initiate data transmission.
- There's no need to repeatedly create and delete TCP connections, saving resources on bandwidth and server.

We recommend developers use WebSocket API to retrieve market data and order book depth.

### CONNECT

**Connection limit:** 3 requests per second (based on IP)

When subscribing to a public channel, use the address of the public service. When subscribing to a private channel, use the address of the private service

**Request limit:**

The total number of 'subscribe'/'unsubscribe'/'login' requests per connection is limited to 480 times per hour.

If there's a network problem, the system will automatically disable the connection.

The connection will break automatically if the subscription is not established or data has not been pushed for more than 30 seconds.

To keep the connection stable:

1. Set a timer of N seconds whenever a response message is received, where N is less than 30.
2. If the timer is triggered, which means that no new message is received within N seconds, send the String 'ping'.

3. Expect a 'pong' as a response. If the response message is not received within N seconds, please raise an error or reconnect.

## CONNECTION COUNT LIMIT

The limit will be set at 30 WebSocket connections per specific WebSocket channel per sub-account. Each WebSocket connection is identified by the unique `connId`.

The WebSocket channels subject to this limitation are as follows:

1. Orders channel
2. Account channel
3. Positions channel
4. Balance and positions channel
5. Position risk warning channel
6. Account greeks channel

If users subscribe to the same channel through the same WebSocket connection through multiple arguments, for example, by using `{"channel": "orders", "instType": "ANY"}` and `{"channel": "orders", "instType": "SWAP"}`, it will be counted once only. If users subscribe to the listed channels (such as orders and accounts) using either the same or different connections, it will not affect the counting, as these are considered as two different channels. The system calculates the number of WebSocket connections per channel.

The platform will send the number of active connections to clients through the `channel-conn-count` event message **to new channel subscriptions**.

### Connection count update

```
{  
  "event": "channel-conn-count",  
  "channel": "orders",  
  "connCount": "2",  
  "connId": "abcd1234"  
}
```

When the limit is breached, generally the latest connection that sends the subscription request will be rejected. Client will receive the usual subscription acknowledgement followed by the `channel-conn-count-error` from the connection that the subscription has been terminated. In exceptional circumstances the platform may unsubscribe existing connections.

### Connection limit error

```
{  
  "event": "channel-conn-count-error",  
  "channel": "orders",  
  "connCount": "20",  
  "connId": "a4d3ae55"  
}
```

Order operations through WebSocket, including place, amend and cancel orders, are not impacted through this change.

## LOGIN

### Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
op	String	Yes	Operation <code>login</code>

Parameter	Type	Required	Description
args	Array of objects	Yes	List of account to login
> apiKey	String	Yes	API Key
> passphrase	String	Yes	API Key password
> timestamp	String	Yes	Unix Epoch time, the unit is seconds
> sign	String	Yes	Signature string

#### Successful Response Example

```
{
  "event": "login",
  "code": "0",
  "msg": "",
  "connId": "a4d3ae55"
}
```

#### Failure Response Example

```
{
  "event": "error",
  "code": "60009",
  "msg": "Login failed.",
  "connId": "a4d3ae55"
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Required	Description
event	String	Yes	Operation <code>login</code> <code>error</code>
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

**apiKey:** Unique identification for invoking API. Requires user to apply one manually.

**passphrase:** API Key password

**timestamp:** the Unix Epoch time, the unit is seconds, e.g. 1704876947

**sign:** signature string, the signature algorithm is as follows:

First concatenate `timestamp`, `method`, `requestPath`, strings, then use HMAC SHA256 method to encrypt the concatenated string with `SecretKey`, and then perform Base64 encoding.

**secretKey:** The security key generated when the user applies for API key, e.g. `22582BD0cff14c41edbf1ab98506286d`

**Example of timestamp:** `const timestamp = " + Date.now() / 1,000`

**Among sign example:** `sign=CryptoJS.enc.Base64.stringify(CryptoJS.HmacSHA256(timestamp + 'GET' + '/users/self/verify', secretKey))`

**method:** always 'GET'.

**requestPath :** always '/users/self/verify'

The request will expire 30 seconds after the timestamp. If your server time differs from the API server time, we recommended using the REST API to query the API server time and then set the timestamp.

## SUBSCRIBE

### Subscription Instructions

Request format description

WebSocket channels are divided into two categories: **public** and **private** channels.

**Public channels** -- No authentication is required, include tickers channel, K-Line channel, limit price channel, order book channel, and mark price channel etc.

**Private channels** -- including account channel, order channel, and position channel, etc -- require log in.

Users can choose to subscribe to one or more channels, and the total length of multiple channels cannot exceed 64 KB.

Below is an example of subscription parameters. The requirement of subscription parameters for each channel is different. For details please refer to the specification of each channels.

Request Example

### Request parameters

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <b>subscribe</b>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name
> instType	String	No	Instrument type <b>SPOT</b> <b>MARGIN</b> <b>SWAP</b> <b>FUTURES</b> <b>OPTION</b> <b>ANY</b>
> instFamily	String	No	Instrument family Applicable to <b>FUTURES</b> / <b>SWAP</b> / <b>OPTION</b>
> instId	String	No	Instrument ID

Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "tickers",
    "instId": "BTC-USDT"
  },
  "connId": "accb8e21"
}
```

### Return parameters

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event  
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> instType	String	No	Instrument type      
> instFamily	String	No	Instrument family Applicable to   
> instId	String	No	Instrument ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

## UNSUBSCRIBE

Unsubscribe from one or more channels.

Request format description

Request Example

## Request parameters

Parameter	Type	Required	Description
op	String	Yes	Operation 
args	Array of objects	Yes	List of channels to unsubscribe from
> channel	String	Yes	Channel name
> instType	String	No	Instrument type      
> instFamily	String	No	Instrument family Applicable to   
> instId	String	No	Instrument ID

Response Example

```
{
  "event": "unsubscribe",
  "arg": {
    "channel": "tickers",
    "instId": "BTC-USDT"
  },
  "connId": "d0b44253"
}
```

## Response parameters

Parameter	Type	Required	Description
event	String	Yes	Event unsubscribe error
arg	Object	No	Unsubscribed channel
> channel	String	Yes	Channel name
> instType	String	No	Instrument type SPOT MARGIN SWAP FUTURES OPTION
> instFamily	String	No	Instrument family Applicable to FUTURES / SWAP / OPTION
> instId	String	No	Instrument ID
code	String	No	Error code
msg	String	No	Error message

## NOTIFICATION

WebSocket has introduced a new message type (event = `notice`).

Client will receive the information in the following scenarios:

- Websocket disconnect for service upgrade

60 seconds prior to the upgrade of the WebSocket service, the notification message will be sent to users indicating that the connection will soon be disconnected. Users are encouraged to establish a new connection to prevent any disruptions caused by disconnection.

### Response Example

```
{
  "event": "notice",
  "code": "64008",
  "msg": "The connection will soon be closed for a service upgrade. Please reconnect.",
  "connId": "a4d3ae55"
}
```

The feature is supported by WebSocket Public (/ws/v5/public) and Private (/ws/v5/private) for now.

## Account mode

To facilitate your trading experience, please set the appropriate account mode before starting trading.

In the trading account trading system, 4 account modes are supported: [Spot mode](#), [Futures mode](#), [Multi-currency margin mode](#), and [Portfolio margin mode](#).

You need to set on the Web/App for the first set of every account mode.

## Production Trading Services

The Production Trading URL:

- REST: <https://www.okx.com>
- Public WebSocket: <wss://ws.okx.com:8443/ws/v5/public>
- Private WebSocket: <wss://ws.okx.com:8443/ws/v5/private>
- Business WebSocket: <wss://ws.okx.com:8443/ws/v5/business>

## Demo Trading Services

Currently, the API works for Demo Trading, but some functions are not supported, such as [withdraw](#), [deposit](#), [purchase/redemption](#), etc.

The Demo Trading URL:

- REST: <https://www.okx.com>
- Public WebSocket: <wss://wspap.okx.com:8443/ws/v5/public>
- Private WebSocket: <wss://wspap.okx.com:8443/ws/v5/private>
- Business WebSocket: <wss://wspap.okx.com:8443/ws/v5/business>

OKX account can be used for login on Demo Trading. If you already have an OKX account, you can log in directly.

Start API Demo Trading by the following steps:

Login OKX —> Trade —> Demo Trading —> Personal Center —> Demo Trading API -> Create Demo Trading API Key —> Start your Demo Trading

Note: `x-simulated-trading: 1` needs to be added to the header of the Demo Trading request.

### Http Header Example

```
Content-Type: application/json

OK-ACCESS-KEY: 37c541a1_*****_*****_*****-10fe7a038418

OK-ACCESS-SIGN: leaVRETrtaoEQ3yI9qEtI1CZ82ikZ4xSG5Kj8gnl3uw=

OK-ACCESS-PASSPHRASE: 1*****6

OK-ACCESS-TIMESTAMP: 2020-03-28T12:21:41.274Z

x-simulated-trading: 1
```

### DEMO TRADING EXPLORER

You need to sign in to your OKX account before accessing the explorer. The interface only allow access to the demo trading environment.

- Clicking [Try it out](#) button in Parameters Panel and editing request parameters.
- Clicking [Execute](#) button to send your request. You can check response in Responses panel.

Try demo trading explorer

# General Info

## The rules for placing orders at the exchange level are as follows:

- The maximum number of pending orders (including post only orders, limit orders and taker orders that are being processed): 4,000
- The maximum number of pending orders per trading symbol is 500, the limit of 500 pending orders applies to the following **order types**:
  - Limit
  - Market
  - Post only
  - Fill or Kill (FOK)
  - Immediate or Cancel (IOC)
  - Market order with Immediate-or-Cancel order (optimal limit IOC)
  - Take Profit / Stop Loss (TP/SL)
  - Limit and market orders triggered under the order types below:
    - Take Profit / Stop Loss (TP/SL)
    - Trigger
    - Trailing stop
    - Arbitrage
    - Iceberg
    - TWAP
    - Recurring buy
- The maximum number of pending spread orders: 500 across all spreads
- The maximum number of pending algo orders:
  - TP/SL order: 100 per instrument
  - Trigger order: 500
  - Trailing order: 50
  - Iceberg order: 100
  - TWAP order: 20
- The maximum number of grid trading
  - Spot grid: 100
  - Contract grid: 100

## The rules for trading are as follows:

- When the number of maker orders matched with a taker order exceeds the maximum number limit of 1000, the taker order will be canceled.
  - The limit orders will only be executed with a portion corresponding to 1000 maker orders and the remainder will be canceled.
  - Fill or Kill (FOK) orders will be canceled directly.

## The rules for the returning data are as follows:

- `code` and `msg` represent the request result or error reason when the return data has `code`, and has not `sCode`;
- It is `sCode` and `sMsg` that represent the request result or error reason when the return data has `sCode` rather than `code` and `msg`.

## `instFamily` and `uly` parameter explanation:

- The following explanation is based on the `BTC` contract, other contracts are similar.
- `uly` is the index, like "BTC-USD", and there is a one-to-many relationship with the settlement and margin currency (`settleCcy`).
- `instFamily` is the trading instrument family, like `BTC-USD_UM`, and there is a one-to-one relationship with the settlement and margin currency (`settleCcy`).
- The following table shows the corresponding relationship of `uly`, `instFamily`, `settleCcy` and `instId`.

Contract Type	uly	instFamily	settleCcy	Delivery contract instId	Swap contract instId
USDT-margined contract	BTC-USDT	BTC-USDT	USDT	BTC-USDT-250808	BTC-USDT-SWAP

Contract Type	uly	instFamily	settleCcy	Delivery contract instId	Swap contract instId
USDC-margined contract	BTC-USDC	BTC-USDC	USDC	BTC-USDC-250808	BTC-USDC-SWAP
USD-margined contract	BTC-USD	<b>BTC-USD_UM</b>	<b>USD\$</b>	<b>BTC-USD_UM-250808</b>	<b>BTC-USD_UM-SWAP</b>
Coin-margined contract	BTC-USD	<b>BTC-USD</b>	<b>BTC</b>	<b>BTC-USD-250808</b>	<b>BTC-USD-SWAP</b>

Note:

1. USD\$ represents USD and multiple USD stable coins, like USDC, USDG.
2. The settlement and margin currency refers to the `settleCcy` field returned by the Get instruments endpoint.

## Transaction Timeouts

Orders may not be processed in time due to network delay or busy OKX servers. You can configure the expiry time of the request using `expTime` if you want the order request to be discarded after a specific time.

If `expTime` is specified in the requests for Place (multiple) orders or Amend (multiple) orders, the request will not be processed if the current system time of the server is after the `expTime`.

### REST API

Set the following parameters in the request header

Parameter	Type	Required	Description
expTime	String	No	Request effective deadline. Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

The following endpoints are supported:

- Place order
- Place multiple orders
- Amend order
- Amend multiple orders
- POST / Place sub order under signal bot trading

### Request Example

```
curl -X 'POST' \
  'https://www.okx.com/api/v5/trade/order' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -H 'OK-ACCESS-KEY: *****' \
  -H 'OK-ACCESS-SIGN: *****' \
  -H 'OK-ACCESS-TIMESTAMP: *****' \
  -H 'OK-ACCESS-PASSPHRASE: *****' \
  -H 'expTime: 1597026383085' \
  -d '{
  "instId": "BTC-USDT",
  "tdMode": "cash",
  "side": "buy",
  "ordType": "limit",
  "px": "1000",
  "sz": "0.01"
}'
```

### WEBSOCKET

The following parameters are set in the request

Parameter	Type	Required	Description
expTime	String	No	Request effective deadline. Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

The following endpoints are supported:

- Place order
- Place multiple orders
- Amend order
- Amend multiple orders

#### Request Example

```
{  
  "id": "1512",  
  "op": "order",  
  "expTime": "1597026383085", // request effective deadline  
  "args": [{"  
    "side": "buy",  
    "instId": "BTC-USDT",  
    "tdMode": "isolated",  
    "ordType": "market",  
    "sz": "100"  
  }]  
}
```

## Rate Limits

Our REST and WebSocket APIs use rate limits to protect our APIs against malicious usage so our trading platform can operate reliably and fairly. When a request is rejected by our system due to rate limits, the system returns error code 50011 (Rate limit reached. Please refer to API documentation and throttle requests accordingly).

The rate limit is different for each endpoint. You can find the limit for each endpoint from the endpoint details. Rate limit definitions are detailed below:

- WebSocket login and subscription rate limits are based on connection.
- Public unauthenticated REST rate limits are based on IP address.
- Private REST rate limits are based on User ID (sub-accounts have individual User IDs).
- WebSocket order management rate limits are based on User ID (sub-accounts have individual User IDs).

### TRADING-RELATED APIs

For Trading-related APIs (place order, cancel order, and amend order) the following conditions apply:

- Rate limits are shared across the REST and WebSocket channels.
- Rate limits for placing orders, amending orders, and cancelling orders are independent from each other.
- Rate limits are defined on the Instrument ID level (except Options)
- Rate limits for Options are defined based on the Instrument Family level. Refer to the Get instruments endpoint to view Instrument Family information.
- Rate limits for a multiple order endpoint and a single order endpoint are also independent, with the exception being when there is only one order sent to a multiple order endpoint, the order will be counted as a single order and adopt the single order rate limit.

### SUB-ACCOUNT RATE LIMIT

At the sub-account level, we allow a maximum of 1000 order requests per 2 seconds. Only new order requests and amendment order requests will be counted towards this limit. The limit encompasses all requests from the endpoints below. For batch order requests consisting of multiple orders, each order will be counted individually. Error code 50061 is returned when the sub-account rate limit is exceeded. The existing rate limit rule per instrument ID remains unchanged and the existing rate limit and sub-account rate limit will operate in parallel. If clients require a higher rate limit, clients can trade via multiple sub-accounts.

- POST / Place order
- POST / Place multiple orders
- POST / Amend order
- POST / Amend multiple orders

- WS / Place order
- WS / Place multiple orders
- WS / Amend order
- WS / Amend multiple orders

#### FILL RATIO BASED SUB-ACCOUNT RATE LIMIT

This is only applicable to >= VIP5 customers.

As an incentive for more efficient trading, the exchange will offer a higher sub-account rate limit to clients with a high trade fill ratio.

The exchange calculates two ratios based on the transaction data from the past 7 days at 00:00 UTC.

1. Sub-account fill ratio: This ratio is determined by dividing (the trade volume in USDT of the sub-account) by (sum of (new and amendment request count per symbol \* symbol multiplier) of the sub-account). Note that the master trading account itself is also considered as a sub-account in this context.
2. Master account aggregated fill ratio: This ratio is calculated by dividing (the trade volume in USDT on the master account level) by (the sum (new and amendment count per symbol \* symbol multiplier) of all sub-accounts).

The symbol multiplier allows for fine-tuning the weight of each symbol. A smaller symbol multiplier (<1) is used for smaller pairs that require more updates per trading volume. All instruments have a default symbol multiplier, and some instruments will have overridden symbol multipliers.

InstType	Override rule	Overridden symbol multiplier	Default symbol multiplier
Perpetual Futures	Per instrument ID	<div style="display: flex; align-items: center;"> <span>1</span> <div style="margin-left: 10px;">Instrument ID: BTC-USDT-SWAP BTC-USD-SWAP ETH-USDT-SWAP ETH-USD-SWAP</div> </div>	<div style="display: flex; align-items: center;"> <span>0.2</span> </div>
Expiry Futures	Per instrument Family	<div style="display: flex; align-items: center;"> <span>0.3</span> <div style="margin-left: 10px;">Instrument Family: BTC-USDT BTC-USD ETH-USDT ETH-USD</div> </div>	<div style="display: flex; align-items: center;"> <span>0.1</span> </div>
Spot	Per instrument ID	<div style="display: flex; align-items: center;"> <span>0.5</span> <div style="margin-left: 10px;">Instrument ID: BTC-USDT ETH-USDT</div> </div>	<div style="display: flex; align-items: center;"> <span>0.1</span> </div>
Options	Per instrument Family		<div style="display: flex; align-items: center;"> <span>0.1</span> </div>

The fill ratio computation excludes block trading, spread trading, MMP and fiat orders for order count; and excludes block trading, spread trading for trade volume. Only successful order requests (sCode=0) are considered.

At 08:00 UTC, the system will use the maximum value between the sub-account fill ratio and the master account aggregated fill ratio based on the data snapshot at 00:00 UTC to determine the sub-account rate limit based on the table below. For broker (non-disclosed) clients, the system considers the sub-account fill ratio only.

Fill ratio[x<=ratio<y)		Sub-account rate limit per 2 seconds(new and amendment)
Tier 1	[0,1)	1,000
Tier 2	[1,2)	1,250
Tier 3	[2,3)	1,500

Fill ratio[x <= ratio < y)		Sub-account rate limit per 2 seconds(new and amendment)
Tier 4	[3,5)	1,750
Tier 5	[5,10)	2,000
Tier 6	[10,20)	2,500
Tier 7	[20,50)	3,000
Tier 8	>= 50	10,000

If there is an improvement in the fill ratio and rate limit to be uplifted, the uplift will take effect immediately at 08:00 UTC. However, if the fill ratio decreases and the rate limit needs to be lowered, a one-day grace period will be granted, and the lowered rate limit will only be implemented on T+1 at 08:00 UTC. On T+1, if the fill ratio improves, the higher rate limit will be applied accordingly. In the event of client demotion to VIP4, their rate limit will be downgraded to Tier 1, accompanied by a one-day grace period.

If the 7-day trading volume of a sub-account is less than 1,000,000 USDT, the fill ratio of the master account will be applied to it.

For newly created sub-accounts, the Tier 1 rate limit will be applied at creation until T+1 8am UTC, at which the normal rules will be applied.

Block trading, spread trading, MMP and spot/margin orders are exempted from the sub-account rate limit.

The exchange offers GET / Account rate limit endpoint that provides ratio and rate limit data, which will be updated daily at 8am UTC. It will return the sub-account fill ratio, the master account aggregated fill ratio, current sub-account rate limit and sub-account rate limit on T+1 (applicable if the rate limit is going to be demoted).

The fill ratio and rate limit calculation example is shown below. Client has 3 accounts, symbol multiplier for BTC-USDT-SWAP = 1 and XRP-USDT = 0.1.

1. Account A (master account):

1. BTC-USDT-SWAP trade volume = 100 USDT, order count = 10;
2. XRP-USDT trade volume = 20 USDT, order count = 15;
3. Sub-account ratio =  $(100+20) / (10 * 1 + 15 * 0.1) = 10.4$

2. Account B (sub-account):

1. BTC-USDT-SWAP trade volume = 200 USDT, order count = 100;
2. XRP-USDT trade volume = 20 USDT, order count = 30;
3. Sub-account ratio =  $(200+20) / (100 * 1 + 30 * 0.1) = 2.13$

3. Account C (sub-account):

1. BTC-USDT-SWAP trade volume = 300 USDT, order count = 1000;
2. XRP-USDT trade volume = 20 USDT, order count = 45;
3. Sub-account ratio =  $(300+20) / (100 * 1 + 45 * 0.1) = 3.06$

4. Master account aggregated fill ratio =  $(100+20+200+20+300+20) / (10 * 1 + 15 * 0.1 + 100 * 1 + 30 * 0.1 + 100 * 1 + 45 * 0.1) = 3.01$

5. Rate limit of accounts

1. Account A =  $\max(10.4, 3.01) = 10.4 \rightarrow 2500 \text{ order requests/2s}$
2. Account B =  $\max(2.13, 3.01) = 3.01 \rightarrow 1750 \text{ order requests/2s}$
3. Account C =  $\max(3.06, 3.01) = 3.06 \rightarrow 1750 \text{ order requests/2s}$

## BEST PRACTICES

If you require a higher request rate than our rate limit, you can set up different sub-accounts to batch request rate limits. We recommend this method for throttling or spacing out requests in order to maximize each accounts' rate limit and avoid disconnections or rejections.

## Market Maker Program

High-caliber trading teams are welcomed to work with OKX as market makers in providing a liquid, fair, and orderly platform to all users. OKX market makers could enjoy favourable fees in return for meeting the market making obligations.

Prerequisites (Satisfy any condition):

- VIP 2 or above on fee schedule
- Qualified Market Maker on other exchange

Interested parties can reach out to us using this form: <https://okx.typeform.com/contact-sales>

Remarks:

Market making obligations and trading fees will be shared to successful parties only.

OKX reserves the right of final decision and interpretation for the content hereinabove.

In fairness to all users, market makers will be ineligible for other VIP-related and volume-related promotions or rebates.

## Broker Program

If your business platform offers cryptocurrency services, you can apply to join the OKX Broker Program, become our partner broker, enjoy exclusive broker services, and earn high rebates through trading fees generated by OKX users.

The Broker Program includes, and is not limited to, integrated trading platforms, trading bots, copy trading platforms, trading bot providers, quantitative strategy institutions, asset management platforms etc.

- Click to apply
- Broker rules
- If you have any questions, feel free to contact our customer support.

Relevant information for specific Broker Program documentation and product services will be provided following successful applications.

## Trading Account

The API endpoints of `Account` require authentication.

## REST API

### GET INSTRUMENTS

Retrieve available instruments info of current account.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID + INSTRUMENT TYPE**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/account/instruments`

Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"
```

```

flag = "1" # Production trading: 0, Demo trading: 1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

result = accountAPI.get_instruments(instType="SPOT")
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	Yes	<p>Instrument type</p> <p><b>SPOT</b>: Spot</p> <p><b>MARGIN</b>: Margin</p> <p><b>SWAP</b>: Perpetual Futures</p> <p><b>FUTURES</b>: Expiry Futures</p> <p><b>OPTION</b>: Option</p>
instFamily	String	Conditional	<p>Instrument family</p> <p>Only applicable to <b>FUTURES</b>/<b>SWAP</b>/<b>OPTION</b>. If <b>instType</b> is <b>OPTION</b>, <b>instFamily</b> is required.</p>
instId	String	No	Instrument ID

## Response Example

```

{
  "code": "0",
  "data": [
    {
      "auctionEndTime": "",
      "baseCcy": "BTC",
      "ctMult": "",
      "ctType": "",
      "ctVal": "",
      "ctValCcy": "",
      "contTdSwTime": "1704876947000",
      "expTime": "",
      "futureSettlement": false,
      "instFamily": "",
      "instId": "BTC-EUR",
      "instType": "SPOT",
      "lever": "",
      "listTime": "1704876947000",
      "lotSz": "0.0000001",
      "maxIcebergSz": "999999999.000000000000000",
      "maxLmtAmt": "1000000",
      "maxLmtSz": "999999999",
      "maxMktAmt": "1000000",
      "maxMktSz": "1000000",
      "maxPlatOI": "1000000000",
      "maxStopSz": "1000000",
      "maxTriggerSz": "999999999.000000000000000",
      "maxTwapSz": "999999999.000000000000000",
      "minSz": "0.0001",
      "optType": "",
      "openType": "call_auction",
      "preMktSwTime": "",
      "posLmtPct": "30",
      "posLmtAmt": "2500000",
      "quoteCcy": "EUR",
      "tradeQuoteCcyList": [
        "EUR"
      ],
      "settleCcy": "",
      "state": "live",
      "ruleType": "normal",
      "stk": "",
      "tickSz": "1",
      "uly": "",
      "instIdCode": 1000000000
    }
  ]
}

```

```
],  
"msg": ""  
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type
instId	String	Instrument ID, e.g. <code>BTC-USD-SWAP</code>
uly	String	Underlying, e.g. <code>BTC-USD</code> Only applicable to <code>MARGIN/FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>
instFamily	String	Instrument family, e.g. <code>BTC-USD</code> Only applicable to <code>MARGIN/FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>
baseCcy	String	Base currency, e.g. <code>BTC</code> in <code>BTC-USDT</code> Only applicable to <code>SPOT</code> / <code>MARGIN</code>
quoteCcy	String	Quote currency, e.g. <code>USDT</code> in <code>BTC-USDT</code> Only applicable to <code>SPOT</code> / <code>MARGIN</code>
settleCcy	String	Settlement and margin currency, e.g. <code>BTC</code> Only applicable to <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>
ctVal	String	Contract value Only applicable to <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>
ctMult	String	Contract multiplier Only applicable to <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>
ctValCcy	String	Contract value currency Only applicable to <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>
optType	String	Option type, <code>C</code> : Call <code>P</code> : put Only applicable to <code>OPTION</code>
stk	String	Strike price Only applicable to <code>OPTION</code>
listTime	String	Listing time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
auctionEndTime	String	The end time of call auction, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code> Only applicable to <code>SPOT</code> that are listed through call auctions, return "" in other cases (deprecated, use <code>contTdSwTime</code> )
contTdSwTime	String	Continuous trading switch time. The switch time from call auction, prequote to continuous trading, Unix timestamp format in milliseconds. e.g. <code>1597026383085</code> . Only applicable to <code>SPOT</code> / <code>MARGIN</code> that are listed through call auction or prequote, return "" in other cases.
preMktSwTime	String	The time premarket swap switched to normal swap, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code> . Only applicable premarket <code>SWAP</code>
openType	String	Open type <code>fix_price</code> : fix price opening <code>pre_quote</code> : pre-quote <code>call_auction</code> : call auction Only applicable to <code>SPOT</code> / <code>MARGIN</code> , return "" for all other business lines
expTime	String	Expiry time Applicable to <code>SPOT</code> / <code>MARGIN</code> / <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code> . For <code>FUTURES</code> / <code>OPTION</code> , it is natural delivery/exercise time. It is the instrument offline time when there is <code>SPOT/MARGIN/FUTURES/SWAP</code> manual offline. Update once change.

Parameter	Type	Description
lever	String	Max Leverage, Not applicable to <code>SPOT</code> , <code>OPTION</code>
tickSz	String	Tick size, e.g. <code>0.0001</code> For Option, it is minimum tickSz among tick band, please use "Get option tick bands" if you want get option tickBands.
lotSz	String	Lot size If it is a derivatives contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in <code>base currency</code> .
minSz	String	Minimum order size If it is a derivatives contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in <code>base currency</code> .
ctType	String	Contract type <code>linear</code> : linear contract <code>inverse</code> : inverse contract Only applicable to <code>FUTURES</code> / <code>SWAP</code>
state	String	Instrument status <code>live</code> <code>suspend</code> <code>preopen</code> e.g. Futures and options contracts rollover from generation to trading start; certain symbols before they go live <code>test</code> : Test pairs, can't be traded
ruleType	String	Trading rule types <code>normal</code> : normal trading <code>pre_market</code> : pre-market trading
posLmtAmt	String	Maximum position value (USD) for this instrument at the user level, based on the notional value of all same-direction open positions and resting orders. The effective user limit is $\max(\text{posLmtAmt}, \text{oiUSD} \times \text{posLmtPct})$ . Applicable to <code>SWAP</code> / <code>FUTURES</code> .
posLmtPct	String	Maximum position ratio (e.g., 30 for 30%) a user may hold relative to the platform's current total position value. The effective user limit is $\max(\text{posLmtAmt}, \text{oiUSD} \times \text{posLmtPct})$ . Applicable to <code>SWAP</code> / <code>FUTURES</code> .
maxPlatOILmt	String	Platform-wide maximum position value (USD) for this instrument. If the global position limit switch is enabled and platform total open interest reaches or exceeds this value, all users' new opening orders for this instrument are rejected; otherwise, orders pass.
maxLmtSz	String	The maximum order quantity of a single limit order. If it is a derivatives contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in <code>base currency</code> .
maxMktSz	String	The maximum order quantity of a single market order. If it is a derivatives contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in <code>USDT</code> .
maxLmtAmt	String	Max USD amount for a single limit order
maxMktAmt	String	Max USD amount for a single market order Only applicable to <code>SPOT</code> / <code>MARGIN</code>
maxTwapSz	String	The maximum order quantity of a single TWAP order. If it is a derivatives contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in <code>base currency</code> . The minimum order quantity of a single TWAP order is $\text{minSz} \times 2$
maxIcebergSz	String	The maximum order quantity of a single iceBerg order. If it is a derivatives contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in <code>base currency</code> .

Parameter	Type	Description
maxTriggerSz	String	The maximum order quantity of a single trigger order. If it is a derivatives contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in <code>base currency</code> .
maxStopSz	String	The maximum order quantity of a single stop market order. If it is a derivatives contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in <code>USDT</code> .
futureSettlement	Boolean	Whether daily settlement for expiry feature is enabled Applicable to <code>FUTURES</code> <code>cross</code>
tradeQuoteCcyList	Array of strings	List of quote currencies available for trading, e.g. <code>["USD", "USDC"]</code> .
instIdCode	Integer	Instrument ID code. For simple binary encoding, you must use <code>instIdCode</code> instead of <code>instId</code> . For the same <code>instId</code> , its value may be different between production and demo trading.

#### listTime and contTdSwTime

For spot symbols listed through a call auction or pre-open, listTime represents the start time of the auction or pre-open, and contTdSwTime indicates the end of the auction or pre-open and the start of continuous trading. For other scenarios, listTime will mark the beginning of continuous trading, and contTdSwTime will return an empty value "".

#### state

The state will always change from `'preopen'` to `'live'` when the listTime is reached.

When a product is going to be delisted (e.g. when a `FUTURES` contract is settled or `OPTION` contract is exercised), the instrument will not be available.

## GET BALANCE

Retrieve a list of assets (with non-zero balance), remaining balance, and available amount in the trading account.

Interest-free quota and discount rates are public data and not displayed on the account interface.

### RATE LIMIT: 10 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: READ

### HTTP REQUEST

`GET /api/v5/account/balance`

#### Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Get account balance
```

```
result = accountAPI.get_account_balance()  
print(result)
```

## REQUEST PARAMETERS

Parameters	Types	Required	Description
ccy	String	No	Single currency or multiple currencies (no more than 20) separated with comma, e.g. <code>BTC</code> or <code>BTC,ETH</code> .

## Response Example

```
{  
  "code": "0",  
  "data": [  
    {  
      "adjEq": "55415.624719833286",  
      "availEq": "55415.624719833286",  
      "borrowFroz": "0",  
      "delta": "0",  
      "deltaLever": "0",  
      "deltaNeutralStatus": "0",  
      "details": [  
        {  
          "autoLendStatus": "off",  
          "autoLendMtAmt": "0",  
          "availBal": "4834.317093622894",  
          "availEq": "4834.317093622895",  
          "borrowFroz": "0",  
          "cashBal": "4850.435693622894",  
          "ccy": "USDT",  
          "crossLiab": "0",  
          "colRes": "0",  
          "collateralEnabled": false,  
          "collateralRestrict": false,  
          "colBorrAutoConversion": "0",  
          "disEq": "4991.542013297616",  
          "eq": "4992.890093622894",  
          "eqUsd": "4991.542013297616",  
          "smtSyncEq": "0",  
          "spotCopyTradingEq": "0",  
          "fixedBal": "0",  
          "frozenBal": "158.573",  
          "frpType": "0",  
          "imr": "",  
          "interest": "0",  
          "isoEq": "0",  
          "isolab": "0",  
          "isoUp": "0",  
          "liab": "0",  
          "maxLoan": "0",  
          "mgnRatio": "",  
          "mmr": "",  
          "notionalLever": "",  
          "ordFrozen": "0",  
          "rewardBal": "0",  
          "spotInUseAmt": "",  
          "clsSpotInUseAmt": "",  
          "maxSpotInUse": "",  
          "spotIsoBal": "0",  
          "stgyEq": "150",  
          "twap": "0",  
          "uTime": "1705449605015",  
          "upl": "-7.545600000000006",  
          "uplLiab": "0",  
          "spotBal": "",  
          "openAvgPx": "",  
          "accAvgPx": "",  
          "spotUp": "",  
          "spotUpRatio": "",  
          "totalPnl": "",  
          "totalPnlRatio": ""  
        }  
      ],  
    }  
  ]  
},
```

```

        "imr": "0",
        "isoEq": "0",
        "mgnRatio": "",
        "mmr": "0",
        "notionalUsd": "0",
        "notionalUsdForBorrow": "0",
        "notionalUsdForFutures": "0",
        "notionalUsdForOption": "0",
        "notionalUsdForSwap": "0",
        "ordFroz": "",
        "totalEq": "55837.43556134779",
        "uTime": "1705474164160",
        "upl": "0",
    }
],
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameters	Types	Description
uTime	String	Update time of account information, millisecond format of Unix timestamp, e.g. <code>1597026383085</code>
totalEq	String	The total amount of equity in <code>USD</code>
isoEq	String	Isolated margin equity in <code>USD</code> Applicable to <code>Futures mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
adjEq	String	Adjusted / Effective equity in <code>USD</code> The net fiat value of the assets in the account that can provide margins for spot, expiry futures, perpetual futures and options under the cross-margin mode. In multi-ccy or PM mode, the asset and margin requirement will all be converted to USD value to process the order check or liquidation. Due to the volatility of each currency market, our platform calculates the actual USD value of each currency based on discount rates to balance market risks. Applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> and <code>Portfolio margin</code>
availEq	String	Account level available equity, excluding currencies that are restricted due to the collateralized borrowing limit. Applicable to <code>Multi-currency margin</code> / <code>Portfolio margin</code>
ordFroz	String	Cross margin frozen for pending orders in <code>USD</code> Only applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
imr	String	Initial margin requirement in <code>USD</code> The sum of initial margins of all open positions and pending orders under cross-margin mode in <code>USD</code> . Applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
mmr	String	Maintenance margin requirement in <code>USD</code> The sum of maintenance margins of all open positions and pending orders under cross-margin mode in <code>USD</code> . Applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
borrowFroz	String	Potential borrowing IMR of the account in <code>USD</code> Only applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code> . It is "" for other margin modes.
mgnRatio	String	Maintenance margin ratio in <code>USD</code> Applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
notionalUsd	String	Notional value of positions in <code>USD</code> Applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
notionalUsdForBorrow	String	Notional value for <code>Borrow</code> in USD Applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>

Parameters	Types	Description
notionalUsdForSwap	String	Notional value of positions for <b>Perpetual Futures</b> in USD Applicable to <b>Multi-currency margin</b> / <b>Portfolio margin</b>
notionalUsdForFutures	String	Notional value of positions for <b>Expiry Futures</b> in USD Applicable to <b>Multi-currency margin</b> / <b>Portfolio margin</b>
notionalUsdForOption	String	Notional value of positions for <b>Option</b> in USD Applicable to <b>Spot mode</b> / <b>Multi-currency margin</b> / <b>Portfolio margin</b>
upl	String	Cross-margin info of unrealized profit and loss at the account level in <b>USD</b> Applicable to <b>Multi-currency margin</b> / <b>Portfolio margin</b>
delta	String	Delta (USD)
deltaLever	String	Delta neutral strategy account level delta leverage $\text{deltaLever} = \text{delta} / \text{totalEq}$
deltaNeutralStatus	String	Delta risk status 0: normal 1: transfer restricted 2: delta reducing - cancel all pending orders if delta is greater than 5000 USD, only one delta reducing order allowed per index (spot, futures, swap)
details	Array of objects	Detailed asset information in all currencies
> ccy	String	Currency
> eq	String	Equity of currency
> cashBal	String	Cash balance
> uTime	String	Update time of currency balance information, Unix timestamp format in milliseconds, e.g. 1597026383085
> isoEq	String	Isolated margin equity of currency Applicable to <b>Futures mode</b> / <b>Multi-currency margin</b> / <b>Portfolio margin</b>
> availEq	String	Available equity of currency Applicable to <b>Futures mode</b> / <b>Multi-currency margin</b> / <b>Portfolio margin</b>
> disEq	String	Discount equity of currency in <b>USD</b> . Applicable to <b>Spot mode</b> (enabled spot borrow)/ <b>Multi-currency margin</b> / <b>Portfolio margin</b>
> fixedBal	String	Frozen balance for <b>Dip Sniper</b> and <b>Peak Sniper</b>
> availBal	String	Available balance of currency
> frozenBal	String	Frozen balance of currency
> ordFrozen	String	Margin frozen for open orders Applicable to <b>Spot mode</b> / <b>Futures mode</b> / <b>Multi-currency margin</b>
> liab	String	Liabilities of currency It is a positive value, e.g. 21625.64 Applicable to <b>Spot mode</b> / <b>Multi-currency margin</b> / <b>Portfolio margin</b>
> upl	String	The sum of the unrealized profit & loss of all margin and derivatives positions of currency. Applicable to <b>Futures mode</b> / <b>Multi-currency margin</b> / <b>Portfolio margin</b>
> uplLiab	String	Liabilities due to Unrealized loss of currency Applicable to <b>Multi-currency margin</b> / <b>Portfolio margin</b>

Parameters	Types	Description
> crossLiab	String	Cross liabilities of currency Applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
> rewardBal	String	Trial fund balance
> isoLiab	String	Isolated liabilities of currency Applicable to <code>Multi-currency margin</code> / <code>Portfolio margin</code>
> mgnRatio	String	Cross maintenance margin ratio of currency The index for measuring the risk of a certain asset in the account. Applicable to <code>Futures mode</code> and when there is cross position
> imr	String	Cross initial margin requirement at the currency level Applicable to <code>Futures mode</code> and when there is cross position
> mmr	String	Cross maintenance margin requirement at the currency level Applicable to <code>Futures mode</code> and when there is cross position
> interest	String	Accrued interest of currency It is a positive value, e.g. <code>9.01</code> Applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
> twap	String	Risk indicator of forced repayment Divided into multiple levels from 0 to 5, the larger the number, the more likely the forced repayment will be triggered. Applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
> frpType	String	Forced repayment (FRP) type <code>0</code> : no FRP <code>1</code> : user based FRP <code>2</code> : platform based FRP  Return <code>1</code> / <code>2</code> when <code>twap</code> is $\geq 1$ , applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
> maxLoan	String	Max loan of currency Applicable to <code>cross</code> of <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
> eqUsd	String	Equity in <code>USD</code> of currency
> borrowFroz	String	Potential borrowing IMR of currency in <code>USD</code> Applicable to <code>Multi-currency margin</code> / <code>Portfolio margin</code> . It is "" for other margin modes.
> notionalLever	String	Leverage of currency Applicable to <code>Futures mode</code>
> stgyEq	String	Strategy equity
> isoUpL	String	Isolated unrealized profit and loss of currency Applicable to <code>Futures mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
> spotInUseAmt	String	Spot in use amount Applicable to <code>Portfolio margin</code>
> clSpotInUseAmt	String	User-defined spot risk offset amount Applicable to <code>Portfolio margin</code>
> maxSpotInUse	String	Max possible spot risk offset amount Applicable to <code>Portfolio margin</code>
> spotIsoBal	String	Spot isolated balance Applicable to copy trading Applicable to <code>Spot mode</code> / <code>Futures mode</code>

Parameters	Types	Description
> smtSyncEq	String	Smart sync equity The default is "0", only applicable to copy trader.
> spotCopyTradingEq	String	Spot smart sync equity. The default is "0", only applicable to copy trader.
> spotBal	String	Spot balance. The unit is currency, e.g. BTC. More details
> openAvgPx	String	Spot average cost price. The unit is USD. More details
> accAvgPx	String	Spot accumulated cost price. The unit is USD. More details
> spotUpL	String	Spot unrealized profit and loss. The unit is USD. More details
> spotUpLRatio	String	Spot unrealized profit and loss ratio. More details
> totalPnl	String	Spot accumulated profit and loss. The unit is USD. More details
> totalPnlRatio	String	Spot accumulated profit and loss ratio. More details
> colRes	String	Platform level collateral restriction status 0: The restriction is not enabled. 1: The restriction is not enabled. But the crypto is close to the platform's collateral limit. 2: The restriction is enabled. This crypto can't be used as margin for your new orders. This may result in failed orders. But it will still be included in the account's adjusted equity and doesn't impact margin ratio. Refer to <a href="#">Introduction to the platform collateralized borrowing limit</a> for more details.
> colBorrAutoConversion	String	Risk indicator of auto conversion. Divided into multiple levels from 1-5, the larger the number, the more likely the repayment will be triggered. The default will be 0, indicating there is no risk currently. 5 means this user is undergoing auto conversion now, 4 means this user will undergo auto conversion soon whereas 1/2/3 indicates there is a risk for auto conversion. Applicable to <a href="#">Spot mode</a> / <a href="#">Futures mode</a> / <a href="#">Multi-currency margin</a> / <a href="#">Portfolio margin</a> When the total liability for each crypto set as collateral exceeds a certain percentage of the platform's total limit, the auto-conversion mechanism may be triggered. This may result in the automatic sale of excess collateral crypto if you've set this crypto as collateral and have large borrowings. To lower this risk, consider reducing your use of the crypto as collateral or reducing your liabilities. Refer to <a href="#">Introduction to the platform collateralized borrowing limit</a> for more details.
> collateralRestrict	Boolean	Platform level collateralized borrow restriction <a href="#">true</a> <a href="#">false</a> (deprecated, use colRes instead)
> collateralEnabled	Boolean	<a href="#">true</a> : Collateral enabled <a href="#">false</a> : Collateral disabled Applicable to <a href="#">Multi-currency margin</a>
> autoLendStatus	String	Auto lend status <a href="#">unsupported</a> : auto lend is not supported by this currency <a href="#">off</a> : auto lend is supported but turned off <a href="#">pending</a> : auto lend is turned on but pending matching <a href="#">active</a> : auto lend is turned on and matched
> autoLendMtAmt	String	Auto lend currency matched amount Return "0" when autoLendStatus is <a href="#">unsupported/off/pending</a> . Return matched amount when autoLendStatus is <a href="#">active</a>

- Regarding more parameter details, you can refer to product documentations below:  
 Futures mode: cross margin trading  
 Multi-currency margin mode: cross margin trading  
 Multi-currency margin mode vs. Portfolio margin mode

"" will be returned for inapplicable fields under the current account level.

The currency details will not be returned when cashBal and eq is both 0.

Distribution of applicable fields under each account level are as follows:

Parameters	Spot mode	Futures mode	Multi-currency margin mode	Portfolio margin mode
uTime	Yes	Yes	Yes	Yes
totalEq	Yes	Yes	Yes	Yes
isoEq		Yes	Yes	Yes
adjEq	Yes		Yes	Yes
availEq			Yes	Yes
ordFroz	Yes		Yes	Yes
imr	Yes		Yes	Yes
mmr	Yes		Yes	Yes
borrowFroz	Yes		Yes	Yes
mgnRatio	Yes		Yes	Yes
notionalUsd	Yes		Yes	Yes
notionalUsdForSwap			Yes	Yes
notionalUsdForFutures			Yes	Yes
notionalUsdForOption	Yes		Yes	Yes
notionalUsdForBorrow	Yes		Yes	Yes
upl			Yes	Yes
details			Yes	Yes
> ccy	Yes	Yes	Yes	Yes
> eq	Yes	Yes	Yes	Yes
> cashBal	Yes	Yes	Yes	Yes
> uTime	Yes	Yes	Yes	Yes
> isoEq		Yes	Yes	Yes
> availEq		Yes	Yes	Yes
> disEq	Yes		Yes	Yes
> availBal	Yes	Yes	Yes	Yes
> frozenBal	Yes	Yes	Yes	Yes
> ordFrozen	Yes	Yes	Yes	Yes
> liab	Yes		Yes	Yes
> upl		Yes	Yes	Yes

Parameters	Spot mode	Futures mode	Multi-currency margin mode	Portfolio margin mode
> uplLiab		Yes		Yes
> crossLiab	Yes		Yes	
> isoLiab		Yes		Yes
> mgnRatio		Yes		
> interest	Yes		Yes	Yes
> twap	Yes		Yes	Yes
> maxLoan	Yes		Yes	Yes
> eqUsd	Yes	Yes	Yes	Yes
> borrowFroz	Yes		Yes	Yes
> notionalLever		Yes		
> stgyEq	Yes	Yes	Yes	Yes
> isoUpl		Yes	Yes	Yes
> spotInUseAmt				Yes
> spotIsoBal	Yes	Yes		
> imr		Yes		
> mmr		Yes		
> smtSyncEq	Yes	Yes	Yes	Yes
> spotCopyTradingEq	Yes	Yes	Yes	Yes
> spotBal	Yes	Yes	Yes	Yes
> openAvgPx	Yes	Yes	Yes	Yes
> accAvgPx	Yes	Yes	Yes	Yes
> spotUpI	Yes	Yes	Yes	Yes
> spotUpIRatio	Yes	Yes	Yes	Yes
> totalPnl	Yes	Yes	Yes	Yes
> totalPnlRatio c	Yes	Yes	Yes	Yes
> collateralEnabled		Yes		

## GET POSITIONS

Retrieve information on your positions. When the account is in `net` mode, `net` positions will be displayed, and when the account is in `long/short` mode, `long` or `short` positions will be displayed. Return in reverse chronological order using `ctime`.

**RATE LIMIT: 10 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

## Request Example

```

import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Get positions information
result = accountAPI.get_positions()
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	<p>Instrument type</p> <div style="display: flex; justify-content: space-around;"> <span>MARGIN</span> <span>SWAP</span> <span>FUTURES</span> <span>OPTION</span> </div> <p><code>instId</code> will be checked against <code>instType</code> when both parameters are passed.</p>
instId	String	No	<p>Instrument ID, e.g. <code>BTC-USDT-SWAP</code>. Single instrument ID or multiple instrument IDs (no more than 10) separated with comma</p>
posId	String	No	<p>Single position ID or multiple position IDs (no more than 20) separated with comma.</p> <p>There is attribute expiration, the posId and position information will be cleared if it is more than 30 days after the last full close position.</p>

## instId

If the instrument ever had position and its open interest is 0, it will return the position information with specific instId. It will not return the position information with specific instId if there is no valid posId; it will not return the position information without specific instId.

In the isolated margin trading settings, if it is set to the manual transfers mode, after the position is transferred to the margin, a position with a position of 0 will be generated

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "adl": "1",
      "availPos": "0.00190433573",
      "avgPx": "62961.4",
      "baseBal": "",
      "baseBorrowed": "",
      "baseInterest": "",
      "bePx": "",
      "bizRefId": "",
      "bizRefType": "",
      "cTime": "1724740225685",
      "ccy": "BTC",
      "clSpotInUseAmt": "",
      "closeOrderAlgo": [],
      "deltaBS": "",
      "deltaPA": ""
    }
  ]
}
```

```

"fee": "",
"fundingFee": "",
"gammaBS": "",
"gammaPA": "",
"hedgedPos": "",
"idxPx": "62890.5",
"imr": "",
"instId": "BTC-USDT",
"instType": "MARGIN",
"interest": "0",
"last": "62892.9",
"lever": "5",
"liab": "-99.999817776581948",
"liabCcy": "USDT",
"liqPenalty": "",
"liqPx": "53615.448336593756",
"margin": "0.000317654",
"markPx": "62891.9",
"maxSpotInUseAmt": "",
"mgnMode": "isolated",
"mgnRatio": "9.404143929947395",
"mmr": "0.000318005395854",
"notionalUsd": "119.756628017499",
"optVal": "",
"pendingCloseOrdLiabVal": "0",
"pnl": "",
"pos": "0.00190433573",
"posCcy": "BTC",
"posId": "1752810569801498626",
"posSide": "net",
"quoteBal": "",
"quoteBorrowed": "",
"quoteInterest": "",
"realizedPnl": "",
"spotInUseAmt": "",
"spotInUseCcy": "",
"thetaBS": "",
"thetaPA": "",
"tradeId": "785524470",
"uTime": "1724742632153",
"upl": "-0.000033452492717",
"uplLastPx": "-0.000033199677697",
"uplRatio": "-0.0105311101755551",
"uplRatioLastPx": "-0.0104515220008934",
"usdPx": "",
"vegaBS": "",
"vegaPA": "",
"nonSettleAvgPx": "",
"settledPnl": ""
}
],
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type
mgnMode	String	Margin mode <div style="display: flex; justify-content: space-around; align-items: center;"> <span><input type="checkbox"/></span> <span><input checked="" type="checkbox"/></span> </div>
posId	String	Position ID
posSide	String	Position side <div style="display: flex; justify-content: space-around; align-items: center;"> <span><input type="checkbox"/></span> <span><input type="checkbox"/></span> </div> <span><input type="checkbox"/></span> is positive <span><input type="checkbox"/></span> is positive <span><input type="checkbox"/></span> ( <span><input type="checkbox"/></span> / <span><input type="checkbox"/></span> / <span><input type="checkbox"/></span> ): positive <span><input type="checkbox"/></span> means long position and negative <span><input type="checkbox"/></span> means short position. For <span><input type="checkbox"/></span> , <span><input type="checkbox"/></span> is always positive, <span><input type="checkbox"/></span> being base currency means long position, <span><input type="checkbox"/></span> being quote currency means short position.)

Parameter	Type	Description
pos	String	Quantity of positions. In the isolated margin mode, when doing manual transfers, a position with pos of <code>0</code> will be generated after the deposit is transferred
hedgedPos	String	Hedged position size Only return for accounts in delta neutral strategy, stgyType:1. Return "" for accounts in general strategy.
baseBal	String	Base currency balance, only applicable to <code>MARGIN</code> (Quick Margin Mode) (Deprecated)
quoteBal	String	Quote currency balance, only applicable to <code>MARGIN</code> (Quick Margin Mode) (Deprecated)
baseBorrowed	String	Base currency amount already borrowed, only applicable to <code>MARGIN</code> (Quick Margin Mode) (Deprecated)
baseInterest	String	Base Interest, undeducted interest that has been incurred, only applicable to <code>MARGIN</code> (Quick Margin Mode) (Deprecated)
quoteBorrowed	String	Quote currency amount already borrowed, only applicable to <code>MARGIN</code> (Quick Margin Mode) (Deprecated)
quoteInterest	String	Quote Interest, undeducted interest that has been incurred, only applicable to <code>MARGIN</code> (Quick Margin Mode) (Deprecated)
posCcy	String	Position currency, only applicable to <code>MARGIN</code> positions.
availPos	String	Position that can be closed Only applicable to <code>MARGIN</code> and <code>OPTION</code> . For <code>MARGIN</code> position, the rest of sz will be <code>SPOT</code> trading after the liability is repaid while closing the position. Please get the available reduce-only amount from "Get maximum available tradable amount" if you want to reduce the amount of <code>SPOT</code> trading as much as possible.
avgPx	String	Average open price Under cross-margin mode, the entry price of expiry futures will update at settlement to the last settlement price, and when the position is opened or increased.
nonSettleAvgPx	String	Non-settlement entry price The non-settlement entry price only reflects the average price at which the position is opened or increased. Applicable to <code>cross</code> <code>FUTURES</code> positions.
markPx	String	Latest Mark price
upl	String	Unrealized profit and loss calculated by mark price.
uplRatio	String	Unrealized profit and loss ratio calculated by mark price.
uplLastPx	String	Unrealized profit and loss calculated by last price. Main usage is showing, actual value is upl.
uplRatioLastPx	String	Unrealized profit and loss ratio calculated by last price.
instId	String	Instrument ID, e.g. <code>BTC-USDT-SWAP</code>
lever	String	Leverage Not applicable to <code>OPTION</code> and positions of cross margin mode under <code>Portfolio margin</code>
liqPx	String	Estimated liquidation price Not applicable to <code>OPTION</code>
imr	String	Initial margin requirement, only applicable to <code>cross</code> .
margin	String	Margin, can be added or reduced. Only applicable to <code>isolated</code> .
mgnRatio	String	Maintenance margin ratio
mmr	String	Maintenance margin requirement

Parameter	Type	Description
liab	String	Liabilities, only applicable to <code>MARGIN</code> .
liabCcy	String	Liabilities currency, only applicable to <code>MARGIN</code> .
interest	String	Interest. Undeducted interest that has been incurred.
tradeld	String	Last trade ID
optVal	String	Option Value, only applicable to <code>OPTION</code> .
pendingCloseOrdLiabVal	String	The amount of close orders of isolated margin liability.
notionalUsd	String	Notional value of positions in <code>USD</code>
adl	String	Auto-deleveraging (ADL) indicator Divided into 6 levels, from 0 to 5, the smaller the number, the weaker the adl intensity. Only applicable to <code>FUTURES/SWAP/OPTION</code>
ccy	String	Currency used for margin
last	String	Latest traded price
idxPx	String	Latest underlying index price
usdPx	String	Latest USD price of the <code>ccy</code> on the market, only applicable to <code>FUTURES/SWAP/OPTION</code>
bePx	String	Breakeven price
deltaBS	String	delta: Black-Scholes Greeks in dollars, only applicable to <code>OPTION</code>
deltaPA	String	delta: Greeks in coins, only applicable to <code>OPTION</code>
gammaBS	String	gamma: Black-Scholes Greeks in dollars, only applicable to <code>OPTION</code>
gammaPA	String	gamma: Greeks in coins, only applicable to <code>OPTION</code>
thetaBS	String	theta: Black-Scholes Greeks in dollars, only applicable to <code>OPTION</code>
thetaPA	String	theta: Greeks in coins, only applicable to <code>OPTION</code>
vegaBS	String	vega: Black-Scholes Greeks in dollars, only applicable to <code>OPTION</code>
vegaPA	String	vega: Greeks in coins, only applicable to <code>OPTION</code>
spotInUseAmt	String	Spot in use amount Applicable to <code>Portfolio margin</code>
spotInUseCcy	String	Spot in use unit, e.g. <code>BTC</code> Applicable to <code>Portfolio margin</code>
clSpotInUseAmt	String	User-defined spot risk offset amount Applicable to <code>Portfolio margin</code>
maxSpotInUseAmt	String	Max possible spot risk offset amount Applicable to <code>Portfolio margin</code>
bizRefId	String	External business id, e.g. experience coupon id
bizRefType	String	External business type
realizedPnl	String	Realized profit and loss Only applicable to <code>FUTURES/SWAP/OPTION</code>

Parameter	Type	Description
		$\text{realizedPnl} = \text{pnl} + \text{fee} + \text{fundingFee} + \text{liqPenalty} + \text{settledPnl}$
settledPnl	String	Accumulated settled profit and loss (calculated by settlement price) Only applicable to <code>cross</code> <code>FUTURES</code>
pnl	String	Accumulated pnl of closing order(s) (excluding the fee).
fee	String	Accumulated fee Negative number represents the user transaction fee charged by the platform. Positive number represents rebate.
fundingFee	String	Accumulated funding fee
liqPenalty	String	Accumulated liquidation penalty. It is negative when there is a value.
closeOrderAlgo	Array of objects	Close position algo orders attached to the position. This array will have values only after you request "Place algo order" with <code>closeFraction</code> =1.
> algold	String	Algo ID
> slTriggerPx	String	Stop-loss trigger price.
> slTriggerPxType	String	Stop-loss trigger price type. <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price
> tpTriggerPx	String	Take-profit trigger price.
> tpTriggerPxType	String	Take-profit trigger price type. <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price
> closeFraction	String	Fraction of position to be closed when the algo order is triggered.
cTime	String	Creation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
uTime	String	Latest time position was adjusted, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

As for portfolio margin account, the IMR and MMR of the position are calculated in risk unit granularity, thus their values of the same risk unit cross positions are the same.

## GET POSITIONS HISTORY

Retrieve the updated position data for the last 3 months. Return in reverse chronological order using utime. Getting positions history is supported under Portfolio margin mode since **04:00 AM (UTC) on November 11, 2024**.

## RATE LIMIT: 10 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: READ

### HTTP REQUEST

```
GET /api/v5/account/positions-history
```

Request Example

```

import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Get positions history
result = accountAPI.get_positions_history()
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type <input type="button" value="MARGIN"/> <input type="button" value="SWAP"/> <input type="button" value="FUTURES"/> <input type="button" value="OPTION"/>
instId	String	No	Instrument ID, e.g. <input type="button" value="BTC-USD-SWAP"/>
mgnMode	String	No	Margin mode <input type="button" value="cross"/> <input type="button" value="isolated"/>
type	String	No	The type of latest close position 1: Close position partially; 2: Close all; 3: Liquidation; 4: Partial liquidation; 5: ADL - position not fully closed; 6: ADL - position fully closed It is the latest type if there are several types for the same position.
posId	String	No	Position ID. There is attribute expiration. The posId will be expired if it is more than 30 days after the last full close position, then position will use new posId.
after	String	No	Pagination of data to return records earlier than the requested <input format="" in="" milliseconds,<br="" timestamp="" type="button" unix="" value="uTime" },=""/> e.g. <input type="button" value="1597026383085"/>
before	String	No	Pagination of data to return records newer than the requested <input format="" in="" milliseconds,<br="" timestamp="" type="button" unix="" value="uTime" },=""/> e.g. <input type="button" value="1597026383085"/>
limit	String	No	Number of results per request. The maximum is 100. The default is 100. All records that have the same <input type="button" value="uTime"/> will be returned at the current request

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "cTime": "1654177169995",
      "ccy": "BTC",
      "closeAvgPx": "29786.599999789081085",
      "closeTotalPos": "1",
      "instId": "BTC-USD-SWAP",
      "instType": "SWAP",
      "lever": "10.0",
      "mgnMode": "cross",
      "openAvgPx": "29783.89999999535393",
      "openMaxPos": "1",
      "realizedPnl": "0.001",
      "fee": "-0.0001",
      "fundingFee": "0",
      "liqPenalty": "0",
      "pnl": "0.0011",
      "pnlRatio": "0.000906447858888"
    }
  ]
}
```

```

    "posId": "452587086133239818",
    "posSide": "long",
    "direction": "long",
    "triggerPx": "",
    "type": "1",
    "uTime": "1654177174419",
    "uly": "BTC-USD",
    "nonSettleAvgPx": "",
    "settledPnl": ""
  }
],
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type
instId	String	Instrument ID
mgnMode	String	Margin mode cross isolated
type	String	The type of latest close position ①: Close position partially;②: Close all;③: Liquidation;④: Partial liquidation; ⑤: ADL; It is the latest type if there are several types for the same position.
cTime	String	Created time of position
uTime	String	Updated time of position
openAvgPx	String	Average price of opening position Under cross-margin mode, the entry price of expiry futures will update at settlement to the last settlement price, and when the position is opened or increased.
nonSettleAvgPx	String	Non-settlement entry price The non-settlement entry price only reflects the average price at which the position is opened or increased. Only applicable to cross FUTURES
closeAvgPx	String	Average price of closing position
posId	String	Position ID
openMaxPos	String	Max quantity of position
closeTotalPos	String	Position's cumulative closed volume
realizedPnl	String	Realized profit and loss Only applicable to FUTURES / SWAP / OPTION $\text{realizedPnl} = \text{pnl} + \text{fee} + \text{fundingFee} + \text{liqPenalty} + \text{settledPnl}$
settledPnl	String	Accumulated settled profit and loss (calculated by settlement price) Only applicable to cross FUTURES
pnlRatio	String	Realized P&L ratio
fee	String	Accumulated fee Negative number represents the user transaction fee charged by the platform. Positive number represents rebate.
fundingFee	String	Accumulated funding fee
liqPenalty	String	Accumulated liquidation penalty. It is negative when there is a value.
pnl	String	Profit and loss (excluding the fee).

Parameter	Type	Description
posSide	String	Position mode side long: Hedge mode long short: Hedge mode short net: Net mode
lever	String	Leverage
direction	String	Direction: long short Only applicable to MARGIN/FUTURES/SWAP/OPTION
triggerPx	String	trigger mark price. There is value when type is equal to 3, 4 or 5. It is "" when type is equal to 1 or 2
uly	String	Underlying
ccy	String	Currency used for margin

## GET ACCOUNT AND POSITION RISK

Get account and position risk

Obtain basic information about accounts and positions on the same time snapshot

**RATE LIMIT: 10 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

GET /api/v5/account/account-position-risk

Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Get account and position risk
result = accountAPI.get_account_position_risk()
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type MARGIN SWAP FUTURES OPTION

Response Example

```
{
  "code": "0",
  "data": [
```

```

{
  "adjEq": "174238.6793649711331679",
  "balData": [
    {
      "ccy": "BTC",
      "disEq": "78846.7803721021362242",
      "eq": "1.3863533369419636"
    },
    {
      "ccy": "USDT",
      "disEq": "73417.2495112863300127",
      "eq": "73323.395564963177146"
    }
  ],
  "posData": [
    {
      "baseBal": "0.4",
      "ccy": "",
      "instId": "BTC-USDT",
      "instType": "MARGIN",
      "mgnMode": "isolated",
      "notionalCcy": "0",
      "notionalUsd": "0",
      "pos": "0",
      "posCcy": "",
      "posId": "310388685292318723",
      "posSide": "net",
      "quoteBal": "0"
    }
  ],
  "ts": "1620282889345"
},
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameters	Types	Description
ts	String	Update time of account information, millisecond format of Unix timestamp, e.g. 1597026383085
adjEq	String	Adjusted / Effective equity in <a href="#">USD</a> Applicable to <a href="#">Multi-currency margin</a> and <a href="#">Portfolio margin</a>
balData	Array of objects	Detailed asset information in all currencies
> ccy	String	Currency
> eq	String	Equity of currency
> disEq	String	Discount equity of currency in <a href="#">USD</a> .
posData	Array of objects	Detailed position information in all currencies
> instType	String	Instrument type
> mgnMode	String	Margin mode <a href="#">cross</a> <a href="#">isolated</a>
> posId	String	Position ID
> instId	String	Instrument ID, e.g. <a href="#">BTC-USDT-SWAP</a>
> pos	String	Quantity of positions <a href="#">contract</a> . In the isolated margin mode, when doing manual transfers, a position with pos of <a href="#">0</a> will be generated after the deposit is transferred

Parameters	Types	Description
> baseBal	String	Base currency balance, only applicable to <code>MARGIN</code> (Quick Margin Mode) (Deprecated)
> quoteBal	String	Quote currency balance, only applicable to <code>MARGIN</code> (Quick Margin Mode) (Deprecated)
> posSide	String	Position side <code>long</code> <code>short</code> <code>net</code> ( <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code> ): positive <code>pos</code> means long position and negative <code>pos</code> means short position. <code>MARGIN</code> : <code>pos</code> <code>ccy</code> being base currency means long position, <code>pos</code> <code>ccy</code> being quote currency means short position.)
> posCcy	String	Position currency, only applicable to <code>MARGIN</code> positions.
> ccy	String	Currency used for margin
> notionalCcy	String	Notional value of positions in <code>coin</code>
> notionalUsd	String	Notional value of positions in <code>USD</code>

#### GET BILLS DETAILS (LAST 7 DAYS)

Retrieve the bills of the account. The bill refers to all transaction records that result in changing the balance of an account. Pagination is supported, and the response is sorted with the most recent first. This endpoint can retrieve data from the last 7 days.

**RATE LIMIT: 5 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/account/bills`

Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Get bills details (last 7 days)
result = accountAPI.get_account_bills()
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type <code>SPOT</code> <code>MARGIN</code> <code>SWAP</code> <code>FUTURES</code> <code>OPTION</code>
instId	String	No	Instrument ID, e.g. <code>BTC-USDT</code>
ccy	String	No	Bill currency

Parameter	Type	Required	Description
mgnMode	String	No	<p>Margin mode</p> <p><input type="checkbox"/> <code>isolated</code></p> <p><input type="checkbox"/> <code>cross</code></p>
ctType	String	No	<p>Contract type</p> <p><input type="checkbox"/> <code>linear</code></p> <p><input type="checkbox"/> <code>inverse</code></p> <p>Only applicable to <code>FUTURES</code> / <code>SWAP</code></p>
type	String	No	<p>Bill type</p> <p><input type="checkbox"/> <code>1</code>: Transfer</p> <p><input type="checkbox"/> <code>2</code>: Trade</p> <p><input type="checkbox"/> <code>3</code>: Delivery</p> <p><input type="checkbox"/> <code>4</code>: Forced repayment</p> <p><input type="checkbox"/> <code>5</code>: Liquidation</p> <p><input type="checkbox"/> <code>6</code>: Margin transfer</p> <p><input type="checkbox"/> <code>7</code>: Interest deduction</p> <p><input type="checkbox"/> <code>8</code>: Funding fee</p> <p><input type="checkbox"/> <code>9</code>: ADL</p> <p><input type="checkbox"/> <code>10</code>: Clawback</p> <p><input type="checkbox"/> <code>11</code>: System token conversion</p> <p><input type="checkbox"/> <code>12</code>: Strategy transfer</p> <p><input type="checkbox"/> <code>13</code>: DDH</p> <p><input type="checkbox"/> <code>14</code>: Block trade</p> <p><input type="checkbox"/> <code>15</code>: Quick Margin</p> <p><input type="checkbox"/> <code>16</code>: Borrowing</p> <p><input type="checkbox"/> <code>22</code>: Repay</p> <p><input type="checkbox"/> <code>24</code>: Spread trading</p> <p><input type="checkbox"/> <code>26</code>: Structured products</p> <p><input type="checkbox"/> <code>27</code>: Convert</p> <p><input type="checkbox"/> <code>28</code>: Easy convert</p> <p><input type="checkbox"/> <code>29</code>: One-click repay</p> <p><input type="checkbox"/> <code>30</code>: Simple trade</p> <p><input type="checkbox"/> <code>32</code>: Move position</p> <p><input type="checkbox"/> <code>33</code>: Loans</p> <p><input type="checkbox"/> <code>34</code>: Settlement</p> <p><input type="checkbox"/> <code>250</code>: Copy trader profit sharing expenses</p> <p><input type="checkbox"/> <code>251</code>: Copy trader profit sharing refund</p>
subType	String	No	<p>Bill subtype</p> <p><input type="checkbox"/> <code>1</code>: Buy</p> <p><input type="checkbox"/> <code>2</code>: Sell</p> <p><input type="checkbox"/> <code>3</code>: Open long</p> <p><input type="checkbox"/> <code>4</code>: Open short</p> <p><input type="checkbox"/> <code>5</code>: Close long</p> <p><input type="checkbox"/> <code>6</code>: Close short</p> <p><input type="checkbox"/> <code>9</code>: Interest deduction for Market loans</p> <p><input type="checkbox"/> <code>11</code>: Transfer in</p> <p><input type="checkbox"/> <code>12</code>: Transfer out</p> <p><input type="checkbox"/> <code>14</code>: Interest deduction for VIP loans</p> <p><input type="checkbox"/> <code>160</code>: Manual margin increase</p> <p><input type="checkbox"/> <code>161</code>: Manual margin decrease</p> <p><input type="checkbox"/> <code>162</code>: Auto margin increase</p> <p><input type="checkbox"/> <code>114</code>: Forced repayment buy</p> <p><input type="checkbox"/> <code>115</code>: Forced repayment sell</p> <p><input type="checkbox"/> <code>118</code>: System token conversion transfer in</p> <p><input type="checkbox"/> <code>119</code>: System token conversion transfer out</p> <p><input type="checkbox"/> <code>100</code>: Partial liquidation close long</p> <p><input type="checkbox"/> <code>101</code>: Partial liquidation close short</p> <p><input type="checkbox"/> <code>102</code>: Partial liquidation buy</p> <p><input type="checkbox"/> <code>103</code>: Partial liquidation sell</p> <p><input type="checkbox"/> <code>104</code>: Liquidation long</p>

Parameter	Type	Required	Description
			105: Liquidation short
			106: Liquidation buy
			107: Liquidation sell
			108: Clawback
			110: Liquidation transfer in
			111: Liquidation transfer out
			125: ADL close long
			126: ADL close short
			127: ADL buy
			128: ADL sell
			131: ddh buy
			132: ddh sell
			170: Exercised(ITM buy side)
			171: Counterparty exercised(ITM sell side)
			172: Expired(Non-ITM buy and sell side)
			112: Delivery long (applicable to <b>FUTURES</b> expiration and <b>SWAP</b> delisting)
			113: Delivery short (applicable to <b>FUTURES</b> expiration and <b>SWAP</b> delisting)
			117: Delivery/Exercise clawback
			173: Funding fee expense
			174: Funding fee income
			200: System transfer in
			201: Manually transfer in
			202: System transfer out
			203: Manually transfer out
			204: block trade buy
			205: block trade sell
			206: block trade open long
			207: block trade open short
			208: block trade close long
			209: block trade close short
			210: Manual Borrowing of quick margin
			211: Manual Repayment of quick margin
			212: Auto borrow of quick margin
			213: Auto repay of quick margin
			220: Transfer in when using USDT to buy OPTION
			221: Transfer out when using USDT to buy OPTION
			16: Repay forcibly
			17: Repay interest by borrowing forcibly
			224: Repayment transfer in
			225: Repayment transfer out
			236: Easy convert in
			237: Easy convert out
			250: Profit sharing expenses
			251: Profit sharing refund
			280: SPOT profit sharing expenses
			281: SPOT profit sharing refund
			282: Spot profit share income
			283: Asset transfer for spot copy trading
			270: Spread trading buy
			271: Spread trading sell
			272: Spread trading open long
			273: Spread trading open short
			274: Spread trading close long
			275: Spread trading close short
			280: SPOT profit sharing expenses
			281: SPOT profit sharing refund
			284: Copy trade automatic transfer in
			285: Copy trade manual transfer in
			286: Copy trade automatic transfer out
			287: Copy trade manual transfer out
			290: Crypto dust auto-transfer out
			293: Fixed loan interest deduction

Parameter	Type	Required	Description
			<p>294: Fixed loan interest refund</p> <p>295: Fixed loan overdue penalty</p> <p>296: From structured order placements</p> <p>297: To structured order placements</p> <p>298: From structured settlements</p> <p>299: To structured settlements</p> <p>306: Manual borrow</p> <p>307: Auto borrow</p> <p>308: Manual repay</p> <p>309: Auto repay</p> <p>312: Auto offset</p> <p>318: Convert in</p> <p>319: Convert out</p> <p>320: Simple buy</p> <p>321: Simple sell</p> <p>324: Move position buy</p> <p>325: Move position sell</p> <p>326: Move position open long</p> <p>327: Move position open short</p> <p>328: Move position close long</p> <p>329: Move position close short</p> <p>332: Margin transfer in isolated margin position</p> <p>333: Margin transfer out isolated margin position</p> <p>334: Margin loss when closing isolated margin position</p> <p>355: Settlement PnL</p> <p>376: Collateralized borrowing auto conversion buy</p> <p>377: Collateralized borrowing auto conversion sell</p> <p>381: Auto lend interest transfer in</p> <p>372: Bot airdrop (transfer in)</p> <p>373: Bot airdrop (transfer out)</p> <p>374: Bot airdrop reclaim (transfer in)</p> <p>375: Bot airdrop reclaim (transfer out)</p> <p>381: Auto earn (auto lend)</p>
after	String	No	Pagination of data to return records earlier than the requested bill ID.
before	String	No	Pagination of data to return records newer than the requested bill ID.
begin	String	No	Filter with a begin timestamp <code>ts</code> . Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
end	String	No	Filter with an end timestamp <code>ts</code> . Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. The maximum is <code>100</code> . The default is <code>100</code> .

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "bal": "8694.2179403378290202",
      "balChg": "0.0219338232210000",
      "billId": "623950854533513219",
      "ccy": "USDT",
      "clOrdId": "",
      "earnAmt": "",
      "earnApr": "",
      "execType": "T",
      "fee": "-0.000021955779",
      "fillFwdPx": "",
      "fillIdxPx": "27104.1",
      "fillMarkPx": "",
      "fillMarkVol": "",
      "fillPxUsd": "",
      "fillPxVol": ""
    }
  ]
}
```

```

"fillTime": "1695033476166",
"from": "",
"instId": "BTC-USDT",
"instType": "SPOT",
"interest": "0",
"mgnMode": "isolated",
"notes": "",
"ordId": "623950854525124608",
"pnl": "0",
"posBal": "0",
"posBalChg": "0",
"px": "27105.9",
"subType": "1",
"sz": "0.021955779",
"tag": "",
"to": "",
"tradeId": "586760148",
"ts": "1695033476167",
"type": "2"
}
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type
billId	String	Bill ID
type	String	Bill type
subType	String	Bill subtype
ts	String	The time when the balance complete update, Unix timestamp format in milliseconds, e.g. 1597026383085
balChg	String	Change in balance amount at the account level
posBalChg	String	Change in balance amount at the position level
bal	String	Balance at the account level
posBal	String	Balance at the position level
sz	String	Quantity For FUTURES/SWAP/OPTION, it is fill quantity or position quantity, the unit is contract. The value is always positive. For other scenarios, the unit is account balance currency(ccy).
px	String	Price which related to subType <ul style="list-style-type: none"> <li>Trade filled price for <ul style="list-style-type: none"> <li>Buy (1): Buy (2): Sell (3): Open long (4): Open short (5): Close long (6): Close short (204): block trade buy (205): block trade sell (206): block trade open long (207): block trade open short (208): block trade close long (209): block trade close short (114): Forced repayment buy (115): Forced repayment sell</li> <li>Liquidation Price for <ul style="list-style-type: none"> <li>Partial liquidation close long (101): Partial liquidation close short (102): Partial liquidation buy (103): Partial liquidation sell (104): Liquidation long (105): Liquidation short (106): Liquidation buy (107): Liquidation sell (16): Repay forcibly (17): Repay interest by borrowing forcibly (110): Liquidation transfer in (111): Liquidation transfer out</li> </ul> </li> <li>Delivery price for <ul style="list-style-type: none"> <li>Delivery long (112): Delivery short (113)</li> </ul> </li> <li>Exercise price for <ul style="list-style-type: none"> <li>Exercised (170): Counterparty exercised (171): Expired OTM</li> </ul> </li> <li>Mark price for <ul style="list-style-type: none"> <li>Funding fee expense (173): Funding fee income (174)</li> </ul> </li> </ul> </li> </ul>
ccy	String	Account balance currency
pnl	String	Profit and loss

Parameter	Type	Description
fee	String	<p>Fee</p> <p>Negative number represents the user transaction fee charged by the platform.</p> <p>Positive number represents rebate.</p> <p>Trading fee rule</p>
earnAmt	String	<p>Auto earn amount</p> <p>Only applicable when type is 381</p>
earnApr	String	<p>Auto earn APR</p> <p>Only applicable when type is 381</p>
mgnMode	String	<p>Margin mode</p> <p><code>isolated</code> <code>cross</code> <code>cash</code></p> <p>When bills are not generated by trading, the field returns ""</p>
instId	String	Instrument ID, e.g. <code>BTC-USDT</code>
ordId	String	<p>Order ID</p> <p>Return order ID when the type is <code>2</code>/<code>5</code>/<code>9</code></p> <p>Return "" when there is no order.</p>
execType	String	<p>Liquidity taker or maker</p> <p><code>T</code>: taker</p> <p><code>M</code>: maker</p>
from	String	<p>The remitting account</p> <p><code>6</code>: Funding account</p> <p><code>18</code>: Trading account</p> <p>Only applicable to <code>transfer</code>. When bill type is not <code>transfer</code>, the field returns "".</p>
to	String	<p>The beneficiary account</p> <p><code>6</code>: Funding account</p> <p><code>18</code>: Trading account</p> <p>Only applicable to <code>transfer</code>. When bill type is not <code>transfer</code>, the field returns "".</p>
notes	String	Notes
interest	String	Interest
tag	String	Order tag
fillTime	String	Last filled time
tradeId	String	Last traded ID
clOrdId	String	<p>Client Order ID as assigned by the client</p> <p>A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.</p>
fillIdxPx	String	<p>Index price at the moment of trade execution</p> <p>For cross currency spot pairs, it returns baseCcy-USDT index price. For example, for LTC-ETH, this field returns the index price of LTC-USDT.</p>
fillMarkPx	String	<p>Mark price when filled</p> <p>Applicable to FUTURES/SWAP/OPTIONS, return "" for other instrument types</p>
fillPxVol	String	<p>Implied volatility when filled</p> <p>Only applicable to options; return "" for other instrument types</p>
fillPxUsd	String	<p>Options price when filled, in the unit of USD</p> <p>Only applicable to options; return "" for other instrument types</p>
fillMarkVol	String	<p>Mark volatility when filled</p> <p>Only applicable to options; return "" for other instrument types</p>

Parameter	Type	Description
fillFwdPx	String	Forward price when filled Only applicable to options; return "" for other instrument types

### Funding Fee expense (subType = 173)

You may refer to "pnl" for the fee payment

#### GET BILLS DETAILS (LAST 3 MONTHS)

Retrieve the account's bills. The bill refers to all transaction records that result in changing the balance of an account. Pagination is supported, and the response is sorted with most recent first. This endpoint can retrieve data from the last 3 months.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

GET /api/v5/account/bills-archive

#### Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Get bills details (last 3 months)
result = accountAPI.get_account_bills_archive()
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	<p>Instrument type</p> <div style="display: flex; justify-content: space-around;"> <span>SPOT</span> <span>MARGIN</span> <span>SWAP</span> <span>FUTURES</span> <span>OPTION</span> </div>
instId	String	No	Instrument ID, e.g. <code>BTC-USDT</code>
ccy	String	No	Bill currency
mgnMode	String	No	<p>Margin mode</p> <div style="display: flex; justify-content: space-around;"> <span>isolated</span> <span>cross</span> </div>
ctType	String	No	<p>Contract type</p> <div style="display: flex; justify-content: space-around;"> <span>linear</span> <span>inverse</span> </div> <p>Only applicable to <code>FUTURES</code>/<code>SWAP</code></p>
type	String	No	<p>Bill type</p> <div style="display: flex; justify-content: space-around;"> <span>1: Transfer</span> </div>

Parameter	Type	Required	Description
			<p>[2]: Trade</p> <p>[3]: Delivery</p> <p>[4]: Forced repayment</p> <p>[5]: Liquidation</p> <p>[6]: Margin transfer</p> <p>[7]: Interest deduction</p> <p>[8]: Funding fee</p> <p>[9]: ADL</p> <p>[10]: Clawback</p> <p>[11]: System token conversion</p> <p>[12]: Strategy transfer</p> <p>[13]: DDH</p> <p>[14]: Block trade</p> <p>[15]: Quick Margin</p> <p>[16]: Borrowing</p> <p>[22]: Repay</p> <p>[24]: Spread trading</p> <p>[26]: Structured products</p> <p>[27]: Convert</p> <p>[28]: Easy convert</p> <p>[29]: One-click repay</p> <p>[30]: Simple trade</p> <p>[32]: Move position</p> <p>[33]: Loans</p> <p>[34]: Settlement</p> <p>[250]: Copy trader profit sharing expenses</p> <p>[251]: Copy trader profit sharing refund</p>
subType	String	No	<p>Bill subtype</p> <p>[1]: Buy</p> <p>[2]: Sell</p> <p>[3]: Open long</p> <p>[4]: Open short</p> <p>[5]: Close long</p> <p>[6]: Close short</p> <p>[9]: Interest deduction for Market loans</p> <p>[11]: Transfer in</p> <p>[12]: Transfer out</p> <p>[14]: Interest deduction for VIP loans</p> <p>[160]: Manual margin increase</p> <p>[161]: Manual margin decrease</p> <p>[162]: Auto margin increase</p> <p>[114]: Forced repayment buy</p> <p>[115]: Forced repayment sell</p> <p>[118]: System token conversion transfer in</p> <p>[119]: System token conversion transfer out</p> <p>[100]: Partial liquidation close long</p> <p>[101]: Partial liquidation close short</p> <p>[102]: Partial liquidation buy</p> <p>[103]: Partial liquidation sell</p> <p>[104]: Liquidation long</p> <p>[105]: Liquidation short</p> <p>[106]: Liquidation buy</p> <p>[107]: Liquidation sell</p> <p>[108]: Clawback</p> <p>[110]: Liquidation transfer in</p> <p>[111]: Liquidation transfer out</p> <p>[125]: ADL close long</p> <p>[126]: ADL close short</p> <p>[127]: ADL buy</p> <p>[128]: ADL sell</p> <p>[131]: ddh buy</p>

Parameter	Type	Required	Description
			132: ddh sell
			170: Exercised(ITM buy side)
			171: Counterparty exercised(ITM sell side)
			172: Expired(Non-ITM buy and sell side)
			112: Delivery long (applicable to <b>FUTURES</b> expiration and <b>SWAP</b> delisting)
			113: Delivery short (applicable to <b>FUTURES</b> expiration and <b>SWAP</b> delisting)
			117: Delivery/Exercise clawback
			173: Funding fee expense
			174: Funding fee income
			200: System transfer in
			201: Manually transfer in
			202: System transfer out
			203: Manually transfer out
			204: block trade buy
			205: block trade sell
			206: block trade open long
			207: block trade open short
			208: block trade close long
			209: block trade close short
			210: Manual Borrowing of quick margin
			211: Manual Repayment of quick margin
			212: Auto borrow of quick margin
			213: Auto repay of quick margin
			220: Transfer in when using USDT to buy OPTION
			221: Transfer out when using USDT to buy OPTION
			16: Repay forcibly
			17: Repay interest by borrowing forcibly
			224: Repayment transfer in
			225: Repayment transfer out
			236: Easy convert in
			237: Easy convert out
			250: Profit sharing expenses
			251: Profit sharing refund
			280: SPOT profit sharing expenses
			281: SPOT profit sharing refund
			282: Spot profit share income
			283: Asset transfer for spot copy trading
			270: Spread trading buy
			271: Spread trading sell
			272: Spread trading open long
			273: Spread trading open short
			274: Spread trading close long
			275: Spread trading close short
			280: SPOT profit sharing expenses
			281: SPOT profit sharing refund
			284: Copy trade automatic transfer in
			285: Copy trade manual transfer in
			286: Copy trade automatic transfer out
			287: Copy trade manual transfer out
			290: Crypto dust auto-transfer out
			293: <b>Fixed loan interest deduction</b>
			294: <b>Fixed loan interest refund</b>
			295: <b>Fixed loan overdue penalty</b>
			296: From structured order placements
			297: To structured order placements
			298: From structured settlements
			299: To structured settlements
			306: Manual borrow
			307: Auto borrow
			308: Manual repay
			309: Auto repay
			312: Auto offset

Parameter	Type	Required	Description
			318: Convert in 319: Convert out 320: Simple buy 321: Simple sell 324: Move position buy 325: Move position sell 326: Move position open long 327: Move position open short 328: Move position close long 329: Move position close short 332: Margin transfer in isolated margin position 333: Margin transfer out isolated margin position 334: Margin loss when closing isolated margin position 355: Settlement PnL 376: Collateralized borrowing auto conversion buy 377: Collateralized borrowing auto conversion sell 381: Auto lend interest transfer in 372: Bot airdrop (transfer in) 373: Bot airdrop (transfer out) 374: Bot airdrop reclaim (transfer in) 375: Bot airdrop reclaim (transfer out) 381: Auto earn (auto lend)
after	String	No	Pagination of data to return records earlier than the requested bill ID.
before	String	No	Pagination of data to return records newer than the requested bill ID.
begin	String	No	Filter with a begin timestamp <code>ts</code> . Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
end	String	No	Filter with an end timestamp <code>ts</code> . Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. The maximum is <code>100</code> . The default is <code>100</code> .

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "bal": "8694.2179403378290202",
      "balChg": "0.0219338232210000",
      "billId": "623950854533513219",
      "ccy": "USDT",
      "clOrdId": "",
      "earnAmt": "",
      "earnApr": "",
      "execType": "T",
      "fee": "-0.000021955779",
      "fillFwdPx": "",
      "fillIdxPx": "27104.1",
      "fillMarkPx": "",
      "fillMarkVol": "",
      "fillPxUsd": "",
      "fillPxVol": "",
      "fillTime": "1695033476166",
      "from": "",
      "instId": "BTC-USDT",
      "instType": "SPOT",
      "interest": "0",
      "mgnMode": "isolated",
      "notes": "",
      "ordId": "623950854525124608",
      "pnl": "0",
      "posBal": "0",
      "posBalChg": "0",
      "px": "27105.9",
      "subType": "1"
    }
  ]
}
```

```

    "sz": "0.021955779",
    "tag": "",
    "to": "",
    "tradeId": "586760148",
    "ts": "1695033476167",
    "type": "2"
  }
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type
billId	String	Bill ID
type	String	Bill type
subType	String	Bill subtype
ts	String	The time when the balance complete update, Unix timestamp format in milliseconds, e.g. 1597026383085
balChg	String	Change in balance amount at the account level
posBalChg	String	Change in balance amount at the position level
bal	String	Balance at the account level
posBal	String	Balance at the position level
sz	String	Quantity For FUTURES / SWAP / OPTION, it is fill quantity or position quantity, the unit is contract. The value is always positive. For other scenarios, the unit is account balance currency (ccy).
px	String	Price which related to subType <ul style="list-style-type: none"> <li>Trade filled price for           <ul style="list-style-type: none"> <li>1: Buy 2: Sell 3: Open long 4: Open short 5: Close long 6: Close short 204: block trade buy 205: block trade sell 206: block trade open long 207: block trade open short 208: block trade close long 209: block trade close short 114: Forced repayment buy 115: Forced repayment sell</li> </ul> </li> <li>Liquidation Price for           <ul style="list-style-type: none"> <li>100: Partial liquidation close long 101: Partial liquidation close short 102: Partial liquidation buy 103: Partial liquidation sell 104: Liquidation long 105: Liquidation short 106: Liquidation buy 107: Liquidation sell 16: Repay forcibly 17: Repay interest by borrowing forcibly 110: Liquidation transfer in 111: Liquidation transfer out</li> </ul> </li> <li>Delivery price for           <ul style="list-style-type: none"> <li>112: Delivery long 113: Delivery short</li> </ul> </li> <li>Exercise price for           <ul style="list-style-type: none"> <li>170: Exercised 171: Counterparty exercised 172: Expired OTM</li> </ul> </li> <li>Mark price for           <ul style="list-style-type: none"> <li>173: Funding fee expense 174: Funding fee income</li> </ul> </li> </ul>
ccy	String	Account balance currency
pnl	String	Profit and loss
fee	String	Fee Negative number represents the user transaction fee charged by the platform. Positive number represents rebate. Trading fee rule
earnAmt	String	Auto earn amount Only applicable when type is 381
earnApr	String	Auto earn APR Only applicable when type is 381

Parameter	Type	Description
mgnMode	String	Margin mode Margin mode isolate cross cash When bills are not generated by trading, the field returns ""
instId	String	Instrument ID, e.g. <code>BTC-USDT</code>
ordId	String	Order ID Return order ID when the type is <code>2/5/9</code> Return "" when there is no order.
execType	String	Liquidity taker or maker <code>T</code> : taker <code>M</code> : maker
from	String	The remitting account <code>6</code> : Funding account <code>18</code> : Trading account Only applicable to <code>transfer</code> . When bill type is not <code>transfer</code> , the field returns "".
to	String	The beneficiary account <code>6</code> : Funding account <code>18</code> : Trading account Only applicable to <code>transfer</code> . When bill type is not <code>transfer</code> , the field returns "".
notes	String	Notes
interest	String	Interest
tag	String	Order tag
fillTime	String	Last filled time
tradId	String	Last traded ID
clOrdId	String	Client Order ID as assigned by the client A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
fillIdxPx	String	Index price at the moment of trade execution For cross currency spot pairs, it returns baseCcy-USDT index price. For example, for LTC-ETH, this field returns the index price of LTC-USDT.
fillMarkPx	String	Mark price when filled Applicable to FUTURES/SWAP/OPTIONS, return "" for other instrument types
fillPxVol	String	Implied volatility when filled Only applicable to options; return "" for other instrument types
fillPxUsd	String	Options price when filled, in the unit of USD Only applicable to options; return "" for other instrument types
fillMarkVol	String	Mark volatility when filled Only applicable to options; return "" for other instrument types
fillFwdPx	String	Forward price when filled Only applicable to options; return "" for other instrument types

#### Funding Fee expense (subType = 173)

You may refer to "pnl" for the fee payment

#### APPLY BILLS DETAILS (SINCE 2021)

Apply for bill data since 1 February, 2021 except for the current quarter.

RATE LIMIT: 12 REQUESTS PER DAY

RATE LIMIT RULE: USER ID

PERMISSION: READ

HTTP REQUEST

POST /api/v5/account/bills-history-archive

Request Example

REQUEST PARAMETERS

Parameter	Type	Required	Description
year	String	Yes	4 digits year
quarter	String	Yes	Quarter, valid value is <code>Q1</code> , <code>Q2</code> , <code>Q3</code> , <code>Q4</code>

Response Example

```
{
  "code": "0",
  "data": [
    {
      "result": "true",
      "ts": "1646892328000"
    }
  ],
  "msg": ""
}
```

RESPONSE PARAMETERS

Parameter	Type	Description
result	String	Whether there is already a download link for this section <code>true</code> : Existed, can check from "Get bills details (since 2021)". <code>false</code> : Does not exist and is generating, can check the download link after 2 hours The data of file is in reverse chronological order using <code>billId</code> .
ts	String	The first request time when the server receives. Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

The rule introduction, only applicable to the file generated after 11 October, 2024

1. Taking 2024 Q2 as an example. The date range are [2024-07-01, 2024-10-01). The begin date is included, The end date is excluded.
2. The data of file is in reverse chronological order using 'billId'

Check the file link from the "Get bills details (since 2021)" endpoint in 2 hours to allow for data generation.

During peak demand, data generation may take longer. If the file link is still unavailable after 3 hours, reach out to customer support for assistance.

It is only applicable to the data from the unified account.

GET BILLS DETAILS (SINCE 2021)

Apply for bill data since 1 February, 2021 except for the current quarter.

RATE LIMIT: 10 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: READ

## HTTP REQUEST

GET /api/v5/account/bills-history-archive

Response Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
year	String	Yes	4 digits year
quarter	String	Yes	Quarter, valid value is <a href="#">Q1</a> , <a href="#">Q2</a> , <a href="#">Q3</a> , <a href="#">Q4</a>

Response Example

```
{  
  "code": "0",  
  "data": [  
    {  
      "fileHref": "http://xxx",  
      "state": "finished",  
      "ts": "1646892328000"  
    }  
  ],  
  "msg": ""  
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
fileHref	String	Download file link. The expiration of every link is 5 and a half hours. If you already apply the files for the same quarter, then it don't need to apply again within 30 days.
ts	String	The first request time when the server receives. Unix timestamp format in milliseconds, e.g. <a href="#">1597026383085</a>
state	String	Download link status "finished" "ongoing" "failed": Failed, please apply again

It is only applicable to the data from the unified account.

### FIELD DESCRIPTIONS IN THE DECOMPRESSED CSV FILE

Parameter	Type	Description
instType	String	Instrument type
billId	String	Bill ID
subType	String	Bill subtype
ts	String	The time when the balance complete update, Unix timestamp format in milliseconds, e.g. <a href="#">1597026383085</a>
balChg	String	Change in balance amount at the account level
posBalChg	String	Change in balance amount at the position level
bal	String	Balance at the account level
posBal	String	Balance at the position level
sz	String	Quantity

Parameter	Type	Description
px	String	<p>Price which related to subType</p> <ul style="list-style-type: none"> <li>• Trade filled price for <ul style="list-style-type: none"> <li>①: Buy ②: Sell ③: Open long ④: Open short ⑤: Close long ⑥: Close short ②④: block trade buy ②⑤: block trade sell ②⑥: block trade open long ②⑦: block trade open short ②⑧: block trade close long ②⑨: block trade close short ①④: Forced repayment buy ①⑤: Forced repayment sell</li> </ul> </li> <li>• Liquidation Price for <ul style="list-style-type: none"> <li>⑩①: Partial liquidation close long ⑩①: Partial liquidation close short ⑩②: Partial liquidation buy ⑩③: Partial liquidation sell ⑩④: Liquidation long ⑩⑤: Liquidation short ⑩⑥: Liquidation buy ⑩⑦: Liquidation sell ⑩⑧: Repay forcibly ⑩⑨: Repay interest by borrowing forcibly ⑩⑩: Liquidation transfer in ⑩⑪: Liquidation transfer out</li> </ul> </li> <li>• Delivery price for <ul style="list-style-type: none"> <li>⑪②: Delivery long ⑪③: Delivery short</li> </ul> </li> <li>• Exercise price for <ul style="list-style-type: none"> <li>⑯①: Exercised ⑯②: Counterparty exercised ⑯③: Expired OTM</li> </ul> </li> <li>• Mark price for <ul style="list-style-type: none"> <li>⑯③: Funding fee expense ⑯④: Funding fee income</li> </ul> </li> </ul>
ccy	String	Account balance currency
pnl	String	Profit and loss
fee	String	<p>Fee</p> <p>Negative number represents the user transaction fee charged by the platform.</p> <p>Positive number represents rebate.</p> <p>Trading fee rule</p>
mgnMode	String	<p>Margin mode</p> <p>④⑤⑥: isolated ④⑦⑧: cross ④⑨⑩: cash</p> <p>When bills are not generated by trading, the field returns ""</p>
instId	String	Instrument ID, e.g. <code>BTC-USDT</code>
ordId	String	<p>Order ID</p> <p>Return order ID when the type is ②/⑤/⑨</p> <p>Return "" when there is no order.</p>
execType	String	<p>Liquidity taker or maker</p> <p>⑤: taker ⑥: maker</p>
interest	String	Interest
tag	String	Order tag
fillTime	String	Last filled time
tradId	String	Last traded ID
clOrdId	String	<p>Client Order ID as assigned by the client</p> <p>A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.</p>
fillIdxPx	String	<p>Index price at the moment of trade execution</p> <p>For cross currency spot pairs, it returns baseCcy-USDT index price. For example, for LTC-ETH, this field returns the index price of LTC-USDT.</p>
fillMarkPx	String	<p>Mark price when filled</p> <p>Applicable to FUTURES/SWAP/OPTIONS, return "" for other instrument types</p>
fillPxVol	String	<p>Implied volatility when filled</p> <p>Only applicable to options; return "" for other instrument types</p>
fillPxUsd	String	<p>Options price when filled, in the unit of USD</p> <p>Only applicable to options; return "" for other instrument types</p>

Parameter	Type	Description
fillMarkVol	String	Mark volatility when filled Only applicable to options; return "" for other instrument types
fillFwdPx	String	Forward price when filled Only applicable to options; return "" for other instrument types

## GET ACCOUNT CONFIGURATION

Retrieve current account configuration.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

GET /api/v5/account/config

Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Retrieve current account configuration
result = accountAPI.get_account_config()
print(result)
```

## REQUEST PARAMETERS

none

Response Example

```
{
  "code": "0",
  "data": [
    {
      "acctLv": "2",
      "acctStpMode": "cancel_maker",
      "autoLoan": false,
      "ctIsoMode": "automatic",
      "enableSpotBorrow": false,
      "greeksType": "PA",
      "feeType": "0",
      "ip": "",
      "type": "0",
      "kycLv": "3",
      "label": "v5 test",
      "level": "Lv1",
      "levelTmp": "",
      "liquidationGear": "-1",
      "mainUid": "44705892343619584",
      "mgnIsoMode": "automatic",
      "opAuth": "1",
      "perm": "read_only,withdraw,trade",
      "posMode": "long_short_mode",
      "roleType": "0",
      "spotBorrowAutoRepay": false,
      "spotOffsetType": "",
      "spotRoleType": "0",
      "spotTraderInsts": []
    }
  ]
}
```

```

        "stgyType": "0",
        "traderInsts": [],
        "uid": "44705892343619584",
        "settleCcy": "USDC",
        "settleCcyList": ["USD", "USDC", "USDG"]
    }
],
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
uid	String	Account ID of current request.
mainUid	String	Main Account ID of current request. The current request account is main account if uid = mainUid. The current request account is sub-account if uid != mainUid.
acctLv	String	Account mode 1: Spot mode 2: Futures mode 3: Multi-currency margin 4: Portfolio margin
acctStpMode	String	Account self-trade prevention mode cancel_maker cancel_taker cancel_both The default value is cancel_maker. Users can log in to the webpage through the master account to modify this configuration
posMode	String	Position mode long_short_mode: long/short, only applicable to FUTURES/SWAP net_mode: net
autoLoan	Boolean	Whether to borrow coins automatically true: borrow coins automatically false: not borrow coins automatically
greeksType	String	Current display type of Greeks PA: Greeks in coins BS: Black-Scholes Greeks in dollars
feeType	String	Fee type 0: fee is charged in the currency you receive from the trade 1: fee is always charged in the quote currency of the trading pair
level	String	The user level of the current real trading volume on the platform, e.g Lv1, which means regular user level.
levelTmp	String	Temporary experience user level of special users, e.g Lv1
ctlIsoMode	String	Contract isolated margin trading settings automatic: Auto transfers autonomy: Manual transfers
mgnIsoMode	String	Margin isolated margin trading settings auto_transfers_ccy: New auto transfers, enabling both base and quote currency as the margin for isolated margin trading automatic: Auto transfers quick_margin: Quick Margin Mode (For new accounts, including subaccounts, some defaults will be automatic, and others will be quick_margin)
spotOffsetType	String	Risk offset type 1: Spot-Derivatives(USDT) to be offsetted

Parameter	Type	Description
		<p>2: Spot Derivatives(Coin) to be offsetted</p> <p>3: Only derivatives to be offsetted</p> <p>Only applicable to <code>Portfolio margin</code></p> <p>(Deprecated)</p>
stgyType	String	<p>Strategy type</p> <p>0: general strategy</p> <p>1: delta neutral strategy</p>
roleType	String	<p>Role type</p> <p>0: General user</p> <p>1: Leading trader</p> <p>2: Copy trader</p>
traderInsts	Array of strings	Leading trade instruments, only applicable to Leading trader
spotRoleType	String	SPOT copy trading role type. 0: General user; 1: Leading trader; 2: Copy trader
spotTraderInsts	Array of strings	Spot lead trading instruments, only applicable to lead trader
opAuth	String	<p>Whether the optional trading was activated</p> <p>0: not activate</p> <p>1: activated</p>
kycLv	String	<p>Main account KYC level</p> <p>0: No verification</p> <p>1: level 1 completed</p> <p>2: level 2 completed</p> <p>3: level 3 completed</p> <p>If the request originates from a subaccount, kycLv is the KYC level of the main account. If the request originates from the main account, kycLv is the KYC level of the current account.</p>
label	String	API key note of current request API key. No more than 50 letters (case sensitive) or numbers, which can be pure letters or pure numbers.
ip	String	IP addresses that linked with current API key, separate with commas if more than one, e.g. 117.37.203.58, 17.37.203.57. It is an empty string "" if there is no IP bonded.
perm	String	<p>The permission of the current requesting API key or Access token</p> <p>read_only: Read</p> <p>trade: Trade</p> <p>withdraw: Withdraw</p>
liquidationGear	String	<p>The maintenance margin ratio level of liquidation alert</p> <p>3 and -1 means that you will get hourly liquidation alerts on app and channel "Position risk warning" when your margin level drops to or below 300%. -1 is the initial value which has the same effect as -3</p> <p>0 means that there is not alert</p>
enableSpotBorrow	Boolean	<p>Whether borrow is allowed or not in <code>Spot mode</code></p> <p>true: Enabled</p> <p>false: Disabled</p>
spotBorrowAutoRepay	Boolean	<p>Whether auto-repay is allowed or not in <code>Spot mode</code></p> <p>true: Enabled</p> <p>false: Disabled</p>
type	String	<p>Account type</p> <p>0: Main account</p> <p>1: Standard sub-account</p> <p>2: Managed trading sub-account</p>

Parameter	Type	Description
		5: Custody trading sub-account - Copper 9: Managed trading sub-account - Copper 12: Custody trading sub-account - Komainu
settleCcy	String	Current account's USD-margined contract settle currency
settleCcyList	String	Current account's USD-margined contract settle currency list, like ["USD", "USDC", "USDG"].

#### SET POSITION MODE

Futures mode and Multi-currency mode: **FUTURES** and **SWAP** support both **long/short** mode and **net** mode. In **net** mode, users can only have positions in one direction; In **long/short** mode, users can hold positions in long and short directions.

Portfolio margin mode: **FUTURES** and **SWAP** only support **net** mode

#### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: USER ID

#### PERMISSION: TRADE

#### HTTP REQUEST

`POST /api/v5/account/set-position-mode`

#### Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Set position mode
result = accountAPI.set_position_mode(
    posMode="long_short_mode"
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
posMode	String	Yes	Position mode <b>long_short_mode</b> : long/short, only applicable to <b>FUTURES</b> / <b>SWAP</b> <b>net_mode</b> : net

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "posMode": "long_short_mode"
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
posMode	String	Position mode

Portfolio margin account only supports net mode

## SET LEVERAGE

There are 10 different scenarios for leverage setting:

1. Set leverage for **MARGIN** instruments under **isolated-margin** trade mode at pairs level.
2. Set leverage for **MARGIN** instruments under **cross-margin** trade mode and Spot mode (enabled borrow) at currency level.
3. Set leverage for **MARGIN** instruments under **cross-margin** trade mode and Futures mode account mode at pairs level.
4. Set leverage for **MARGIN** instruments under **cross-margin** trade mode and Multi-currency margin at currency level.
5. Set leverage for **MARGIN** instruments under **cross-margin** trade mode and Portfolio margin at currency level.
6. Set leverage for **FUTURES** instruments under **cross-margin** trade mode at underlying level.
7. Set leverage for **FUTURES** instruments under **isolated-margin** trade mode and buy/sell position mode at contract level.
8. Set leverage for **FUTURES** instruments under **isolated-margin** trade mode and long/short position mode at contract and position side level.
9. Set leverage for **SWAP** instruments under **cross-margin** trade at contract level.
10. Set leverage for **SWAP** instruments under **isolated-margin** trade mode and buy/sell position mode at contract level.
11. Set leverage for **SWAP** instruments under **isolated-margin** trade mode and long/short position mode at contract and position side level.

Note that the request parameter **posSide** is only required when margin mode is isolated in long/short position mode for FUTURES/SWAP instruments (see scenario 8 and 11 above).

Please refer to the request examples on the right for each case.

### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: TRADE

### HTTP REQUEST

**POST** /api/v5/account/set-leverage

#### Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Set leverage for MARGIN instruments under isolated-margin trade mode at pairs level.
result = accountAPI.set_leverage(
    instId="BTC-USDT",
    lever="5",
    mgnMode="isolated"
)
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Conditional	Instrument ID Only applicable to <b>cross</b> <b>FUTURES</b> <b>SWAP</b> or <b>Spot mode</b> / <b>Multi-currency margin</b> / <b>Portfolio margin</b> , <b>cross</b> <b>MARGIN</b> <b>FUTURES</b> <b>SWAP</b> and <b>isolated</b> position. And required in applicable scenarios.

Parameter	Type	Required	Description
ccy	String	Conditional	Currency used for margin, used for the leverage setting for the currency in auto borrow. Only applicable to <code>cross</code> <code>MARGIN</code> of <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code> . And required in applicable scenarios.
lever	String	Yes	Leverage
mgnMode	String	Yes	Margin mode <code>isolated</code> <code>cross</code> Can only be <code>cross</code> if <code>ccy</code> is passed.
posSide	String	Conditional	Position side <code>long</code> <code>short</code> Only required when margin mode is <code>isolated</code> in <code>long/short</code> mode for <code>FUTURES</code> / <code>SWAP</code> .

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "lever": "30",
      "mgnMode": "isolated",
      "instId": "BTC-USDT-SWAP",
      "posSide": "Long"
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
lever	String	Leverage
mgnMode	String	Margin mode <code>cross</code> <code>isolated</code>
instId	String	Instrument ID
posSide	String	Position side

When setting leverage for `cross` `FUTURES`/`SWAP` at the underlying level, pass in any instId and mgnMode(`cross`).

Leverage cannot be adjusted for the cross positions of Expiry Futures and Perpetual Futures under the portfolio margin account.

#### GET MAXIMUM ORDER QUANTITY

The maximum quantity to buy or sell. It corresponds to the "sz" from placement.

Under the Portfolio Margin account, the calculation of the maximum buy/sell amount or open amount is not supported under the cross mode of derivatives.

#### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: USER ID

#### PERMISSION: READ

#### HTTP REQUEST

## Request Example

```

import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Get maximum buy/sell amount or open amount
result = accountAPI.get_max_order_size(
    instId="BTC-USDT",
    tdMode="isolated"
)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Single instrument or multiple instruments (no more than 5) in the same instrument type separated with comma, e.g. <code>BTC-USDT,ETH-USDT</code>
tdMode	String	Yes	Trade mode <code>cross</code> <code>isolated</code> <code>cash</code> <code>spot_isolated</code> : only applicable to <code>Futures mode</code> .
ccy	String	Conditional	Currency used for margin Applicable to <code>isolated</code> <code>MARGIN</code> and <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code> .
px	String	No	Price When the price is not specified, it will be calculated according to the current limit price for <code>FUTURES</code> and <code>SWAP</code> , the last traded price for other instrument types. The parameter will be ignored when multiple instruments are specified.
leverage	String	No	Leverage for instrument The default is current leverage Only applicable to <code>MARGIN/FUTURES/SWAP</code>
tradeQuoteCcy	String	No	The quote currency used for trading. Only applicable to <code>SPOT</code> . The default value is the quote currency of the <code>instId</code> , for example: for <code>BTC-USD</code> , the default is <code>USD</code> .

## Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ccy": "BTC",
      "instId": "BTC-USDT",
      "maxBuy": "0.0500695098559788",
      "maxSell": "64.4798671570072269"
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID
ccy	String	Currency used for margin
maxBuy	String	<p><b>SPOT</b> / <b>MARGIN</b>: The maximum quantity in base currency that you can buy</p> <p>The cross-margin order under <b>Futures mode</b> mode, quantity of coins is based on base currency.</p> <p><b>FUTURES</b> / <b>SWAP</b> / <b>OPTIONS</b>: The maximum quantity of contracts that you can buy</p>
maxSell	String	<p><b>SPOT</b> / <b>MARGIN</b>: The maximum quantity in quote currency that you can sell</p> <p>The cross-margin order under <b>Futures mode</b> mode, quantity of coins is based on base currency.</p> <p><b>FUTURES</b> / <b>SWAP</b> / <b>OPTIONS</b>: The maximum quantity of contracts that you can sell</p>

#### GET MAXIMUM AVAILABLE BALANCE/EQUITY

Available balance for isolated margin positions and SPOT, available equity for cross margin positions.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

GET /api/v5/account/max-avail-size

Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Get maximum available transaction amount for SPOT BTC-USDT
result = accountAPI.get_max_avail_size(
    instId="BTC-USDT",
    tdMode="cash"
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Single instrument or multiple instruments (no more than 5) separated with comma, e.g. <b>BTC-USDT,ETH-USDT</b>
ccy	String	Conditional	Currency used for margin Applicable to <b>isolated</b> <b>MARGIN</b> and <b>cross</b> <b>MARGIN</b> in <b>Futures mode</b> .
tdMode	String	Yes	Trade mode <b>cross</b> <b>isolated</b> <b>cash</b> <b>spot_isolated</b> : only applicable to <b>Futures mode</b>
reduceOnly	Boolean	No	Whether to reduce position only Only applicable to <b>MARGIN</b>

Parameter	Type	Required	Description
px	String	No	The price of closing position. Only applicable to reduceOnly <b>MARGIN</b> .
tradeQuoteCcy	String	No	The quote currency used for trading. Only applicable to <b>SPOT</b> . The default value is the quote currency of the <b>instId</b> , for example: for <b>BTC-USD</b> , the default is <b>USD</b> .

### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "instId": "BTC-USDT",
      "availBuy": "100",
      "availSell": "1"
    }
  ]
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID
availBuy	String	Maximum available balance/equity to buy
availSell	String	Maximum available balance/equity to sell

In the case of SPOT/MARGIN, availBuy is in the quote currency, and availSell is in the base currency.

In the case of MARGIN with cross tdMode, both availBuy and availSell are in the currency passed in **ccy**.

### INCREASE/DECREASE MARGIN

Increase or decrease the margin of the isolated position. Margin reduction may result in the change of the actual leverage.

#### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: USER ID

#### PERMISSION: TRADE

#### HTTP REQUEST

```
POST /api/v5/account/position/margin-balance
```

### Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Increase margin
result = accountAPI.adjustment_margin(
  instId="BTC-USDT-SWAP",
  posSide="short",
  type= "add",
  amt="1"
```

```
)  
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID
posSide	String	Yes	Position side, the default is <code>net</code> <code>long</code> <code>short</code> <code>net</code>
type	String	Yes	<code>add</code> : add margin <code>reduce</code> : reduce margin
amt	String	Yes	Amount to be increased or decreased.
ccy	String	Conditional	Currency Applicable to <code>isolated</code> <code>MARGIN</code> orders

## Response Example

```
{  
  "code": "0",  
  "msg": "",  
  "data": [{  
    "amt": "0.3",  
    "ccy": "BTC",  
    "instId": "BTC-USDT",  
    "leverage": "",  
    "posSide": "net",  
    "type": "add"  
  }]  
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID
posSide	String	Position side, <code>long</code> <code>short</code>
amt	String	Amount to be increase or decrease
type	String	<code>add</code> : add margin <code>reduce</code> : reduce margin
leverage	String	Real leverage after the margin adjustment
ccy	String	Currency

## Manual transfer mode

The value of the margin initially assigned to the isolated position must be greater than or equal to 10,000 USDT, and a position will be created on the account.

## GET LEVERAGE

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

## HTTP REQUEST

GET /api/v5/account/leverage-info

## Request Example

```

import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Get leverage
result = accountAPI.get_leverage(
    instId="BTC-USDT-SWAP",
    mgnMode="cross"
)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Conditional	Instrument ID Single instrument ID or multiple instrument IDs (no more than 20) separated with comma
ccy	String	Conditional	Currency, used for getting leverage of currency level. Applicable to <code>cross</code> <code>MARGIN</code> of <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code> . Supported single currency or multiple currencies (no more than 20) separated with comma.
mgnMode	String	Yes	Margin mode <code>cross</code> <code>isolated</code>

## Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ccy": "",
      "instId": "BTC-USDT-SWAP",
      "mgnMode": "cross",
      "posSide": "long",
      "lever": "10"
    },
    {
      "ccy": "",
      "instId": "BTC-USDT-SWAP",
      "mgnMode": "cross",
      "posSide": "short",
      "lever": "10"
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID
ccy	String	Currency, used for getting leverage of currency level. Applicable to <code>cross</code> <code>MARGIN</code> of <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code> .
mgnMode	String	Margin mode

Parameter	Type	Description
posSide	String	Position side <input type="button" value="long"/> <input type="button" value="short"/> <input type="button" value="net"/> In <input type="button" value="long/short"/> mode, the leverage in both directions <input type="button" value="long"/> / <input type="button" value="short"/> will be returned.
lever	String	Leverage

Leverage cannot be enquired for the cross positions of Expiry Futures and Perpetual Futures under the portfolio margin account.

#### GET LEVERAGE ESTIMATED INFO

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

#### HTTP REQUEST

GET /api/v5/account/adjust-leverage-info

Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	Yes	Instrument type <input type="button" value="MARGIN"/> <input type="button" value="SWAP"/> <input type="button" value="FUTURES"/>
mgnMode	String	Yes	Margin mode <input type="button" value="isolated"/> <input type="button" value="cross"/>
lever	String	Yes	Leverage
instId	String	Conditional	Instrument ID, e.g. BTC-USDT It is required for these scenarios: <input type="button" value="SWAP"/> and <input type="button" value="FUTURES"/> , Margin isolation, Margin cross in <input type="button" value="Futures mode"/> .
ccy	String	Conditional	Currency used for margin, e.g. BTC It is required for isolated margin and cross margin in <input type="button" value="Futures mode"/> , <input type="button" value="Multi-currency margin"/> and <input type="button" value="Portfolio margin"/>
posSide	String	No	posSide <input type="button" value="net"/> : The default value <input type="button" value="long"/> <input type="button" value="short"/>

Response Example

```
{
  "code": "0",
  "data": [
    {
      "estAvailQuoteTrans": "",
      "estAvailTrans": "1.1398040558348279",
      "estLiqPx": "",
      "estMaxAmt": "10.6095865868904898",
      "estMgn": "0.0701959441651721",
      "estQuoteMaxAmt": "176889.6871254563042714",
      "estQuoteMgn": ""
    }
  ]
}
```

```

        "existOrd": false,
        "maxLever": "10",
        "minLever": "0.01"
    }
],
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
estAvailQuoteTrans	String	The estimated margin(in quote currency) can be transferred out under the corresponding leverage For cross, it is the maximum quantity that can be transferred from the trading account. For isolated, it is the maximum quantity that can be transferred from the isolated position Only applicable to <b>MARGIN</b>
estAvailTrans	String	The estimated margin can be transferred out under the corresponding leverage. For cross, it is the maximum quantity that can be transferred from the trading account. For isolated, it is the maximum quantity that can be transferred from the isolated position The unit is base currency for <b>MARGIN</b> It is not applicable to the scenario when increasing leverage for isolated position under <b>FUTURES</b> and <b>SWAP</b>
estLiqPx	String	The estimated liquidation price under the corresponding leverage. Only return when there is a position.
estMgn	String	The estimated margin needed by position under the corresponding leverage. For the <b>MARGIN</b> position, it is margin in base currency
estQuoteMgn	String	The estimated margin (in quote currency) needed by position under the corresponding leverage
estMaxAmt	String	For <b>MARGIN</b> , it is the estimated maximum loan in base currency under the corresponding leverage For <b>SWAP</b> and <b>FUTURES</b> , it is the estimated maximum quantity of contracts that can be opened under the corresponding leverage
estQuoteMaxAmt	String	The <b>MARGIN</b> estimated maximum loan in quote currency under the corresponding leverage.
existOrd	Boolean	Whether there is pending orders <b>true</b> <b>false</b>
maxLever	String	Maximum leverage
minLever	String	Minimum leverage

## GET THE MAXIMUM LOAN OF INSTRUMENT

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

## HTTP REQUEST

```
GET /api/v5/account/max-loan
```

Request Example

```

import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Max loan of cross MARGIN in Futures mode (Margin Currency is BTC)

```

```

result = accountAPI.get_max_loan(
    instId="BTC-USDT",
    mgnMode="cross",
    mgnCcy="BTC"
)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
mgnMode	String	Yes	Margin mode <code>isolated</code> <code>cross</code>
instId	String	Conditional	Single instrument or multiple instruments (no more than 5) separated with comma, e.g. <code>BTC-USDT,ETH-USD</code> T
ccy	String	Conditional	Currency Applicable to get Max loan of manual borrow for the currency in <code>Spot mode</code> (enabled borrowing)
mgnCcy	String	Conditional	Margin currency Applicable to <code>isolated</code> <code>MARGIN</code> and <code>cross</code> <code>MARGIN</code> in <code>Futures mode</code> .
tradeQuoteCcy	String	No	The quote currency for trading. Only applicable to <code>SPOT</code> . The default value is the quote currency of <code>instId</code> , e.g. <code>USD</code> for <code>BTC-USD</code> .

## Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "instId": "BTC-USDT",
      "mgnMode": "isolated",
      "mgnCcy": "",
      "maxLoan": "0.1",
      "ccy": "BTC",
      "side": "sell"
    },
    {
      "instId": "BTC-USDT",
      "mgnMode": "isolated",
      "mgnCcy": "",
      "maxLoan": "0.2",
      "ccy": "USDT",
      "side": "buy"
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID
mgnMode	String	Margin mode
mgnCcy	String	Margin currency
maxLoan	String	Max loan
ccy	String	Currency
side	String	Order side <code>buy</code> <code>sell</code>

## GET FEE RATES

RATE LIMIT: 5 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: READ

HTTP REQUEST

```
GET /api/v5/account/trade-fee
```

Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Get trading fee rates of current account
result = accountAPI.get_fee_rates(
    instType="SPOT",
    instId="BTC-USDT"
)
print(result)
```

REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	Yes	Instrument type <input type="button" value="SPOT"/> <input type="button" value="MARGIN"/> <input type="button" value="SWAP"/> <input type="button" value="FUTURES"/> <input type="button" value="OPTION"/>
instId	String	No	Instrument ID, e.g. <input type="button" value="BTC-USDT"/> Applicable to <input type="button" value="SPOT"/> / <input type="button" value="MARGIN"/>
instFamily	String	No	Instrument family, e.g. <input type="button" value="BTC-USD"/> Applicable to <input type="button" value="FUTURES"/> / <input type="button" value="SWAP"/> / <input type="button" value="OPTION"/>
ruleType	String	No	Trading rule types <input type="button" value="normal"/> : normal trading <input type="button" value="pre_market"/> : pre-market trading ruleType can not be passed through together with instId/instFamily

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "category": "1", //Deprecated
      "delivery": "",
      "exercise": "",
      "instType": "SPOT",
      "level": "1v1",
      "maker": "-0.0008",
      "makerU": "",
      "makerUSDC": "",
      "taker": "-0.001",
      "takerU": "",
      "takerUSDC": ""
    }
  ]
}
```

```

"ruleType": "normal",
"ts": "1608623351857",
"fiat": []
}
]
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
level	String	Fee rate Level
taker	String	For <b>SPOT</b> / <b>MARGIN</b> , it is taker fee rate of the USDT trading pairs. For <b>FUTURES</b> / <b>SWAP</b> / <b>OPTION</b> , it is the fee rate of crypto-margined contracts
maker	String	For <b>SPOT</b> / <b>MARGIN</b> , it is maker fee rate of the USDT trading pairs. For <b>FUTURES</b> / <b>SWAP</b> / <b>OPTION</b> , it is the fee rate of crypto-margined contracts
takerU	String	Taker fee rate of USDT-margined contracts, only applicable to <b>FUTURES</b> / <b>SWAP</b>
makerU	String	Maker fee rate of USDT-margined contracts, only applicable to <b>FUTURES</b> / <b>SWAP</b>
delivery	String	Delivery fee rate
exercise	String	Fee rate for exercising the option
instType	String	Instrument type
takerUSDC	String	For <b>SPOT</b> / <b>MARGIN</b> , it is taker fee rate of the USD◎&Crypto trading pairs. For <b>FUTURES</b> / <b>SWAP</b> , it is the fee rate of USDC-margined contracts
makerUSDC	String	For <b>SPOT</b> / <b>MARGIN</b> , it is maker fee rate of the USD◎&Crypto trading pairs. For <b>FUTURES</b> / <b>SWAP</b> , it is the fee rate of USDC-margined contracts
ruleType	String	Trading rule types <b>normal</b> : normal trading <b>pre_market</b> : pre-market trading
ts	String	Data return time, Unix timestamp format in milliseconds, e.g. <b>1597026383085</b>
category	String	Currency category. Note: this parameter is already deprecated
fiat	Array of objects	Details of fiat fee rate
> ccy	String	Fiat currency.
> taker	String	Taker fee rate
> maker	String	Maker fee rate

#### Remarks:

The fee rate like maker and taker: positive number, which means the rate of rebate; negative number, which means the rate of commission.  
Exception: The values for delivery and exercise are positive numbers, representing the commission rate.

USD◎ represent the stablecoin besides USDT

The Open API will not reflect zero-fee trading. For zero-fee pairs, please refer to <https://www.okx.com/fees> .

Get the interest accrued data for the past year

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

## RATE LIMIT RULE: USER ID

## PERMISSION: READ

## HTTP REQUEST

GET /api/v5/account/interest-accrued

## Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Get interest accrued data
result = accountAPI.get_interest accrued()
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
type	String	No	Loan type ②: Market loans Default is ②
ccy	String	No	Loan currency, e.g. BTC Only applicable to Market loans Only applicable to MARGIN
instId	String	No	Instrument ID, e.g. BTC-USDT Only applicable to Market loans
mgnMode	String	No	Margin mode cross isolated Only applicable to Market loans
after	String	No	Pagination of data to return records earlier than the requested timestamp, Unix timestamp format in milliseconds, e.g. 1597026383085
before	String	No	Pagination of data to return records newer than the requested, Unix timestamp format in milliseconds, e.g. 1597026383085
limit	String	No	Number of results per request. The maximum is 100. The default is 100.

### Response Example

```

    "interestFreeLiab": "",
    "mgnMode": "",
    "ts": "1637312400000",
    "type": "1"
},
{
    "ccy": "USDT",
    "instId": "",
    "interest": "0.000408333333334",
    "interestRate": "0.00004083333333",
    "liab": "100",
    "mgnMode": "",
    "ts": "1637049600000",
    "type": "1"
}
],
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
type	String	Loan type [2]: Market loans
ccy	String	Loan currency, e.g. <code>BTC</code>
instId	String	Instrument ID, e.g. <code>BTC-USDT</code> Only applicable to <code>Market loans</code>
mgnMode	String	Margin mode <code>cross</code> <code>isolated</code>
interest	String	Interest accrued
interestRate	String	Hourly borrowing interest rate
liab	String	Liability
totalLiab	String	Total liability for current account
interestFreeLiab	String	Interest-free liability for current account
ts	String	Timestamp for interest accrued, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## GET INTEREST RATE

Get the user's current leveraged currency borrowing market interest rate

### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: READ

### HTTP REQUEST

`GET /api/v5/account/interest-rate`

### Request Example

```

import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

```

```
accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Get the user's current leveraged currency borrowing interest rate
result = accountAPI.get_interest_rate()
print(result)
```

## REQUEST PARAMETERS

Parameters	Types	Required	Description
ccy	String	No	Currency, e.g. <code>BTC</code>

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ccy": "BTC",
      "interestRate": "0.0001"
    },
    {
      "ccy": "LTC",
      "interestRate": "0.0003"
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
interestRate	String	Hourly borrowing interest rate
ccy	String	Currency

## SET FEE TYPE

Set the fee type.

fee type selection is only effective for Spot.

## RATE LIMIT: 5 REQUESTS PER 2 SECONDS

## RATE LIMIT RULE: USER ID

## PERMISSION: TRADE

## HTTP REQUEST

`POST /api/v5/account/set-fee-type`

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
feeType	String	Yes	Fee type <input type="radio"/> 0: fee is charged in the currency you receive from the trade (default) <input checked="" type="radio"/> 1: fee is always charged in the quote currency of the trading pair (only effective for Spot)

Response Example

```
{
  "code": "0",
  "msg": "",
```

```

"feeType": [
  {
    "feeType": "0"
  }
]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
feeType	String	Fee type 0: fee is charged in the currency you receive from the trade 1: fee is always charged in the quote currency of the trading pair

## SET GREEKS (PA/BS)

Set the display type of Greeks.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

POST /api/v5/account/set-greeks

Request Example

```

import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Set greeks (PA/BS)
result = accountAPI.set_greeks(greeksType="PA")
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
greeksType	String	Yes	Display type of Greeks. PA: Greeks in coins BS: Black-Scholes Greeks in dollars

Response Example

```

{
  "code": "0",
  "msg": "",
  "data": [
    {
      "greeksType": "PA"
    }
  ]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
greeksType	String	Display type of Greeks.

## ISOLATED MARGIN TRADING SETTINGS

You can set the currency margin and futures/perpetual Isolated margin trading mode

### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: TRADE

### HTTP REQUEST

POST /api/v5/account/set-isolated-mode

#### Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Isolated margin trading settings
result = accountAPI.set_isolated_mode(
    isoMode="automatic",
    type="MARGIN"
)
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
isoMode	String	Yes	Isolated margin trading settings auto_transfers_ccy: New auto transfers, enabling both base and quote currency as the margin for isolated margin trading. Only applicable to MARGIN. automatic: Auto transfers
type	String	Yes	Instrument type MARGIN CONTRACTS

When there are positions and pending orders in the current account, the margin transfer mode from position to position cannot be adjusted.

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "isoMode": "automatic"
    }
  ],
  "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
isoMode	String	Isolated margin trading settings automatic: Auto transfers

## CONTRACTS

Auto transfers: Automatically occupy and release the margin when opening and closing positions

## MARGIN

Auto transfers: Automatically borrow and return coins when opening and closing positions

### GET MAXIMUM WITHDRAWALS

Retrieve the maximum transferable amount from trading account to funding account. If no currency is specified, the transferable amount of all owned currencies will be returned.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

GET /api/v5/account/max-withdrawal

#### Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Get maximum withdrawals
result = accountAPI.get_max_withdrawal()
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	No	Single currency or multiple currencies (no more than 20) separated with comma, e.g. <code>BTC</code> or <code>BTC,ETH</code> .

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ccy": "BTC",
      "maxWd": "124",
      "maxWdEx": "125",
      "spotOffsetMaxWd": "",
      "spotOffsetMaxWdEx": ""
    },
    {
      "ccy": "ETH",
      "maxWd": "10",
      "maxWdEx": "12",
      "spotOffsetMaxWd": "",
      "spotOffsetMaxWdEx": ""
    }
  ]
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency
maxWd	String	Max withdrawal (excluding borrowed assets under <a href="#">Spot mode</a> / <a href="#">Multi-currency margin</a> / <a href="#">Portfolio margin</a> )
maxWdEx	String	Max withdrawal (including borrowed assets under <a href="#">Spot mode</a> / <a href="#">Multi-currency margin</a> / <a href="#">Portfolio margin</a> )
spotOffsetMaxWd	String	Max withdrawal under Spot-Derivatives risk offset mode (excluding borrowed assets under <a href="#">Portfolio margin</a> ) Applicable to <a href="#">Portfolio margin</a>
spotOffsetMaxWdEx	String	Max withdrawal under Spot-Derivatives risk offset mode (including borrowed assets under <a href="#">Portfolio margin</a> ) Applicable to <a href="#">Portfolio margin</a>

#### GET ACCOUNT RISK STATE

Only applicable to Portfolio margin account

**RATE LIMIT: 10 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/account/risk-state`

Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Get account risk state
result = accountAPI.get_account_position_risk()
print(result)
```

Response Example

```
{
  "code": "0",
  "data": [
    {
      "atRisk": false,
      "atRiskIdx": [],
      "atRiskMgn": [],
      "ts": "1635745078794"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameters	Types	Description
atRisk	Boolean	Account risk status in auto-borrow mode true: the account is currently in a specific risk state false: the account is currently not in a specific risk state
atRiskIdx	Array of strings	derivatives risk unit list

Parameters	Types	Description
atRiskMgn	Array of strings	margin risk unit list
ts	String	Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

#### GET BORROW INTEREST AND LIMIT

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

#### HTTP REQUEST

`GET /api/v5/account/interest-limits`

#### Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Get borrow interest and limit
result = accountAPI.get_interest_limits(
    ccy="BTC"
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
type	String	No	Loan type ②: Market loans Default is ②
ccy	String	No	Loan currency, e.g. <code>BTC</code>

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "debt": "0.85893159114900247077000000000000",
      "interest": "0.00000000000000000000000000000000",
      "loanAlloc": "",
      "nextDiscountTime": "1729490400000",
      "nextInterestTime": "1729490400000",
      "records": [
        {
          "availLoan": "",
          "avgRate": "",
          "ccy": "BTC",
          "interest": "0",
          "loanQuota": "175.0000000",
          "posLoan": "",
          "rate": "0.0000276",
          "surplusLmt": "175.0000000",
          "surplusLmtDetails": {},
          "usedLmt": "0.0000000",
          "usedLoan": ""
        }
      ]
    }
  ]
}
```

```

        "interestFreeLiab": "",
        "potentialBorrowingAmt": ""
    }
],
"msg": ""
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
debt	String	Current debt in <a href="#">USD</a>
interest	String	Current interest in <a href="#">USD</a> , the unit is <a href="#">USD</a> Only applicable to <a href="#">Market loans</a>
nextDiscountTime	String	Next deduct time, Unix timestamp format in milliseconds, e.g. <a href="#">1597026383085</a>
nextInterestTime	String	Next accrual time, Unix timestamp format in milliseconds, e.g. <a href="#">1597026383085</a>
loanAlloc	String	VIP Loan allocation for the current trading account 1. The unit is percent(%). Range is [0, 100]. Precision is 0.01% 2. If master account did not assign anything, then "0" 3. "" if shared between master and sub-account
records	Array of objects	Details for currencies
> ccy	String	Loan currency, e.g. <a href="#">BTC</a>
> rate	String	Current daily borrowing rate
> loanQuota	String	Borrow limit of master account If loan allocation has been assigned, then it is the borrow limit of the current trading account
> surplusLmt	String	Available amount across all sub-accounts If loan allocation has been assigned, then it is the available amount to borrow by the current trading account
> usedLmt	String	Borrowed amount for current account If loan allocation has been assigned, then it is the borrowed amount by the current trading account
> interest	String	Interest to be deducted Only applicable to <a href="#">Market loans</a>
> interestFreeLiab	String	Interest-free liability for current account
> potentialBorrowingAmt	String	Potential borrowing amount for current account
> surplusLmtDetails	Object	The details of available amount across all sub-accounts The value of <a href="#">surplusLmt</a> is the minimum value within this array. It can help you judge the reason that <a href="#">surplusLmt</a> is not enough. Only applicable to <a href="#">VIP loans</a> Deprecated
>> allAcctRemainingQuota	String	Total remaining quota for master account and sub-accounts Deprecated
>> curAcctRemainingQuota	String	The remaining quota for the current account. Only applicable to the case in which the sub-account is assigned the loan allocation Deprecated
>> platRemainingQuota	String	Remaining quota for the platform. The format like "600" will be returned when it is more than <a href="#">curAcctRemainingQuota</a> or <a href="#">allAcctRemainingQuota</a> Deprecated

Parameter	Type	Description
> posLoan	String	Frozen amount for current account (Within the locked quota) Only applicable to <b>VIP loans</b> Deprecated
> availLoan	String	Available amount for current account (Within the locked quota) Only applicable to <b>VIP loans</b> Deprecated
> usedLoan	String	Borrowed amount for current account Only applicable to <b>VIP loans</b> Deprecated
> avgRate	String	Average hourly interest of borrowed coin Only applicable to <b>VIP loans</b> Deprecated

#### MANUAL BORROW / REPAY

Only applicable to **Spot mode** (enabled borrowing)

**RATE LIMIT: 1 REQUEST PER 3 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

`POST /api/v5/account/spot-manual-borrow-repay`

Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1
accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

result = accountAPI.spot_manual_borrow_repay(ccy="USDT", side="borrow", amt="1")
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	Yes	Currency, e.g. <b>BTC</b>
side	String	Yes	Side <b>borrow</b> <b>repay</b>
amt	String	Yes	Amount

Response Example

```
{
  "code": "0",
  "data": [
    {
      "ccy": "USDT",
      "side": "borrow",
      "amt": "100"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency, e.g. <code>BTC</code>
side	String	Side <code>borrow</code> <code>repay</code>
amt	String	Actual amount

## SET AUTO REPAY

Only applicable to `Spot mode` (enabled borrowing)

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

`POST /api/v5/account/set-auto-repay`

Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1
accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

result = accountAPI.set_auto_repay(autoRepay=True)
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
autoRepay	Boolean	Yes	Whether auto repay is allowed or not under <code>Spot mode</code> <code>true</code> : Enable auto repay <code>false</code> : Disable auto repay

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "autoRepay": true
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
autoRepay	Boolean	Whether auto repay is allowed or not under <code>Spot mode</code> <code>true</code> : Enable auto repay <code>false</code> : Disable auto repay

## GET BORROW/REPAY HISTORY

Retrieve the borrow/repay history under `Spot mode`

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/account/spot-borrow-repay-history`

Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1
accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

result = accountAPI.spot_borrow_repay_history(ccy="USDT", type="auto_borrow")
print(result)
```

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
ccy	String	No	Currency, e.g. <code>BTC</code>
type	String	No	Event type <code>auto_borrow</code> <code>auto_repay</code> <code>manual_borrow</code> <code>manual_repay</code>
after	String	No	Pagination of data to return records earlier than the requested <code>ts</code> (included), Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
before	String	No	Pagination of data to return records newer than the requested <code>ts</code> (included), Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. The maximum is 100. The default is 100.

Response Example

```
{
  "code": "0",
  "data": [
    {
      "accBorrowed": "0",
      "amt": "6764.802661157592",
      "ccy": "USDT",
      "ts": "1725330976644",
      "type": "auto_repay"
    }
  ],
  "msg": ""
}
```

**RESPONSE PARAMETERS**

Parameter	Type	Description
ccy	String	Currency, e.g. <code>BTC</code>
type	String	Event type <code>auto_borrow</code> <code>auto_repay</code> <code>manual_borrow</code> <code>manual_repay</code>
amt	String	Amount
accBorrowed	String	Accumulated borrow amount
ts	String	Timestamp for the event, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## POSITION BUILDER (NEW)

Calculates portfolio margin information for virtual position/assets or current position of the user.

You can add up to 200 virtual positions and 200 virtual assets in one request.

**RATE LIMIT: 2 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`POST /api/v5/account/position-builder`

Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

result = accountAPI.position_builder(
    inclRealPosAndEq=True,
    simPos=[
        {
            "pos": "10",
            "instId": "BTC-USDT-SWAP"
        },
        {
            "pos": "10",
            "instId": "LTC-USDT-SWAP"
        }
    ]
)
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
acctLv	String	No	Switch to account mode <code>3</code> : Multi-currency margin <code>4</code> : Portfolio margin The default is <code>4</code>
inclRealPosAndEq	Boolean	No	Whether import existing positions and assets The default is <code>true</code>

Parameter	Type	Required	Description
lever	String	No	Cross margin leverage in Multi-currency margin mode, the default is 1. If the allowed leverage is exceeded, set according to the maximum leverage. Only applicable to Multi-currency margin
simPos	Array of objects	No	List of simulated positions
> instId	String	Yes	Instrument ID, e.g. BTC-USDT-SWAP Applicable to SWAP/FUTURES/OPTION
> pos	String	Yes	Quantity of positions
> avgPx	String	Yes	Average open price
> lever	String	No	leverage Only applicable to Multi-currency margin The default is 1 If the allowed leverage is exceeded, set according to the maximum leverage.
simAsset	Array of objects	No	List of simulated assets When inclRealPosAndEq is true, only real assets are considered and virtual assets are ignored
> ccy	String	Yes	Currency, e.g. BTC
> amt	String	Yes	Currency amount
greeksType	String	No	Greeks type BS: Black-Scholes Model Greeks PA: Crypto Greeks CASH: Empirical Greeks The default is BS
idxVol	String	No	Price volatility percentage, indicating what this price change means towards each of the values. In decimal form, range -0.99 ~ 1, in 0.01 increment. Default 0

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "acctLever": "-0.1364949794742562",
      "assets": [
        {
          "availEq": "0",
          "borrowImr": "0",
          "borrowMmr": "",
          "ccy": "BTC",
          "spotInUse": "0"
        },
        {
          "availEq": "0",
          "borrowImr": "0",
          "borrowMmr": "",
          "ccy": "LTC",
          "spotInUse": "0"
        },
        {
          "availEq": "0",
          "borrowImr": "0",
          "borrowMmr": "",
          "ccy": "USDC",
          "spotInUse": "0"
        },
        {
          "availEq": "-78589.37",
          "borrowImr": "0",
          "borrowMmr": "",
          "ccy": "USDT",
          "spotInUse": "0"
        }
      ]
    }
  ]
}
```

```

"borrowImr": "78553.21888979999",
"borrowMmr": "",
"ccy": "USDT",
"spotInUse": "0"
},
],
"borrowMmr": "1571.064377796",
"derivMmr": "1375.4837063088003",
"eq": "-78553.21888979999",
"marginRatio": "-25.95365779811705",
"positions": [],
"riskUnitData": [
{
"delta": "-9704.903689800001",
"gamma": "0",
"imrBf": "",
"imr": "1538.9669514070802",
"mmrBf": "",
"mmr": "1183.8207318516002",
"mr1": "1164.4109244719994",
"mr1FinalResult": {
"pnl": "-1164.4109244719994",
"spotShock": "0.12",
"volShock": "up"
},
"mr1Scenarios": {
"volSame": {
"0": "0",
"0.08": "-776.2739496480004",
"-0.08": "776.2739496480004",
"0.04": "-388.1369748240002",
"0.12": "-1164.4109244719994",
"-0.12": "1164.4109244719994",
"-0.04": "388.1369748240002"
},
"volShockDown": {
"0": "0",
"0.08": "-776.2739496480004",
"-0.08": "776.2739496480004",
"0.04": "-388.1369748240002",
"0.12": "-1164.4109244719994",
"-0.12": "1164.4109244719994",
"-0.04": "388.1369748240002"
},
"volShockUp": {
"0": "0",
"0.08": "-776.2739496480004",
"-0.08": "776.2739496480004",
"0.04": "-388.1369748240002",
"0.12": "-1164.4109244719994",
"-0.12": "1164.4109244719994",
"-0.04": "388.1369748240002"
}
},
"mr2": "0",
"mr3": "0",
"mr4": "19.4098073796",
"mr5": "0",
"mr6": "1164.4109244720003",
"mr6FinalResult": {
"pnl": "-2328.8218489440005",
"spotShock": "0.24"
},
"mr7": "43.67206660410001",
"mr8": "1571.064377796",
"mr9": "0",
"portfolios": [
{
"amt": "-10",
"avgPx": "100000",
"delta": "-9704.903689800001",
"floatPnl": "290.630000000003",
"gamma": "0",
"instId": "BTC-USDT-SWAP",
"instType": "SWAP",
"isRealPos": false,
"markPxBr": ""
}
]
}

```

```

        "markPx": "97093.7",
        "notionalUsd": "9703.22",
        "posSide": "net",
        "theta": "0",
        "vega": "0"
    }
],
"riskUnit": "BTC",
"theta": "0",
"up1": "290.49631020000027",
"vega": "0"
},
{
"delta": "1019.5308",
"gamma": "0",
"imrBf": "",
"imr": "249.16186679436",
"mmrBf": "",
"mmr": "191.6629744572",
"mr1": "183.50672805719995",
"mr1FinalResult": {
    "pnl": "-183.50672805719995",
    "spotShock": "-0.18",
    "volShock": "up"
},
"mr1Scenarios": {
    "volSame": {
        "0": "0",
        "-0.06": "-61.168909352399936",
        "0.06": "61.168909352399936",
        "-0.18": "-183.50672805719995",
        "0.18": "183.50672805719995",
        "0.12": "122.33781870480001",
        "-0.12": "-122.33781870480001"
    },
    "volShockDown": {
        "0": "0",
        "-0.06": "-61.168909352399936",
        "0.06": "61.168909352399936",
        "-0.18": "-183.50672805719995",
        "0.18": "183.50672805719995",
        "0.12": "122.33781870480001",
        "-0.12": "-122.33781870480001"
    },
    "volShockUp": {
        "0": "0",
        "-0.06": "-61.168909352399936",
        "0.06": "61.168909352399936",
        "-0.18": "-183.50672805719995",
        "0.18": "183.50672805719995",
        "0.12": "122.33781870480001",
        "-0.12": "-122.33781870480001"
    }
},
"mr2": "0",
"mr3": "0",
"mr4": "8.1562464",
"mr5": "0",
"mr6": "183.5067280572",
"mr6FinalResult": {
    "pnl": "-367.0134561144",
    "spotShock": "-0.36"
},
"mr7": "7.1367156",
"mr8": "1571.064377796",
"mr9": "0",
"portfolios": [
    {
        "amt": "10",
        "avgPx": "8000",
        "delta": "1019.5308",
        "floatPnl": "-78980",
        "gamma": "0",
        "instId": "LTC-USDT-SWAP",
        "instType": "SWAP",
        "isRealPos": false,
        "markPxBf": ""
    }
]
}

```

```

        "markPx": "102",
        "notionalUsd": "1018.9",
        "posSide": "net",
        "theta": "0",
        "vega": "0"
    }
],
"riskUnit": "LTC",
"theta": "0",
"upl": "-78943.6692",
"vega": "0"
}
],
"totalImr": "9643.45070718144",
"totalMmr": "2946.5480841048",
"ts": "1736936801642",
"upl": "-78653.1728898"
}
],
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameters	Types	Description
eq	String	Adjusted equity (USD) for the account
totalMmr	String	Total MMR (USD) for the account
totalImr	String	Total IMR (USD) for the account
borrowMmr	String	Borrow MMR (USD) for the account
derivMmr	String	Derivatives MMR (USD) for the account
marginRatio	String	Cross maintenance margin ratio for the account
upl	String	UPL for the account
acctLever	String	Leverage of the account
ts	String	Update time for the account, Unix timestamp format in milliseconds, e.g. 1597026383085
assets	Array of objects	Asset info
> ccy	String	Currency, e.g. BTC
> availEq	String	Currency equity
> spotInUse	String	Spot in use
> borrowMmr	String	Borrowing MMR (USD) (Deprecated)
> borrowImr	String	Borrowing IMR (USD)
riskUnitData	Array of objects	Risk unit info
> riskUnit	String	Risk unit, e.g. BTC
> mmrBf	String	Risk unit MMR before volatility (USD) Return "" if users don't pass in idxVol
> mmr	String	Risk unit MMR (USD)

Parameters	Types	Description
> imrBf	String	Risk unit IMR before volatility (USD) Return "" if users don't pass in idxVol
> imr	String	Risk unit IMR (USD)
> upl	String	Risk unit UPL (USD)
> mr1	String	Stress testing value of spot and volatility (all derivatives, and spot trading in spot-derivatives risk offset mode)
> mr2	String	Stress testing value of time value of money (TVM) (for options)
> mr3	String	Stress testing value of volatility span (for options)
> mr4	String	Stress testing value of basis (for all derivatives)
> mr5	String	Stress testing value of interest rate risk (for options)
> mr6	String	Stress testing value of extremely volatile markets (for all derivatives, and spot trading in spot-derivatives risk offset mode)
> mr7	String	Stress testing value of position reduction cost (for all derivatives)
> mr8	String	Borrowing MMR/IMR
> mr9	String	USDT-USDC-USD hedge risk
> mr1Scenarios	Object	MR1 scenario analysis
>> volShockDown	Object	When volatility shocks down, the P&L of stress tests under different price volatility ratios, format in { change: value,...} change: price volatility ratio (in percentage), e.g. 0.01 representing 1% value: P&L under stress tests, measured in USD e.g. {"-0.15": "-2333.23", ...}
		When volatility keeps the same, the P&L of stress tests under different price volatility ratios, format in { change: value,...} change: price volatility ratio (in percentage), e.g. 0.01 representing 1% value: P&L under stress tests, measured in USD e.g. {"-0.15": "-2333.23", ...}
>> volShockUp	Object	When volatility shocks up, the P&L of stress tests under different price volatility ratios, format in { change: value,...} change: price volatility ratio (in percentage), e.g. 0.01 representing 1% value: P&L under stress tests, measured in USD e.g. {"-0.15": "-2333.23", ...}
		MR1 worst-case scenario
>> pnl	String	MR1 stress P&L (USD)
>> spotShock	String	MR1 worst-case scenario spot shock (in percentage), e.g. 0.01 representing 1%
>> volShock	String	MR1 worst-case scenario volatility shock down: volatility shock down unchange: volatility unchanged up: volatility shock up
		MR6 scenario analysis
>> pnl	String	MR6 stress P&L (USD)
>> spotShock	String	MR6 worst-case scenario spot shock (in percentage), e.g. 0.01 representing 1%

Parameters	Types	Description
> delta	String	(Risk unit) The rate of change in the contract's price with respect to changes in the underlying asset's price. When the price of the underlying changes by x, the option's price changes by delta multiplied by x.
> gamma	String	(Risk unit) The rate of change in the delta with respect to changes in the underlying price. When the price of the underlying changes by x%, the option's delta changes by gamma multiplied by x%.
> theta	String	(Risk unit) The change in contract price each day closer to expiry.
> vega	String	(Risk unit) The change of the option price when underlying volatility increases by 1%.
> portfolios	Array of objects	Portfolios info Only applicable to <a href="#">Portfolio margin</a>
>> instId	String	Instrument ID, e.g. <a href="#">BTC-USDT-SWAP</a>
>> instType	String	Instrument type <a href="#">SPOT</a> <a href="#">SWAP</a> <a href="#">FUTURES</a> <a href="#">OPTION</a>
>> amt	String	When <a href="#">instType</a> is <a href="#">SPOT</a> , it represents spot in use. When <a href="#">instType</a> is <a href="#">SWAP</a> / <a href="#">FUTURES</a> / <a href="#">OPTION</a> , it represents position amount.
>> posSide	String	Position side <a href="#">long</a> <a href="#">short</a> <a href="#">net</a>
>> avgPx	String	Average open price
>> markPxBf	String	Mark price before price volatility Return "" if users don't pass in idxVol
>> markPx	String	Mark price
>> floatPnl	String	Float P&L
>> notionalUsd	String	Notional in <a href="#">usd</a>
>> delta	String	When <a href="#">instType</a> is <a href="#">SPOT</a> , it represents asset amount. When <a href="#">instType</a> is <a href="#">SWAP</a> / <a href="#">FUTURES</a> / <a href="#">OPTION</a> , it represents the rate of change in the contract's price with respect to changes in the underlying asset's price (by Instrument ID).
>> gamma	String	The rate of change in the delta with respect to changes in the underlying price (by Instrument ID). When <a href="#">instType</a> is <a href="#">SPOT</a> , it will return "".
>> theta	String	The change in contract price each day closer to expiry (by Instrument ID). When <a href="#">instType</a> is <a href="#">SPOT</a> , it will return "".
>> vega	String	The change of the option price when underlying volatility increases by 1% (by Instrument ID). When <a href="#">instType</a> is <a href="#">SPOT</a> , it will return "".
>> isRealPos	Boolean	Whether it is a real position If <a href="#">instType</a> is <a href="#">SWAP</a> / <a href="#">FUTURES</a> / <a href="#">OPTION</a> , it is a valid parameter, else it will return <a href="#">false</a>
positions	Array of objects	Position info Only applicable to <a href="#">Multi-currency margin</a>
> instId	String	Instrument ID, e.g. <a href="#">BTC-USDT-SWAP</a>
> instType	String	Instrument type <a href="#">SPOT</a>

Parameters	Types	Description
		<div style="display: flex; justify-content: space-around; align-items: center;"> <span>SWAP</span> <span>FUTURES</span> <span>OPTION</span> </div>
> amt	String	When <code>instType</code> is <code>SPOT</code> , it represents spot in use. When <code>instType</code> is <code>SWAP</code> / <code>FUTURES</code> / <code>OPTION</code> , it represents position amount.
> posSide	String	Position side <div style="display: flex; justify-content: space-around; align-items: center;"> <span>long</span> <span>short</span> <span>net</span> </div>
> avgPx	String	Average open price
> markPxBf	String	Mark price before price volatility Return "" if users don't pass in <code>idxVol</code>
> markPx	String	Mark price
> floatPnl	String	Float P&L
> imrBf	String	IMR before price volatility
> imr	String	IMR
> mgnRatio	String	Maintenance margin ratio
> lever	String	Leverage
> notionalUsd	String	Notional in <code>usd</code>
> isRealPos	Boolean	Whether it is a real position If <code>instType</code> is <code>SWAP</code> / <code>FUTURES</code> / <code>OPTION</code> , it is a valid parameter, else it will return <code>false</code>

## POSITION BUILDER TREND GRAPH

RATE LIMIT: 1 REQUEST PER 5 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: READ

HTTP REQUEST

`POST /api/v5/account/position-builder-graph`

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
inclRealPosAndEq	Boolean	No	Whether to import existing positions and assets The default is <code>true</code>
simPos	Array of objects	No	List of simulated positions
> instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT-SWAP</code> Applicable to <code>SWAP</code> / <code>FUTURES</code> / <code>OPTION</code>
> pos	String	Yes	Quantity of positions
> avgPx	String	Yes	Average open price
> lever	String	No	leverage Only applicable to <code>Multi-currency margin</code>

Parameter	Type	Required	Description
			The default is 1 If the allowed leverage is exceeded, set according to the maximum leverage.
simAsset	Array of objects	No	List of simulated assets When <code>inclRealPosAndEq</code> is true, only real assets are considered and virtual assets are ignored
> ccy	String	Yes	Currency, e.g. BTC
> amt	String	Yes	Currency amount
type	String	Yes	Trending graph type mmr
mmrConfig	Object	Yes	MMR configuration
> acctLv	String	No	Switch to account mode 3: Multi-currency margin 4: Portfolio margin
> lever	String	No	Cross margin leverage in Multi-currency margin mode, the default is 1. If the allowed leverage is exceeded, set according to the maximum leverage. Only applicable to Multi-currency margin

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "type": "mmr",
      "mmrData": [
        ....
        {
          "mmr": "1415.0254039225917",
          "mmrRatio": "-47.45603627655477",
          "shockFactor": "-0.94"
        },
        {
          "mmr": "1417.732491243024",
          "mmrRatio": "-47.436684685735386",
          "shockFactor": "-0.93"
        }
        ....
      ]
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameters	Types	Description
type	String	Graph type mmr
mmrData	Array	Array of mmrData Return data in shockFactor ascending order
> shockFactor	String	Price change ratio, data range -1 to 1.
> mmr	String	Mmr at specific price
> mmrRatio	String	Maintenance margin ratio at specific price

## SET RISK OFFSET AMOUNT

Set risk offset amount. This does not represent the actual spot risk offset amount. Only applicable to Portfolio Margin Mode.

### RATE LIMIT: 10 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: TRADE

### HTTP REQUEST

```
POST /api/v5/account/set-riskOffset-amt
```

Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	Yes	Currency
clSpotInUseAmt	String	Yes	Spot risk offset amount defined by users

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ccy": "BTC",
      "clSpotInUseAmt": "0.5"
    }
  ]
}
```

### RESPONSE PARAMETERS

Parameters	Types	Description
ccy	String	Currency
clSpotInUseAmt	String	Spot risk offset amount defined by users

## GET GREEKS

Retrieve a greeks list of all assets in the account.

### RATE LIMIT: 10 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: READ

### HTTP REQUEST

```
GET /api/v5/account/greeks
```

Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Retrieve a greeks list of all assets in the account
```

```
result = accountAPI.get_greeks()
print(result)
```

## REQUEST PARAMETERS

Parameters	Types	Required	Description
ccy	String	No	Single currency, e.g. <code>BTC</code> .

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "thetaBS": "",
      "thetaPA": "",
      "deltaBS": "",
      "deltaPA": "",
      "gammaBS": "",
      "gammaPA": "",
      "vegaBS": "",
      "vegaPA": "",
      "ccy": "BTC",
      "ts": "1620282889345"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameters	Types	Description
deltaBS	String	delta: Black-Scholes Greeks in dollars
deltaPA	String	delta: Greeks in coins
gammaBS	String	gamma: Black-Scholes Greeks in dollars, only applicable to OPTION
gammaPA	String	gamma: Greeks in coins, only applicable to OPTION
thetaBS	String	theta: Black-Scholes Greeks in dollars, only applicable to <code>OPTION</code>
thetaPA	String	theta: Greeks in coins, only applicable to <code>OPTION</code>
vegaBS	String	vega: Black-Scholes Greeks in dollars, only applicable to <code>OPTION</code>
vegaPA	String	vega: Greeks in coins, only applicable to <code>OPTION</code>
ccy	String	Currency
ts	String	Time of getting Greeks, Unix timestamp format in milliseconds, e.g. 1597026383085

## GET PM POSITION LIMITATION

Retrieve cross position limitation of SWAP/FUTURES/OPTION under Portfolio margin mode.

### RATE LIMIT: 10 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: READ

### HTTP REQUEST

```
GET /api/v5/account/position-tiers
```

## Request Example

```
import okx.Account as Account

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

accountAPI = Account.AccountAPI(apikey, secretkey, passphrase, False, flag)

# Get PM position limitation
result = accountAPI.get_account_position_tiers(
    instType="SWAP",
    uly="BTC-USDT"
)
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	Yes	Instrument type <input type="button" value="SWAP"/> <input type="button" value="FUTURES"/> <input type="button" value="OPTION"/>
instFamily	String	Yes	Single instrument family or instrument families (no more than 5) separated with comma.

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "instFamily": "BTC-USDT",
      "maxSz": "10000",
      "postType": "",
      "uly": "BTC-USDT"
    }
  ],
  "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
uly	String	Underlying Applicable to <input type="button" value="FUTURES"/> / <input type="button" value="SWAP"/> / <input type="button" value="OPTION"/>
instFamily	String	Instrument family Applicable to <input type="button" value="FUTURES"/> / <input type="button" value="SWAP"/> / <input type="button" value="OPTION"/>
maxSz	String	Max number of positions
postType	String	Limitation of position type, only applicable to cross <input type="button" value="OPTION"/> under portfolio margin mode ①: Contracts of pending orders and open positions for all derivatives instruments. ②: Contracts of pending orders for all derivatives instruments. ③: Pending orders for all derivatives instruments. ④: Contracts of pending orders and open positions for all derivatives instruments on the same side. ⑤: Pending orders for one derivatives instrument. ⑥: Contracts of pending orders and open positions for one derivatives instrument. ⑦: Contracts of one pending order.

### ACTIVATE OPTION

RATE LIMIT: 5 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

**PERMISSION: TRADE****HTTP REQUEST**`POST /api/v5/account/activate-option`**Request Example****REQUEST PARAMETERS**

None

**Response Example**

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ts": "1600000000000"
    }
  ]
}
```

**RESPONSE PARAMETERS**

Parameter	Type	Description
ts	String	Activation time

**SET AUTO LOAN**Only applicable to `Multi-currency margin` and `Portfolio margin`**RATE LIMIT: 5 REQUESTS PER 2 SECONDS****RATE LIMIT RULE: USER ID****PERMISSION: TRADE****HTTP REQUEST**`POST /api/v5/account/set-auto-loan`**Request Example****REQUEST PARAMETERS**

Parameter	Type	Required	Description
autoLoan	Boolean	No	Whether to automatically make loans Valid values are <code>true</code> , <code>false</code> The default is <code>true</code>

**Response Example**

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "autoLoan": true
    }
  ]
}
```

**RESPONSE PARAMETERS**

Parameter	Type	Description
autoLoan	Boolean	Whether to automatically make loans

**PRESET ACCOUNT MODE SWITCH**

Pre-set the required information for account mode switching. When switching from [Portfolio margin mode](#) back to [Futures mode](#) / [Multi-currency margin mode](#), and if there are existing cross-margin contract positions, it is mandatory to pre-set leverage.

If the user does not follow the required settings, they will receive an error message during the pre-check or when setting the account mode.

#### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: USER ID

#### PERMISSION: TRADE

#### HTTP REQUEST

```
POST /api/v5/account/account-level-switch-preset
```

Request example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
acctLv	String	Yes	<p>Account mode ②: Futures mode ③: Multi-currency margin mode ④: Portfolio margin mode</p>
lever	String	Optional	<p>Leverage Required when switching from Portfolio margin mode to <a href="#">Futures mode</a> or <a href="#">Multi-currency margin mode</a>, and the user holds cross-margin positions.</p>
riskOffsetType	String	Optional	<p>Risk offset type ①: Spot-derivatives (USDT) risk offset ②: Spot-derivatives (Crypto) risk offset ③: Derivatives only mode ④: Spot-derivatives (USDC) risk offset Applicable when switching from <a href="#">Futures mode</a> or <a href="#">Multi-currency margin mode</a> to <a href="#">Portfolio margin mode</a>. (Deprecated)</p>

Response example 1. Futures mode -> Multi-currency margin mode

```
{
  "acctLv": "3",
  "curAcctLv": "2",
  "lever": "",
  "riskOffsetType": ""
}
```

Response example 2. Multi-currency margin mode -> Futures mode

```
{
  "acctLv": "2",
  "curAcctLv": "3",
  "lever": "",
  "riskOffsetType": ""
}
```

Response example 3. Portfolio margin mode -> Futures mode/Multi-currency margin mode

```
{
  "acctLv": "2",
  "curAcctLv": "4",
  "lever": "10",
  "riskOffsetType": ""
}
```

Response example 4. Portfolio margin mode -> Futures mode/Multi-currency margin mode

```
{
  "acctLv": "3",
  "curAcctLv": "4",
  "lever": "",
  "riskOffsetType": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
curAcctLv	String	Current account mode
acctLv	String	Account mode after switch
lever	String	The leverage user preset for cross-margin positions
riskOffsetType	String	The risk offset type user preset(Deprecated)

lever: When switching from Portfolio margin mode to Futures mode or Multi-currency margin mode, if the user holds cross-margin positions, this parameter must be provided; otherwise, error code 50014 will occur. The maximum allowable value for this parameter is determined by the smallest maximum leverage based on current position sizes under the target mode. For example, if a user in PM mode holds three cross-margin positions, with maximum allowable leverage of 20x, 50x, and 100x respectively, the maximum leverage it can set is 20x.

## PRECHECK ACCOUNT MODE SWITCH

Retrieve precheck information for account mode switching.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

```
GET /api/v5/account/set-account-switch-precheck
```

Request example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
acctLv	String	Yes	Account mode ①: Spot mode ②: Futures mode ③: Multi-currency margin code ④: Portfolio margin mode

Response example. Futures mode->Portfolio margin mode, need to finish the Q&A on web or mobile first

```
{
  "code": "51070",
  "data": [],
  "msg": "You do not meet the requirements for switching to this account mode. Please upgrade the account mode on the OKX website or App"
}
```

Response example. Futures mode->Portfolio margin mode, unmatched information. sCode 1

```
{
  "code": "0",
  "data": [
    {
      "s": "1"
    }
  ]
}
```

```

"acctLv": "3",
"curAcctLv": "1",
"mgnAft": null,
"mgnBf": null,
"posList": [],
"posTierCheck": [],
"riskOffsetType": "",
"sCode": "1",
"unmatchedInfoCheck": [
  {
    "posList": [],
    "totalAsset": "",
    "type": "repay_borrowings"
  }
]
},
"msg": ""
}

```

Response example. Portfolio margin mode->Multi-currency margin code, the user has cross-margin positions but doesn't preset leverage. sCode 3

```

{
  "code": "0",
  "data": [
    {
      "acctLv": "3",
      "curAcctLv": "4",
      "mgnAft": null,
      "mgnBf": null,
      "posList": [
        {
          "lever": "50",
          "posId": "2005456500916518912"
        },
        {
          "lever": "10",
          "posId": "2005456108363218944"
        },
        {
          "lever": "100",
          "posId": "2005456332909477888"
        },
        {
          "lever": "1",
          "posId": "2005456415990251520"
        }
      ],
      "posTierCheck": [],
      "riskOffsetType": "",
      "sCode": "3",
      "unmatchedInfoCheck": []
    }
  ],
  "msg": ""
}

```

Response example. Portfolio margin mode->Multi-currency margin code, the user finishes the leverage setting to 10, and passes the position tier an margin check. sCode 0.

```

{
  "code": "0",
  "data": [
    {
      "acctLv": "3",
      "curAcctLv": "4",
      "mgnAft": {
        "acctAvailEq": "106002.2061970689",
        "details": [],
        "mgnRatio": "148.1652396878421"
      },
      "mgnBf": {

```

```

"acctAvailEq": "77308.89735228613",
"details": [],
"mgnRatio": "4.460069474634038"
},
"posList": [
{
"lever": "50",
"posId": "2005456500916518912"
},
{
"lever": "50",
"posId": "2005456108363218944"
},
{
"lever": "50",
"posId": "2005456332909477888"
},
{
"lever": "50",
"posId": "2005456415990251520"
}
],
"posTierCheck": [],
"riskOffsetType": "",
"sCode": "0",
"unmatchedInfoCheck": []
}
],
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
sCode	String	<p>Check code</p> <p>0: pass all checks 1: unmatched information 3: leverage setting is not finished 4: position tier or margin check is not passed</p>
curAcctLv	String	<p>Account mode</p> <p>1: Spot mode 2: Futures mode 3: Multi-currency margin mode 4: Portfolio margin mode</p> <p>Applicable to all scenarios</p>
acctLv	String	<p>Account mode</p> <p>1: Spot mode 2: Futures mode 3: Multi-currency margin mode 4: Portfolio margin mode</p> <p>Applicable to all scenarios</p>
riskOffsetType	String	<p>Risk offset type</p> <p>1: Spot-derivatives (USDT) risk offset 2: Spot-derivatives (Crypto) risk offset 3: Derivatives only mode 4: Spot-derivatives (USDC) risk offset</p> <p>Applicable when acctLv is 4, return "" for other scenarios</p> <p>If the user preset before, it will use the user's specified value; if not, the default value 3 will be applied (Deprecated)</p>
unmatchedInfoCheck	Array of objects	<p>Unmatched information list</p> <p>Applicable when sCode is 1, indicating there is unmatched information; return [] for other scenarios</p>
>> type	String	<p>Unmatched information type</p> <p>asset_validation: asset validation</p>

Parameter	Type	Description
		<p><code>pending_orders</code>: order book pending orders</p> <p><code>pending_algos</code>: pending algo orders and trading bots, such as iceberg, recurring buy and twap</p> <p><code>isolated_margin</code>: isolated margin (quick margin and manual transfers)</p> <p><code>isolated_contract</code>: isolated contract (manual transfers)</p> <p><code>contract_long_short</code>: contract positions in hedge mode</p> <p><code>cross_margin</code>: cross margin positions</p> <p><code>cross_option_buyer</code>: cross options buyer</p> <p><code>isolated_option</code>: isolated options (only applicable to spot mode)</p> <p><code>growth_fund</code>: positions with trial funds</p> <p><code>all_positions</code>: all positions</p> <p><code>spot_lead_copy_only_simple_single</code>: copy trader and customize lead trader can only use spot mode or Futures mode</p> <p><code>stop_spot_custom</code>: spot customize copy trading</p> <p><code>stop_futures_custom</code>: contract customize copy trading</p> <p><code>lead_portfolio</code>: lead trader can not switch to portfolio margin mode</p> <p><code>futures_smart_sync</code>: you can not switch to spot mode when having smart contract sync</p> <p><code>vip_fixed_loan</code>: vip loan</p> <p><code>repay_borrowings</code>: borrowings</p> <p><code>compliance_restriction</code>: due to compliance restrictions, margin trading services are unavailable</p> <p><code>compliance_kyc2</code>: Due to compliance restrictions, margin trading services are unavailable. If you are not a resident of this region, please complete kyc2 identity verification.</p>
<code>&gt;&gt; totalAsset</code>	<code>String</code>	<p>Total assets</p> <p>Only applicable when type is <code>asset_validation</code>, return "" for other scenarios</p>
<code>&gt;&gt; posList</code>	<code>Array of strings</code>	<p>Unmatched position list (posId)</p> <p>Applicable when type is related to positions, return [] for other scenarios</p>
<code>posList</code>	<code>Array of objects</code>	<p>Cross margin contract position list</p> <p>Applicable when curAcctLv is 4, acctLv is 2/3 and user has cross margin contract positions</p> <p>Applicable when sCode is 0/3/4</p>
<code>&gt; posId</code>	<code>String</code>	Position ID
<code>&gt; lever</code>	<code>String</code>	Leverage of cross margin contract positions after switch
<code>posTierCheck</code>	<code>Array of objects</code>	<p>Cross margin contract positions that don't pass the position tier check</p> <p>Only applicable when sCode is 4</p>
<code>&gt; instFamily</code>	<code>String</code>	Instrument family
<code>&gt; instType</code>	<code>String</code>	<p>Instrument type</p> <p><code>SWAP</code></p> <p><code>FUTURES</code></p> <p><code>OPTION</code></p>
<code>&gt; pos</code>	<code>String</code>	Quantity of position
<code>&gt; lever</code>	<code>String</code>	Leverage
<code>&gt; maxSz</code>	<code>String</code>	If acctLv is 2/3, it refers to the maximum position size allowed at the current leverage. If acctLv is 4, it refers to the maximum position limit for cross-margin positions under the PM mode.
<code>mgnBf</code>	<code>Object</code>	<p>The margin related information before switching account mode</p> <p>Applicable when sCode is 0/4, return null for other scenarios</p>
<code>&gt; acctAvailEq</code>	<code>String</code>	<p>Account available equity in USD</p> <p>Applicable when curAcctLv is 3/4, return "" for other scenarios</p>
<code>&gt; mgnRatio</code>	<code>String</code>	<p>Maintenance Margin ratio in USD</p> <p>Applicable when curAcctLv is 3/4, return "" for other scenarios</p>

Parameter	Type	Description
> details	Array of objects	Detailed information Only applicable when curAcctLv is 2, return "" for other scenarios
>> ccy	String	Currency
>> availEq	String	Available equity of currency
>> mgnRatio	String	Maintenance margin ratio of currency
mgnAft	Object	The margin related information after switching account mode Applicable when sCode is 0/4, return null for other scenarios
> acctAvailEq	String	Account available equity in USD Applicable when acctLv is 3/4, return "" for other scenarios
> mgnRatio	String	Maintenance margin ratio in USD Applicable when acctLv is 3/4, return "" for other scenarios
> details	Array of objects	Detailed information Only applicable when acctLv is 2, return "" for other scenarios
>> ccy	String	Currency
>> availEq	String	Available equity of currency
>> mgnRatio	String	Maintenance margin ratio of currency

#### SET ACCOUNT MODE

You need to set on the Web/App for the first set of every account mode. If users plan to switch account modes while holding positions, they should first call the preset endpoint to conduct necessary settings, then call the precheck endpoint to get unmatched information, margin check, and other related information, and finally call the account mode switch endpoint to switch account modes.

#### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: USER ID

#### PERMISSION: TRADE

#### HTTP REQUEST

POST /api/v5/account/set-account-level

Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
acctLv	String	Yes	Account mode 1: Spot mode 2: Futures mode 3: Multi-currency margin code 4: Portfolio margin mode

Response Example

```
{
  "code": "0",
  "data": [
    {
      "acctLv": "1"
    }
  ],
}
```

```
        "msg": ""  
    }  
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
acctLv	String	Account mode

## SET COLLATERAL ASSETS

RATE LIMIT: 5 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: TRADE

HTTP REQUEST

```
POST /api/v5/account/set-collateral-assets
```

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
type	String	true	Type <code>all</code> <code>custom</code>
collateralEnabled	Boolean	true	Whether or not set the assets to be collateral <code>true</code> : Set to be collateral <code>false</code> : Set to be non-collateral
ccyList	Array of strings	conditional	Currency list, e.g. ["BTC", "ETH"] If type= <code>custom</code> , the parameter is required.

Response Example

```
{  
    "code": "0",  
    "msg": "",  
    "data": [  
        {  
            "type": "all",  
            "ccyList": ["BTC", "ETH"],  
            "collateralEnabled": false  
        }  
    ]  
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
type	String	Type <code>all</code> <code>custom</code>
collateralEnabled	Boolean	Whether or not set the assets to be collateral <code>true</code> : Set to be collateral <code>false</code> : Set to be non-collateral
ccyList	Array of strings	Currency list, e.g. ["BTC", "ETH"]

## GET COLLATERAL ASSETS

RATE LIMIT: 5 REQUESTS PER 2 SECONDS

**RATE LIMIT RULE: USER ID****PERMISSION: READ****HTTP REQUEST**`GET /api/v5/account/collateral-assets`**Request Example****REQUEST PARAMETERS**

Parameters	Types	Required	Description
ccy	String	No	Single currency or multiple currencies (no more than 20) separated with comma, e.g. "BTC" or "BTC,ETH".
collateralEnabled	Boolean	No	Whether or not to be a collateral asset

**Response Example**

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ccy": "BTC",
      "collateralEnabled": true
    },
    {
      "ccy": "ETH",
      "collateralEnabled": false
    }
  ]
}
```

**RESPONSE PARAMETERS**

Parameter	Type	Description
ccy	String	Currency, e.g. <code>BTC</code>
collateralEnabled	Boolean	Whether or not to be a collateral asset

**RESET MMP STATUS**

You can unfreeze by this endpoint once MMP is triggered.

Only applicable to Option in Portfolio Margin mode, and MMP privilege is required.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS****RATE LIMIT RULE: USER ID****PERMISSION: TRADE****HTTP REQUEST**`POST /api/v5/account/mmp-reset`**Request Example****REQUEST PARAMETERS**

Parameter	Type	Required	Description
instType	String	No	Instrument type <code>OPTION</code> The default is `OPTION`
instFamily	String	Yes	Instrument family

## Response Example

```
{  
  "code": "0",  
  "msg": "",  
  "data": [  
    {  
      "result": true  
    }  
  ]  
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
result	Boolean	Result of the request <code>true</code> , <code>false</code>

### SET MMP

This endpoint is used to set MMP config

Only applicable to Option in Portfolio Margin mode, and MMP privilege is required.

#### What is MMP?

Market Maker Protection (MMP) is an automated mechanism for market makers to pull their quotes when their executions exceed a certain threshold(`qtyLimit') within a certain time frame(`timeInterval'). Once mmp is triggered, any pre-existing mmp pending orders(`mmp` and `mmp\_and\_post\_only` orders) will be automatically canceled, and new orders tagged as MMP will be rejected for a specific duration(`frozenInterval'), or until manual reset by makers.

#### How to enable MMP?

Please send an email to [institutional@okx.com](mailto:institutional@okx.com) or contact your business development (BD) manager to apply for MMP. The initial threshold will be upon your request.

### RATE LIMIT: 2 REQUESTS PER 10 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: TRADE

### HTTP REQUEST

```
POST /api/v5/account/mmp-config
```

#### Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instFamily	String	Yes	Instrument family
timeInterval	String	Yes	Time window (ms). MMP interval where monitoring is done "0" means disable MMP
frozenInterval	String	Yes	Frozen period (ms). "0" means the trade will remain frozen until you request "Reset MMP Status" to unfrozen
qtyLimit	String	Yes	Trade qty limit in number of contracts Must be > 0

#### Response Example

```
{  
  "code": "0",  
  "msg": "",  
  "data": [  
    {  
      "result": true  
    }  
  ]  
}
```

```

"msg": "",
"data": [
  {
    "frozenInterval": "2000",
    "instFamily": "BTC-USD",
    "qtyLimit": "100",
    "timeInterval": "5000"
  }
]
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
instFamily	String	Instrument family
timeInterval	String	Time window (ms). MMP interval where monitoring is done
frozenInterval	String	Frozen period (ms).
qtyLimit	String	Trade qty limit in number of contracts

#### GET MMP CONFIG

This endpoint is used to get MMP configure information

Only applicable to Option in Portfolio Margin mode, and MMP privilege is required.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

GET /api/v5/account/mmp-config

Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instFamily	String	No	Instrument Family

Response Example

```

{
  "code": "0",
  "data": [
    {
      "frozenInterval": "2000",
      "instFamily": "ETH-USD",
      "mmpFrozen": true,
      "mmpFrozenUntil": "1000",
      "qtyLimit": "10",
      "timeInterval": "5000"
    }
  ],
  "msg": ""
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
instFamily	String	Instrument Family

Parameter	Type	Description
mmpFrozen	Boolean	Whether MMP is currently triggered. <code>true</code> or <code>false</code>
mmpFrozenUntil	String	If frozenInterval is configured and mmpFrozen = True, it is the time interval (in ms) when MMP is no longer triggered, otherwise "".
timeInterval	String	Time window (ms). MMP interval where monitoring is done
frozenInterval	String	Frozen period (ms). If it is "0", the trade will remain frozen until manually reset and <code>mmpFrozenUntil</code> will be "".
qtyLimit	String	Trade qty limit in number of contracts

## MOVE POSITIONS

Only applicable to users with a trading level greater than or equal to VIP5, and can only be called through the API Key of the master account. Users can check their trading level through the fee details table on the My trading fees page.

To move positions between different accounts under the same master account. Each source account can trigger up to fifteen move position requests every 24 hours. There is no limitation to the destination account to receive positions. Refer to the "Things to note" part for more details.

### RATE LIMIT: 1 REQUEST PER SECOND

### RATE LIMIT RULE: MASTER ACCOUNT USER ID

### HTTP REQUEST

`POST /api/v5/account/move-positions`

Request example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
fromAcct	String	Yes	Source account name. If it's a master account, it should be "0"
toAcct	String	Yes	Destination account name. If it's a master account, it should be "0"
legs	Array of Objects	Yes	An array of objects containing details of each position to be moved
>from	Object	yes	Details of the position in the source account
>>posId	String	Yes	Position ID in the source account
>>sz	String	Yes	Number of contracts.
>>side	String	Yes	Trade side from the perspective of source account <code>buy</code> <code>sell</code>
>to	Object	Yes	Details of the configuration of the destination account
>>tdMode	String	No	Trading mode in the destination account. <code>cross</code> <code>isolated</code> If not provided, tdMode will take the default values as shown below: Buy options in <code>Futures mode</code> / <code>Multi-currency margin mode</code> : <code>isolated</code> Other cases: <code>cross</code>

Parameter	Type	Required	Description
>>posSide	String	No	<p>Position side</p> <p><input type="checkbox"/> <b>net</b></p> <p><input type="checkbox"/> <b>long</b></p> <p><input type="checkbox"/> <b>short</b></p> <p>This parameter is not mandatory if the destination sub-account is in <b>net</b> mode. If you pass it through, the only valid value is <b>net</b>. It can only be <b>long</b> or <b>short</b> if the destination sub-account is in long/short mode. If not specified, destination account in long/short mode always open new positions.</p>
>>ccy	String	No	Margin currency in destination accountOnly applicable to cross margin positions in <a href="#">[Futures mode]</a> .
clientId	String	Yes	Client-supplied ID. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.

### Response example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "clientId": "test",
      "blockTdId": "2065832911119076864",
      "state": "filled",
      "ts": "1734069018526",
      "fromAcct": "0",
      "toAcct": "test",
      "legs": [
        {
          "from": {
            "posId": "206547111340792832",
            "instId": "BTC-USD-SWAP",
            "px": "100042.7",
            "side": "sell",
            "sz": "1",
            "sCode": "0",
            "sMsg": ""
          },
          "to": {
            "instId": "BTC-USD-SWAP",
            "px": "100042.7",
            "side": "buy",
            "sz": "1",
            "tdMode": "cross",
            "posSide": "net",
            "ccy": "",
            "sCode": "0",
            "sMsg": ""
          }
        },
        {
          "from": {
            "posId": "2063111180412153856",
            "instId": "BTC-USDT-SWAP",
            "px": "100008.1",
            "side": "sell",
            "sz": "1",
            "sCode": "0",
            "sMsg": ""
          },
          "to": {
            "instId": "BTC-USDT-SWAP",
            "px": "100008.1",
            "side": "buy",
            "sz": "1",
            "tdMode": "cross",
            "posSide": "net",
            "ccy": "",
            "sCode": "0",
            "sMsg": ""
          }
        }
      ]
    }
  ]
}
```

```

        }
    ]
}
}

```

### Response example:failure

```

// The destination account position mode (net/longShort) is not matched with the posSide field
{
    "code": "51000",
    "msg": "Incorrect type of posSide (leg with Instrument Id [BTC-USD-SWAP])",
    "data": []
}

// The BTC amount in the destination account is not enough to open the position.
{
    "code": "51008",
    "msg": "Order failed. Insufficient BTC margin in account",
    "data": []
}

```

### RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, <code>0</code> means success
msg	String	The error message, empty if the code is 0
blockTdId	String	Block trade ID
clientId	String	Client-supplied ID
state	String	Status of the order <code>filled</code> , <code>failed</code>
fromAcct	String	Source account name
toAcct	String	Destination account name
legs	Array	An array of objects containing details of each position to be moved
>from	Object	Object describing the "from" leg
>>instId	String	Instrument ID
>>posId	String	Position ID
>>px	String	Transfer price, typically a 60-minute TWAP of the mark price
>>side	String	Direction of the leg in the source account <code>buy</code> <code>sell</code>
>>sz	String	Number of Contracts
>>sCode	String	The code of the event execution result, 0 means success
>>sMsg	String	Rejection message if the request is unsuccessful
>to	Object	Object describing the "to" leg
>>instId	String	Instrument ID
>> side	String	Trade side of the trade in the destination account

Parameter	Type	Description
>>posSide	String	Position side of the trade in the destination account
>>tdMode	String	Trade mode
>>px	String	Transfer price, typically a 60-minute TWAP of the mark price
>>ccy	String	Margin currency
>>sCode	String	The code of the event execution result, 0 means success
>>sMsg	String	Rejection message if the request is unsuccessful
ts	String	Unix timestamp in milliseconds indicating when the transfer request was processed

#### THINGS TO NOTE

1. Only applicable to users with a trading level greater than or equal to VIP5, and can only be called through the API Key of the master account.
2. The source and destination accounts for move positions must be accounts under the same master account and they must be different.
3. For source account, a maximum of fifteen move position requests can be triggered within a 24-hour period. There is no limitation to the destination account to receive positions. Only successful requests are counted toward this limit.
4. The maximum number of legs per move position request is 30.
5. No move position fee will be charged at this time.
6. Moving positions is not supported in margin trading now.
7. The move position price is determined by the TWAP (Time-Weighted Average Price) of the mark price over the past 60 minutes, using the closing mark price per minute. If the symbol is newly listed and a 60-minute TWAP is unavailable, the move position will be rejected with error code 70065
8. The move position will share the same price limit as those in the order book. The move position will fail if the 60-minute mark price TWAP is outside of the price limit.
9. For the source account, move positions must be conducted in a reduce-only manner. You must choose the opposite side of your current position and specify a size equal to or smaller than your existing position size. The system will also process move position requests in a best-effort reduce-only manner.
10. The side field of source account leg (from) should be `sell` if you are holding a long position while the side of destination account leg (to) should be `buy`, vice versa for a short position.
11. The posSide field of destination account (to) should be `net` if it's in one-way mode; `long`/`short` if it's in hedge mode. If in hedge mode, you need to specify `long`/`short` to decide whether to close current positions or open reverse positions. Otherwise, it will always open new positions.
  1. Open long: buy and open long (side: buy; posSide: long)
  2. Open short: sell and open short (side: sell; posSide: short)
  3. Close long: sell and close long (side: sell; posSide: long)
  4. Close short: buy and close short (side: buy; posSide: short)
12. Historical records of move positions can be fetched from the *Get move positions history* endpoint but only for pending or successful requests.
13. Move positions operation counting example.

Transfer done within the day	Account A count (total)	Account B count (total)	Account C count (total)	Account D count (total)
Account A to Account B	1	0	0	0
Account B to Account C	1	1	0	0
Account B to Account D	1	2	0	0

#### GET MOVE POSITIONS HISTORY

Only applicable to users with a trading level greater than or equal to VIP5, and can only be called through the API Key of the master account. Users can check their trading level through the fee details table on the My trading fees page.

Retrieve move position details in the last 3 days.

#### RATE LIMIT: 2 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: MASTER ACCOUNT USERID

#### HTTP REQUEST

`GET /api/v5/account/move-positions-history`

## Request example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
blockTdId	String	No	BlockTdId generated by the system
clientId	String	No	Client-supplied ID. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
beginTs	String	No	Filter with a begin timestamp. Unix timestamp format in milliseconds (inclusive)
endTs	String	No	Filter with an end timestamp. Unix timestamp format in milliseconds (inclusive)
limit	String	No	Number of results per request. The maximum and default are both <a href="#">100</a>
state	String	No	Positions transfer state, <a href="#">filled</a> <a href="#">pending</a>

## Response example

```
{  
  "code": "0",  
  "msg": "",  
  "data": [  
    {  
      "clientId": "test",  
      "blockTdId": "2066393411110139648",  
      "state": "filled",  
      "ts": "1734085725000",  
      "fromAcct": "0",  
      "toAcct": "test",  
      "legs": [  
        {  
          "from": {  
            "posId": "2065477911110792832",  
            "instId": "BTC-USD-SWAP",  
            "px": "100123.8",  
            "side": "sell",  
            "sz": "1"  
          },  
          "to": {  
            "instId": "BTC-USD-SWAP",  
            "px": "100123.8",  
            "side": "buy",  
            "sz": "1",  
            "tdMode": "cross",  
            "posSide": "net",  
            "ccy": ""  
          }  
        },  
        {  
          "from": {  
            "posId": "206353311112153856",  
            "instId": "BTC-USDT-SWAP",  
            "px": "100078.7",  
            "side": "sell",  
            "sz": "1"  
          },  
          "to": {  
            "instId": "BTC-USDT-SWAP",  
            "px": "100078.7",  
            "side": "buy",  
            "sz": "1",  
            "tdMode": "cross",  
            "posSide": "net",  
            "ccy": ""  
          }  
        }  
      ]  
    }  
  ]
```

}

## RESPONSE PARAMETERS

Parameter	Type	Description
clientId	String	Client-supplied ID. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
blockTdId	String	Block trade ID.
state	String	Position transfer state, <code>filled</code> <code>pending</code>
ts	String	Unix timestamp in milliseconds indicating when the transfer request was processed
fromAcct	String	Source account name
toAcct	String	Destination account name
legs	Array	An array of objects containing details of each position to be moved
> from	Object	Object describing the "from" leg
>> instId	String	Instrument ID
>> posId	String	Position ID
>> px	String	Transfer price, typically a 60-minute TWAP of the mark price
>> side	String	Direction of the leg in the source account <code>buy</code> <code>sell</code>
>> sz	String	Number of Contracts
> to	Object	Object describing the "to" leg
>> instId	String	Instrument ID
>> px	String	Transfer price, typically a 60-minute TWAP of the mark price
>> side	String	Trade side from the perspective of destination account <code>buy</code> <code>sell</code>
>> sz	String	Number of contracts.
>> tdMode	String	Trading mode in the destination account <code>cross</code> <code>isolated</code>
>> posSide	String	Position side <code>net</code> <code>long</code> <code>short</code>
>> ccy	String	Margin currency in destination account Only applicable to cross margin positions in <code>Futures mode</code> .

## SET AUTO EARN

Turn on/off auto earn.

## RATE LIMIT: 2 REQUESTS PER 2 SECONDS

## RATE LIMIT RULE: USER ID

## HTTP REQUEST

POST /api/v5/account/set-auto-earn

Request example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
earnType	String	No	Auto earn type ①: auto earn (auto lend, auto staking) ②: auto earn (USDG earn) The default value is ①
ccy	String	Yes	Currency
action	String	Yes	Auto earn operation action turn_on: turn on auto earn turn_off: turn off auto earn amend: amend minimum lending APR, applicable only to earnType ② (deprecated)
apr	String	Optional	Minimum lending APR. Users must pass in this field when earnType is ② and action is turn_on/amend. 0.01 means 1%, available range 0.01-3.65, increment 0.01 (deprecated)

Response example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "earnType": "0",
      "ccy": "BTC",
      "action": "turn_on",
      "apr": "0.01"
    }
  ]
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
earnType	String	Auto earn type ①: auto earn (auto lend, auto staking) ②: auto earn (USDG earn)
ccy	String	Currency
action	Boolean	Auto earn operation action turn_on turn_off amend (deprecated)
apr	String	Minimum lending APR (deprecated)

### SET SETTLE CURRENCY

Only applicable to USD-margined contract.

**RATE LIMIT: 20 TIMES PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADING**

## HTTP REQUEST

POST /api/v5/account/set-settle-currency

Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
settleCcy	String	Yes	USD-margined contract settle currency

Response Example

```
{  
  "code": "0",  
  "msg": "",  
  "data": [  
    {  
      "settleCcy": "USDC"  
    }  
  ]  
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
settleCcy	String	USD-margined contract settle currency

#### SET TRADING CONFIG

RATE LIMIT: 1 REQUEST PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: TRADE

#### HTTP REQUEST

POST /api/v5/account/set-trading-config

Request example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
type	String	Yes	Trading config type <code>stgyType</code>
stgyType	String	No	Strategy type <code>0</code> : general strategy <code>1</code> : delta neutral strategy Only applicable when type is <code>stgyType</code>

Response example

```
{  
  "code": "0",  
  "msg": "",  
  "data": [  
    {  
      "type": "stgyType",  
      "stgyType": "1"  
    }  
  ]  
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
type	String	Trading config type
stgyType	String	Strategy type

#### PRECHECK SET DELTA NEUTRAL

RATE LIMIT: 1 REQUEST PER 2 SECONDS

RATE LIMIT RULE: USER ID

#### HTTP REQUEST

GET /api/v5/account/precheck-set-delta-neutral

Request example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
stgyType	String	Yes	Strategy type ①: general strategy ②: delta neutral strategy

Response example

```
{
  "code": "0",
  "data": [
    {
      "unmatchedInfoCheck": [
        {
          "posList": [],
          "ordList": [],
          "deltaLever": "",
          "type": "spot_mode"
        },
        {
          "posList": ["123", "123", "123"],
          "ordList": [],
          "deltaLever": "",
          "type": "isolated_margin"
        }
      ]
    },
    "msg": ""
  }
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
unmatchedInfoCheck	Array of objects	Unmatched information list
> type	String	Unmatched information type spot_mode: DNA is not supported under spot mode futures_mode: DNA is not supported under futures mode isolated_margin: Isolated margin position is not supported in DNA isolated_contract: Isolated contract position is not supported in DNA positions_options: Options are not supported in DNA isolated_pending_orders: Isolated pending orders are not supported in DNA pending_orders_options: Pending options orders are not supported in DNA trading_bot: Trading bot is not supported in DNA repay_borrowings: borrowing in the targeted strategy will exceed the main account borrowing limit after the switch. Repay liabilities and try again.

Parameter	Type	Description
		<p><code>loan</code>: Flexible loan and DNA cannot be used at the same time</p> <p><code>delta_risk</code>: delta risk check failed, lower delta and try again</p> <p><code>collateral_all</code>: all coins must be set as collateral in DNA</p>
> deltaLever	String	<p>Delta leverage</p> <p>Applicable when type is <code>delta_risk</code></p>
> ordList	Array of strings	<p>Unmatched order list, order ID</p> <p>Applicable when type is <code>isolated_pending_orders</code>/<code>pending_orders_options</code></p>
> posList	Array of strings	<p>Unmatched position list, position ID</p> <p>Applicable when type is <code>isolated_margin</code>/<code>isolated_contract</code>/<code>positions_options</code></p>

## WebSocket

### ACCOUNT CHANNEL

Retrieve account information. Data will be pushed when triggered by events such as placing order, canceling order, transaction execution, etc. It will also be pushed in regular interval according to subscription granularity.

Concurrent connection to this channel will be restricted by the following rules: WebSocket connection count limit.

### URL PATH

/ws/v5/private (required login)

Request Example : single

```
import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/private",
        useServerTime=False
    )
    await ws.start()
    args = [
        {
            "channel": "account",
            "ccy": "BTC"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

Request Example

```

import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/private",
        useServerTime=False
    )
    await ws.start()
    args = [
        {
            "channel": "account",
            "extraParams": "{\"updateInterval\": \"0\"}"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>account</code>
> ccy	String	No	Currency
> extraParams	String	No	Additional configuration
>> updateInterval	int	No	0: only push due to account events The data will be pushed both by events and regularly if this field is omitted or set to other values than 0. The following format should be strictly obeyed when using this field. "extraParams": " { \"updateInterval\": \"0\" } "

Successful Response Example : single

```
{  
  "id": "1512",  
  "event": "subscribe",  
  "arg": {  
    "channel": "account",  
    "ccy": "BTC"  
  },  
  "connId": "a4d3ae55"  
}
```

## Successful Response Example

```
{  
  "id": "1512",  
  "event": "subscribe",  
  "arg": {  
    "channel": "account"  
  },  
  "connId": "a4d3ae55"  
}
```

## Failure Response Example

```
{  
  "id": "1512",  
  "event": "error",  
  "code": "60012",  
  "msg": "Invalid request: {\\"op\\": \\"subscribe\\\", \\"argss\\": [{ \\"channel\\": \\"account\\\", \\"ccy\\": \\"BTC\\\" }]}"  
  "connId": "a4d3ae55"  
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Operation subscribe unsubscribe error
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name account
> ccy	String	No	Currency
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

## Push Data Example

```
{  
  "arg": {  
    "channel": "account",  
    "uid": "44*****584"  
  },  
  "eventType": "snapshot",  
  "curPage": 1,  
  "lastPage": true,  
  "data": [{  
    "addid": "55444_1231696934"  
  }]
```

```

"availEq": "55444.12216906034",
"borrowFroz": "0",
"delta": "0",
"deltaLever": "0",
"deltaNeutralStatus": "0",
"details": [
  {
    "availBal": "4734.371190691436",
    "availEq": "4734.371190691435",
    "borrowFroz": "0",
    "cashBal": "4750.426970691436",
    "ccy": "USDT",
    "coinUsdPrice": "0.99927",
    "crossLiab": "0",
    "colRes": "0",
    "collateralEnabled": false,
    "collateralRestrict": false,
    "colBorrAutoConversion": "0",
    "disEq": "4889.379316336831",
    "eq": "4892.951170691435",
    "eqUsd": "4889.379316336831",
    "smtSyncEq": "0",
    "spotCopyTradingEq": "0",
    "fixedBal": "0",
    "frozenBal": "158.57998",
    "frpType": "0",
    "imr": "",
    "interest": "0",
    "isoEq": "0",
    "isoLiab": "0",
    "isoUpl": "0",
    "liab": "0",
    "maxLoan": "0",
    "mgnRatio": "",
    "mmr": "",
    "notionalLever": "",
    "ordFrozen": "0",
    "rewardBal": "0",
    "spotInUseAmt": "",
    "clSpotInUseAmt": "",
    "maxSpotInUseAmt": "",
    "spotIsoBal": "0",
    "stgyEq": "150",
    "twap": "0",
    "uTime": "1705564213903",
    "upl": "-7.47580000000003",
    "uplLiab": "0",
    "spotBal": "",
    "openAvgPx": "",
    "accAvgPx": "",
    "spotUpl": "",
    "spotUplRatio": "",
    "totalPnl": "",
    "totalPnlRatio": ""
  }
],
"imr": "0",
"isoEq": "0",
"mgnRatio": "",
"mmr": "0",
"notionalUsd": "0",
"notionalUsdForBorrow": "0",
"notionalUsdForFutures": "0",
"notionalUsdForOption": "0",
"notionalUsdForSwap": "0",
"ordFroz": "0",
"totalEq": "55868.06403501676",
"uTime": "1705564223311",
"upl": "0"
}
]
}

```

## PUSH DATA PARAMETERS

Parameters	Types	Description
arg	Object	Successfully subscribed channel

Parameters	Types	Description
> channel	String	Channel name
> uid	String	User Identifier
eventType	String	Event type: <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">snapshot</span> : Initial and regular snapshot push <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">event_update</span> : Event-driven update push
curPage	Integer	Current page number. Only applicable for <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">snapshot</span> events. Not included in <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">event_update</span> events.
lastPage	Boolean	Whether this is the last page of pagination: <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">true</span> <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">false</span> Only applicable for <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">snapshot</span> events. Not included in <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">event_update</span> events.
data	Array of objects	Subscribed data
> uTime	String	The latest time to get account information, millisecond format of Unix timestamp, e.g. <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">1597026383085</span>
> totalEq	String	The total amount of equity in <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">USD</span>
> isoEq	String	Isolated margin equity in <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">USD</span> Applicable to <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Futures mode</span> / <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Multi-currency margin</span> / <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Portfolio margin</span>
> adjEq	String	Adjusted / Effective equity in <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">USD</span> The net fiat value of the assets in the account that can provide margins for spot, expiry futures, perpetual futures and options under the cross-margin mode. In multi-ccy or PM mode, the asset and margin requirement will all be converted to USD value to process the order check or liquidation. Due to the volatility of each currency market, our platform calculates the actual USD value of each currency based on discount rates to balance market risks. Applicable to <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Spot mode</span> / <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Multi-currency margin</span> / <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Portfolio margin</span>
> availEq	String	Account level available equity, excluding currencies that are restricted due to the collateralized borrowing limit. Applicable to <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Multi-currency margin</span> / <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Portfolio margin</span>
> ordFroz	String	Margin frozen for pending cross orders in <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">USD</span> Only applicable to <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Spot mode</span> / <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Multi-currency margin</span>
> imr	String	Initial margin requirement in <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">USD</span> The sum of initial margins of all open positions and pending orders under cross-margin mode in <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">USD</span> . Applicable to <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Spot mode</span> / <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Multi-currency margin</span> / <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Portfolio margin</span>
> mmr	String	Maintenance margin requirement in <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">USD</span> The sum of maintenance margins of all open positions and pending orders under cross-margin mode in <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">USD</span> . Applicable to <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Spot mode</span> / <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Multi-currency margin</span> / <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Portfolio margin</span>
> borrowFroz	String	Potential borrowing IMR of the account in <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">USD</span> Only applicable to <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Spot mode</span> / <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Multi-currency margin</span> / <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Portfolio margin</span> . It is "" for other margin modes.
> mgnRatio	String	Maintenance margin ratio in <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">USD</span> . Applicable to <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Spot mode</span> / <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Multi-currency margin</span> / <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Portfolio margin</span>
> notionalUsd	String	Notional value of positions in <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">USD</span> Applicable to <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Spot mode</span> / <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Multi-currency margin</span> / <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Portfolio margin</span>
> notionalUsdForBorrow	String	Notional value for <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Borrow</span> in USD Applicable to <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Spot mode</span> / <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Multi-currency margin</span> / <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">Portfolio margin</span>

Parameters	Types	Description
> notionalUsdForSwap	String	Notional value of positions for <b>Perpetual Futures</b> in USD Applicable to <b>Multi-currency margin</b> / <b>Portfolio margin</b>
> notionalUsdForFutures	String	Notional value of positions for <b>Expiry Futures</b> in USD Applicable to <b>Multi-currency margin</b> / <b>Portfolio margin</b>
> notionalUsdForOption	String	Notional value of positions for <b>Option</b> in USD Applicable to <b>Spot mode</b> / <b>Multi-currency margin</b> / <b>Portfolio margin</b>
> upl	String	Cross-margin info of unrealized profit and loss at the account level in <b>USD</b> Applicable to <b>Multi-currency margin</b> / <b>Portfolio margin</b>
> delta	String	Delta (USD)
> deltaLever	String	Delta neutral strategy account level delta leverage $\text{deltaLever} = \text{delta} / \text{totalEq}$
> deltaNeutralStatus	String	Delta risk status <input checked="" type="radio"/> 0: normal <input type="radio"/> 1: transfer restricted <input type="radio"/> 2: delta reducing - cancel all pending orders if delta is greater than 5000 USD, only one delta reducing order allowed per index (spot, futures, swap)
> details	Array of objects	Detailed asset information in all currencies
>> ccy	String	Currency
>> eq	String	Equity of currency
>> cashBal	String	Cash Balance
>> uTime	String	Update time, Unix timestamp format in milliseconds, e.g. <b>1597026383085</b>
>> isoEq	String	Isolated margin equity of currency Applicable to <b>Futures mode</b> / <b>Multi-currency margin</b> / <b>Portfolio margin</b>
>> availEq	String	Available equity of currency Applicable to <b>Futures mode</b> / <b>Multi-currency margin</b> / <b>Portfolio margin</b>
>> disEq	String	Discount equity of currency in <b>USD</b>
>> fixedBal	String	Frozen balance for <b>Dip Sniper</b> and <b>Peak Sniper</b>
>> availBal	String	Available balance of currency
>> frozenBal	String	Frozen balance of currency
>> ordFrozen	String	Margin frozen for open orders Applicable to <b>Spot mode</b> / <b>Futures mode</b> / <b>Multi-currency margin</b>
>> liab	String	Liabilities of currency It is a positive value, e.g. <b>21625.64</b> . Applicable to <b>Spot mode</b> / <b>Multi-currency margin</b> / <b>Portfolio margin</b>
>> upl	String	The sum of the unrealized profit & loss of all margin and derivatives positions of currency. Applicable to <b>Futures mode</b> / <b>Multi-currency margin</b> / <b>Portfolio margin</b>
>> uplLiab	String	Liabilities due to Unrealized loss of currency Applicable to <b>Multi-currency margin</b> / <b>Portfolio margin</b>

Parameters	Types	Description
>> crossLiab	String	Cross Liabilities of currency Applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
>> isoLiab	String	Isolated Liabilities of currency Applicable to <code>Multi-currency margin</code> / <code>Portfolio margin</code>
>> rewardBal	String	Trial fund balance
>> mgnRatio	String	Cross Maintenance margin ratio of currency The index for measuring the risk of a certain asset in the account. Applicable to <code>Futures mode</code> and when there is cross position
>> imr	String	Cross initial margin requirement at the currency level Applicable to <code>Futures mode</code> and when there is cross position
>> mmr	String	Cross maintenance margin requirement at the currency level Applicable to <code>Futures mode</code> and when there is cross position
>> interest	String	Interest of currency It is a positive value, e.g."9.01". Applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
>> twap	String	Risk indicator of forced repayment Divided into multiple levels from 0 to 5, the larger the number, the more likely the forced repayment will be triggered. Applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
>> frpType	String	Forced repayment (FRP) type 0: no FRP 1: user based FRP 2: platform based FRP  Return 1/2 when twap is >= 1, applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
>> maxLoan	String	Max loan of currency Applicable to <code>cross</code> of <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
>> eqUsd	String	Equity <code>USD</code> of currency
>> borrowFroz	String	Potential borrowing IMR of currency in <code>USD</code> Only applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code> . It is "" for other margin modes.
>> notionalLever	String	Leverage of currency Applicable to <code>Futures mode</code>
>> coinUsdPrice	String	Price index <code>USD</code> of currency
>> stgyEq	String	strategy equity
>> isoUpl	String	Isolated unrealized profit and loss of currency Applicable to <code>Futures mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
>> spotInUseAmt	String	Spot in use amount Applicable to <code>Portfolio margin</code>
>> clSpotInUseAmt	String	User-defined spot risk offset amount Applicable to <code>Portfolio margin</code>
>> maxSpotInUseAmt	String	Max possible spot risk offset amount Applicable to <code>Portfolio margin</code>

Parameters	Types	Description
>> spotIsoBal	String	Spot isolated balance Applicable to copy trading Applicable to <a href="#">Spot mode</a> / <a href="#">Futures mode</a>
>> smtSyncEq	String	Smart sync equity The default is "0", only applicable to copy trader.
>> spotCopyTradingEq	String	Spot smart sync equity. The default is "0", only applicable to copy trader.
>> spotBal	String	Spot balance. The unit is currency, e.g. BTC. More details
>> openAvgPx	String	Spot average cost price. The unit is USD. More details
>> accAvgPx	String	Spot accumulated cost price. The unit is USD. More details
>> spotUpL	String	Spot unrealized profit and loss. The unit is USD. More details
>> spotUpLRatio	String	Spot unrealized profit and loss ratio. More details
>> totalPnl	String	Spot accumulated profit and loss. The unit is USD. More details
>> totalPnlRatio	String	Spot accumulated profit and loss ratio. More details
>> colRes	String	Platform level collateral restriction status 0: The restriction is not enabled. 1: The restriction is not enabled. But the crypto is close to the platform's collateral limit. 2: The restriction is enabled. This crypto can't be used as margin for your new orders. This may result in failed orders. But it will still be included in the account's adjusted equity and doesn't impact margin ratio. Refer to <a href="#">Introduction to the platform collateralized borrowing limit</a> for more details.
>> colBorrAutoConversion	String	Risk indicator of auto conversion. Divided into multiple levels from 1-5, the larger the number, the more likely the repayment will be triggered. The default will be 0, indicating there is no risk currently. 5 means this user is undergoing auto conversion now, 4 means this user will undergo auto conversion soon whereas 1/2/3 indicates there is a risk for auto conversion. Applicable to <a href="#">Spot mode</a> / <a href="#">Futures mode</a> / <a href="#">Multi-currency margin</a> / <a href="#">Portfolio margin</a> When the total liability for each crypto set as collateral exceeds a certain percentage of the platform's total limit, the auto-conversion mechanism may be triggered. This may result in the automatic sale of excess collateral crypto if you've set this crypto as collateral and have large borrowings. To lower this risk, consider reducing your use of the crypto as collateral or reducing your liabilities. Refer to <a href="#">Introduction to the platform collateralized borrowing limit</a> for more details.
>> collateralRestrict	Boolean	<a href="#">Platform level collateralized borrow restriction</a> <a href="#">true</a> <a href="#">false</a> (deprecated, use colRes instead)
>> collateralEnabled	Boolean	<a href="#">true</a> : Collateral enabled <a href="#">false</a> : Collateral disabled Applicable to <a href="#">Multi-currency margin</a>

"" will be returned for inapplicable fields under the current account level.

- The account data is sent on event basis and regular basis.
- The event push is not pushed in real-time. It is aggregated and pushed at a fixed time interval, around 50ms. For example, if multiple events occur within a fixed time interval, the system will aggregate them into a single message and push it at the end of the fixed time interval. If the data volume is too large, it may be split into multiple messages.
- The regular push sends updates regardless of whether there are activities in the trading account or not.

- Only currencies with non-zero balance will be pushed. Definition of non-zero balance: any value of eq, availEq, availBql parameters is not 0. If the data is too large to be sent in a single push message, it will be split into multiple messages.
- For example, when subscribing to account channel without specifying ccy and there are 5 currencies are with non-zero balance, all 5 currencies data will be pushed in initial snapshot and in regular update. Subsequently when there is change in balance or equity of an token, only the incremental data of that currency will be pushed triggered by this change.

## POSITIONS CHANNEL

Retrieve position information. Initial snapshot will be pushed according to subscription granularity. Data will be pushed when triggered by events such as placing/canceling order, and will also be pushed in regular interval according to subscription granularity.

Concurrent connection to this channel will be restricted by the following rules: WebSocket connection count limit.

### URL PATH

/ws/v5/private (required login)

Request Example : single

```
import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/private",
        useServerTime=False
    )
    await ws.start()
    args = [
        {
            "channel": "positions",
            "instType": "FUTURES",
            "instFamily": "BTC-USD"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

Request Example

```
import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/private",
        useServerTime=False
    )
    await ws.start()
    args = [
        {
            "channel": "positions",
            "instType": "FUTURES",
            "instFamily": "BTC-USD"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

```

url = "wss://ws.okx.com:8443/ws/v5/private",
useServerTime=False
)
await ws.start()
args = [
{
    "channel": "positions",
    "instType": "ANY",
    "extraParams": "{\"updateInterval\": \"0\"}"
}
]
await ws.subscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

await ws.unsubscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation subscribe unsubscribe
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name positions
> instType	String	Yes	Instrument type MARGIN SWAP FUTURES OPTION ANY
> instFamily	String	No	Instrument family Applicable to FUTURES/SWAP/OPTION
> instId	String	No	Instrument ID If instId and instFamily are both passed, instId will be used
> extraParams	String	No	Additional configuration
>> updateInterval	int	No	<p>0: only push due to positions events 2000, 3000, 4000: push by events and regularly according to the time interval setting (ms)</p> <p>The data will be pushed both by events and around per 5 seconds regularly if this field is omitted or set to other values than the valid values above.</p> <p>The following format should be strictly followed when using this field.</p> <pre>"extraParams": {     "updateInterval": "0" }</pre>

Successful Response Example : single

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "positions",
    "instType": "FUTURES",
    "instFamily": "BTC-USD"
  },
  "connId": "a4d3ae55"
}
```

### Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "positions",
    "instType": "ANY"
  },
  "connId": "a4d3ae55"
}
```

### Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\": \"positions\", \"instType\" : \"FUTURES\"}]}",
  "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event   
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> instType	String	Yes	Instrument type     
> instFamily	String	No	Instrument family
> instId	String	No	Instrument ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

```

"arg": {
    "channel": "positions",
    "uid": "77982378738415879",
    "instType": "FUTURES"
},
"eventType": "snapshot",
"curPage": 1,
"lastPage": true,
"data": [
{
    "adl": "1",
    "availPos": "1",
    "avgPx": "2566.31",
    "cTime": "1619507758793",
    "ccy": "ETH",
    "deltaBS": "",
    "deltaPA": "",
    "gammaBS": "",
    "gammaPA": "",
    "hedgedPos": "",
    "imr": "",
    "instId": "ETH-USD-210430",
    "instType": "FUTURES",
    "interest": "0",
    "idxPx": "2566.13",
    "last": "2566.22",
    "lever": "10",
    "liab": "",
    "liabCcy": "",
    "liqPx": "2352.8496681818233",
    "markPx": "2353.849",
    "margin": "0.0003896645377994",
    "mgnMode": "isolated",
    "mgnRatio": "11.731726509588816",
    "mmr": "0.0000311811092368",
    "notionalUsd": "2276.2546609009605",
    "optVal": "",
    "pTime": "1619507761462",
    "pendingCloseOrdliabVal": "0.1",
    "pos": "1",
    "baseBorrowed": "",
    "baseInterest": "",
    "quoteBorrowed": "",
    "quoteInterest": "",
    "posCcy": "",
    "posId": "307173036051017730",
    "posSide": "long",
    "spotInUseAmt": "",
    "clSpotInUseAmt": "",
    "maxSpotInUseAmt": "",
    "bizRefId": "",
    "bizRefType": "",
    "spotInUseCcy": "",
    "thetaBS": "",
    "thetaPA": "",
    "tradeId": "109844",
    "uTime": "1619507761462",
    "upl": "-0.0000009932766034",
    "uplLastPx": "-0.0000009932766034",
    "uplRatio": "-0.0025490556801078",
    "uplRatioLastPx": "-0.0025490556801078",
    "vegaBS": "",
    "vegaPA": "",
    "realizedPnl": "0.001",
    "pnl": "0.0011",
    "fee": "-0.0001",
    "fundingFee": "0",
    "liqPenalty": "0",
    "nonSettleAvgPx": "",
    "settledPnl": "",
    "closeOrderAlgo": [
{
    "algoId": "123",
    "slTriggerPx": "123",
    "slTriggerPxType": "mark",
    "trgPx": "123",
    "trgPxType": "mark"
}
]
}

```

```

        "tpTriggerPxType": "mark",
        "closeFraction": "0.6"
    },
    {
        "algoId": "123",
        "slTriggerPx": "123",
        "slTriggerPxType": "mark",
        "tpTriggerPx": "123",
        "tpTriggerPxType": "mark",
        "closeFraction": "0.4"
    }
]
}
]
}
}

```

## Push Data Example

```

{
    "arg": {
        "channel": "positions",
        "uid": "77982378738415879",
        "instType": "ANY"
    },
    "eventType": "snapshot",
    "curPage": 1,
    "lastPage": true,
    "data": [
        {
            "ad1": "1",
            "availPos": "1",
            "avgPx": "2566.31",
            "cTime": "1619507758793",
            "ccy": "ETH",
            "deltaBS": "",
            "deltaPA": "",
            "gammaBS": "",
            "gammaPA": "",
            "hedgedPos": "",
            "imr": "",
            "instId": "ETH-USD-210430",
            "instType": "FUTURES",
            "interest": "0",
            "idxPx": "2566.13",
            "last": "2566.22",
            "usdPx": "",
            "bePx": "2353.949",
            "lever": "10",
            "liab": "",
            "liabCcy": "",
            "liqPx": "2352.8496681818233",
            "markPx": "2353.849",
            "margin": "0.0003896645377994",
            "mgnMode": "isolated",
            "mgnRatio": "11.731726509588816",
            "mmr": "0.0000311811092368",
            "notionalUsd": "2276.2546609009605",
            "optVal": "",
            "pTime": "1619507761462",
            "pendingCloseOrdLiabVal": "0.1",
            "pos": "1",
            "baseBorrowed": "",
            "baseInterest": "",
            "quoteBorrowed": "",
            "quoteInterest": "",
            "posCcy": "",
            "posId": "307173036051017730",
            "posSide": "long",
            "spotInUseAmt": "",
            "clSpotInUseAmt": "",
            "maxSpotInUseAmt": "",
            "spotInUseCcy": "",
            "bizRefId": "",
            "bizRefType": "",
            "thetaBS": "",
            "thetaPA": ""
        }
    ]
}
}
}
}

```

```
"tradeId": "109844",
"uTime": "1619507761462",
"upl": "-0.0000009932766034",
"uplLastPx": "-0.0000009932766034",
"uplRatio": "-0.0025490556801078",
"uplRatioLastPx": "-0.0025490556801078",
"vegaBS": "",
"vegaPA": "",
"realizedPnl": "0.001",
"pnl": "0.0011",
"fee": "0.0001",
"fundingFee": "0",
"liqPenalty": "0",
"nonSettleAvgPx": "",
"settledPnl": "",
"closeOrderAlgo": [
  {
    "algoId": "123",
    "slTriggerPx": "123",
    "slTriggerPxType": "mark",
    "tpTriggerPx": "123",
    "tpTriggerPxType": "mark",
    "closeFraction": "0.6"
  },
  {
    "algoId": "123",
    "slTriggerPx": "123",
    "slTriggerPxType": "mark",
    "tpTriggerPx": "123",
    "tpTriggerPxType": "mark",
    "closeFraction": "0.4"
  }
],
},
{
  "adl": "1",
  "availPos": "1",
  "avgPx": "2566.31",
  "cTime": "1619507758793",
  "ccy": "ETH",
  "deltaBS": "",
  "deltaPA": "",
  "gammaBS": "",
  "gammaPA": "",
  "hedgedPos": "",
  "imr": "",
  "instId": "ETH-USD-SWAP",
  "instType": "SWAP",
  "interest": "0",
  "idxPx": "2566.13",
  "last": "2566.22",
  "usdPx": "",
  "bePx": "2353.949",
  "lever": "10",
  "liab": "",
  "liabCcy": "",
  "liqPx": "2352.8496681818233",
  "markPx": "2353.849",
  "margin": "0.0003896645377994",
  "mgnMode": "isolated",
  "mgnRatio": "11.731726509588816",
  "mmr": "0.0000311811092368",
  "notionalUsd": "2276.2546609009605",
  "optVal": "",
  "pTime": "1619507761462",
  "pendingCloseOrdLiabVal": "0.1",
  "pos": "1",
  "baseBorrowed": "",
  "baseInterest": "",
  "quoteBorrowed": "",
  "quoteInterest": "",
  "posCcy": "",
  "posId": "307173036051017730",
  "posSide": "long",
  "spotInUseAmt": "",
  "clSpotInUseAmt": "",
  "maxSpotInUseAmt": "",
  "spotInUseCcy": ""
}, {
```

```
"bizRefId": "",  
"bizRefType": "",  
"thetaB5": "",  
"thetaPA": "",  
"tradeId": "109844",  
"uTime": "1619507761462",  
"upl": "-0.0000009932766034",  
"uplLastPx": "-0.0000009932766034",  
"uplRatio": "-0.0025490556801078",  
"uplRatioLastPx": "-0.0025490556801078",  
"vegaBS": "",  
"vegaPA": "",  
"realizedPnl": "0.001",  
"pnl": "0.0011",  
"fee": "-0.0001",  
"fundingFee": "0",  
"liqPenalty": "0",  
"nonSettleAvgPx": "",  
"settledPnl": "",  
"closeOrderAlgo": [  
    {  
        "algoId": "123",  
        "slTriggerPx": "123",  
        "slTriggerPxType": "mark",  
        "tpTriggerPx": "123",  
        "tpTriggerPxType": "mark",  
        "closeFraction": "0.6"  
    },  
    {  
        "algoId": "123",  
        "slTriggerPx": "123",  
        "slTriggerPxType": "mark",  
        "tpTriggerPx": "123",  
        "tpTriggerPxType": "mark",  
        "closeFraction": "0.4"  
    }  
]
```

## PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> uid	String	User Identifier
> instType	String	Instrument type
> instFamily	String	Instrument family
> instId	String	Instrument ID
eventType	String	Event type: <code>snapshot</code> : Initial and regular snapshot push <code>event_update</code> : Event-driven update push
curPage	Integer	Current page number. Only applicable for <code>snapshot</code> events. Not included in <code>event_update</code> events.
lastPage	Boolean	Whether this is the last page of pagination: <code>true</code> <code>false</code> Only applicable for <code>snapshot</code> events. Not included in <code>event_update</code> events.
data	Array of objects	Subscribed data

Parameter	Type	Description
> instType	String	Instrument type
> mgnMode	String	Margin mode, <code>cross</code> <code>isolated</code>
> posId	String	Position ID
> posSide	String	Position side <code>long</code> <code>short</code> <code>net</code> ( <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code> : positive <code>pos</code> means long position and negative <code>pos</code> means short position. <code>MARGIN</code> : <code>posCcy</code> being base currency means long position, <code>posCcy</code> being quote currency means short position.)
> pos	String	Quantity of positions. In the isolated margin mode, when doing manual transfers, a position with pos of <code>0</code> will be generated after the deposit is transferred
> hedgedPos	String	Hedged position size Only return for accounts in delta neutral strategy, stgyType:1. Return "" for accounts in general strategy.
> baseBal	String	Base currency balance, only applicable to <code>MARGIN</code> (Quick Margin Mode) (Deprecated)
> quoteBal	String	Quote currency balance, only applicable to <code>MARGIN</code> (Quick Margin Mode) (Deprecated)
> baseBorrowed	String	Base currency amount already borrowed, only applicable to <code>MARGIN</code> (Quick Margin Mode) (Deprecated)
> baseInterest	String	Base interest, undeducted interest that has been incurred, only applicable to <code>MARGIN</code> (Quick Margin Mode) (Deprecated)
> quoteBorrowed	String	Quote currency amount already borrowed, only applicable to <code>MARGIN</code> (Quick Margin Mode) (Deprecated)
> quoteInterest	String	Quote interest, undeducted interest that has been incurred, only applicable to <code>MARGIN</code> (Quick Margin Mode) (Deprecated)
> posCcy	String	Position currency, only applicable to <code>MARGIN</code> positions
> availPos	String	Position that can be closed Only applicable to <code>MARGIN</code> and <code>OPTION</code> . For <code>Margin</code> position, the rest of sz will be <code>SPOT</code> trading after the liability is repaid while closing the position. Please get the available reduce-only amount from "Get maximum available tradable amount" if you want to reduce the amount of <code>SPOT</code> trading as much as possible.
> avgPx	String	Average open price
> upl	String	Unrealized profit and loss calculated by mark price.
> uplRatio	String	Unrealized profit and loss ratio calculated by mark price.
> uplLastPx	String	Unrealized profit and loss calculated by last price. Main usage is showing, actual value is upl.
> uplRatioLastPx	String	Unrealized profit and loss ratio calculated by last price.
> instId	String	Instrument ID, e.g. <code>BTC-USDT-SWAP</code>
> lever	String	Leverage, not applicable to <code>OPTION</code> seller
> liqPx	String	Estimated liquidation price Not applicable to <code>OPTION</code>
> markPx	String	Latest Mark price
> imr	String	Initial margin requirement, only applicable to <code>cross</code>

Parameter	Type	Description
> margin	String	Margin, can be added or reduced. Only applicable to <a href="#">isolated Margin</a> .
> mgnRatio	String	Maintenance margin ratio
> mmr	String	Maintenance margin requirement
> liab	String	Liabilities, only applicable to <a href="#">MARGIN</a> .
> liabCcy	String	Liabilities currency, only applicable to <a href="#">MARGIN</a> .
> interest	String	Interest accrued that has not been settled.
> tradeld	String	Last trade ID
> notionalUsd	String	Notional value of positions in <a href="#">USD</a>
> optVal	String	Option Value, only applicable to <a href="#">OPTION</a> .
> pendingCloseOrdLiabVal	String	The amount of close orders of isolated margin liability.
> adl	String	Automatic-Deleveraging, signal area Divided into 6 levels, from 0 to 5, the smaller the number, the weaker the adl intensity. Only applicable to <a href="#">FUTURES/SWAP/OPTION</a>
> bizRefId	String	External business id, e.g. experience coupon id
> bizRefType	String	External business type
> ccy	String	Currency used for margin
> last	String	Latest traded price
> idxPx	String	Latest underlying index price
> usdPx	String	Latest USD price of the <a href="#">ccy</a> on the market, only applicable to <a href="#">FUTURES/SWAP/OPTION</a>
> bePx	String	Breakeven price
> deltaBS	String	delta: Black-Scholes Greeks in dollars, only applicable to <a href="#">OPTION</a>
> deltaPA	String	delta: Greeks in coins, only applicable to <a href="#">OPTION</a>
> gammaBS	String	gamma: Black-Scholes Greeks in dollars, only applicable to <a href="#">OPTION</a>
> gammaPA	String	gamma: Greeks in coins, only applicable to <a href="#">OPTION</a>
> thetaBS	String	theta: Black-Scholes Greeks in dollars, only applicable to <a href="#">OPTION</a>
> thetaPA	String	theta: Greeks in coins, only applicable to <a href="#">OPTION</a>
> vegaBS	String	vega: Black-Scholes Greeks in dollars, only applicable to <a href="#">OPTION</a>
> vegaPA	String	vega: Greeks in coins, only applicable to <a href="#">OPTION</a>
> spotInUseAmt	String	Spot in use amount Applicable to <a href="#">Portfolio margin</a>
> spotInUseCcy	String	Spot in use unit, e.g. <a href="#">BTC</a> Applicable to <a href="#">Portfolio margin</a>

Parameter	Type	Description
> clSpotInUseAmt	String	User-defined spot risk offset amount Applicable to <b>Portfolio margin</b>
> maxSpotInUseAmt	String	Max possible spot risk offset amount Applicable to <b>Portfolio margin</b>
> realizedPnl	String	Realized profit and loss Only applicable to <b>FUTURES</b> / <b>SWAP</b> / <b>OPTION</b> realizedPnl=pnl+fee+fundingFee+liqPenalty+settledPnl
> pnl	String	Accumulated pnl of closing order(s) (excluding the fee).
> fee	String	Accumulated fee Negative number represents the user transaction fee charged by the platform. Positive number represents rebate.
> fundingFee	String	Accumulated funding fee
> liqPenalty	String	Accumulated liquidation penalty. It is negative when there is a value.
> closeOrderAlgo	Array of objects	Close position algo orders attached to the position. This array will have values only after you request "Place algo order" with <b>closeFraction</b> =1.
>> algold	String	Algo ID
>> slTriggerPx	String	Stop-loss trigger price.
>> slTriggerPxType	String	Stop-loss trigger price type. <b>last</b> : last price <b>index</b> : index price <b>mark</b> : mark price
>> tpTriggerPx	String	Take-profit trigger price.
>> tpTriggerPxType	String	Take-profit trigger price type. <b>last</b> : last price <b>index</b> : index price <b>mark</b> : mark price
>> closeFraction	String	Fraction of position to be closed when the algo order is triggered.
> cTime	String	Creation time, Unix timestamp format in milliseconds, e.g. <b>1597026383085</b> .
> uTime	String	Latest time position was adjusted, Unix timestamp format in milliseconds, e.g. <b>1597026383085</b> .
> pTime	String	Push time of positions information, Unix timestamp format in milliseconds, e.g. <b>1597026383085</b> .
> nonSettleAvgPx	String	Non-Settlement entry price The non-settlement entry price only reflects the average price at which the position is opened or increased. Applicable to <b>FUTURES</b> <b>cross</b>
> settledPnl	String	Accumulated settled P&L (calculated by settlement price) Applicable to <b>FUTURES</b> <b>cross</b>

- The position data is sent on event basis and regular basis

- The event push is not pushed in real-time. It is aggregated and pushed at a fixed time interval, around 50ms. For example, if multiple events occur within a fixed time interval, the system will aggregate them into a single message and push it at the end of the fixed time interval. If the data volume is too large, it may be split into multiple messages.

- The regular push sends updates regardless of whether there are position activities or not.
- If an event push and a regular push happen at the same time, the system will send the event push first, followed by the regular push.

As for portfolio margin account, the IMR and MMR of the position are calculated in risk unit granularity, thus their values of the same risk unit cross positions are the same.

In the position-by-position trading setting, it is an autonomous transfer mode. After the margin is transferred, positions with a position of 0 will be pushed

- Only position with non-zero position quantity will be pushed. Definition of non-zero quantity: value of pos parameter is not 0. If the data is too large to be sent in a single push message, it will be split into multiple messages.
- For example, when subscribing to positions channel specifying an underlying and there are 20 positions are with non-zero quantity, all 20 positions data will be pushed in initial snapshot and in regular push. Subsequently when there is change in pos of a position, only the data of that position will be pushed triggered by this change.

## BALANCE AND POSITION CHANNEL

Retrieve account balance and position information. Data will be pushed when triggered by events such as filled order, funding transfer. This channel applies to getting the account cash balance and the change of position asset ASAP. Concurrent connection to this channel will be restricted by the following rules: WebSocket connection count limit.

### URL PATH

/ws/v5/private (required login)

#### Request Example

```
import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/private",
        useServerTime=False
    )
    await ws.start()
    args = [{
        "channel": "balance_and_position"
    }]

    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>balance_and_position</code>

### Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "balance_and_position"
  },
  "connId": "a4d3ae55"
}
```

### Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\"op\": \"subscribe\", \"args\": [{ \"channel\": \"balance_and_position\"}]}",
  "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code> <code>error</code>
arg	Object	No	List of subscribed channels
> channel	String	Yes	Channel name <code>balance_and_position</code>
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

### Push Data Example

```
{
  "arg": {
    "channel": "balance_and_position",
    "uid": "77982378738415879"
  },
}
```

```

"data": [
  "pTime": "1597026383085",
  "eventType": "snapshot",
  "balData": [
    {
      "ccy": "BTC",
      "cashBal": "1",
      "uTime": "1597026383085"
    }
  ],
  "posData": [
    {
      "posId": "1111111111",
      "tradeId": "2",
      "instId": "BTC-USD-191018",
      "instType": "FUTURES",
      "mgnMode": "cross",
      "posSide": "long",
      "pos": "10",
      "ccy": "BTC",
      "posCcy": "",
      "avgPx": "3320",
      "nonSettleAvgPx": "",
      "settledPnl": "",
      "uTime": "1597026383085"
    }
  ],
  "trades": [
    {
      "instId": "BTC-USD-191018",
      "tradeId": "2",
    }
  ]
}

```

#### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Channel to subscribe to
> channel	String	Channel name
> uid	String	User Identifier
data	Array of objects	Subscribed data
> pTime	String	Push time of both balance and position information, millisecond format of Unix timestamp, e.g. <code>1597026383085</code>
> eventType	String	Event Type <div style="display: flex; flex-wrap: wrap;"> <span>snapshot</span> <span>delivered</span> <span>exercised</span> <span>transferred</span> <span>filled</span> <span>liquidation</span> <span>claw_back</span> <span>adl</span> <span>funding_fee</span> <span>adjust_margin</span> <span>set_leverage</span> <span>interest_deduction</span> <span>settlement</span> </div>
> balData	Array of objects	Balance data
>> ccy	String	Currency
>> cashBal	String	Cash Balance
>> uTime	String	Update time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

Parameter	Type	Description
> posData	Array of objects	Position data
>> posId	String	Position ID
>> tradeId	String	Last trade ID
>> instId	String	Instrument ID, e.g. <code>BTC-USD-180213</code>
>> instType	String	Instrument type
>> mgnMode	String	Margin mode <code>isolated</code> , <code>cross</code>
>> avgPx	String	Average open price
>> ccy	String	Currency used for margin
>> posSide	String	Position side <code>long</code> , <code>short</code> , <code>net</code>
>> pos	String	Quantity of positions. In the isolated margin mode, when doing manual transfers, a position with pos of <code>0</code> will be generated after the deposit is transferred
>> baseBal	String	Base currency balance, only applicable to <code>MARGIN</code> (Quick Margin Mode) (Deprecated)
>> quoteBal	String	Quote currency balance, only applicable to <code>MARGIN</code> (Quick Margin Mode) (Deprecated)
>> posCcy	String	Position currency, only applicable to MARGIN positions.
>> nonSettleAvgPx	String	Non-Settlement entry price The non-settlement entry price only reflects the average price at which the position is opened or increased. Applicable to <code>FUTURES</code> <code>cross</code>
>> settledPnl	String	Accumulated settled P&L (calculated by settlement price) Applicable to <code>FUTURES</code> <code>cross</code>
>> uTime	String	Update time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> trades	Array of objects	Details of trade
>> instId	String	Instrument ID, e.g. <code>BTC-USDT</code>
>> tradeId	String	Trade ID

Only balData will be pushed if only the account balance changes; only posData will be pushed if only the position changes.

- Initial snapshot: Only either position with non-zero position quantity or cash balance with non-zero quantity will be pushed. If the data is too large to be sent in a single push message, it will be split into multiple messages.
- For example, if you subscribe according to all currencies and the user has 5 currency balances that are not 0 and 20 positions, all 20 positions data and 5 currency balances data will be pushed in initial snapshot; Subsequently when there is change in pos of a position, only the data of that position will be pushed triggered by this change.

#### POSITION RISK WARNING

This push channel is only used as a risk warning, and is not recommended as a risk judgment for strategic trading

In the case that the market is volatile, there may be the possibility that the position has been liquidated at the same time that this message is pushed.

The warning is sent when a position is at risk of liquidation for isolated margin positions. The warning is sent when all the positions are at risk of liquidation for cross-margin positions.

Concurrent connection to this channel will be restricted by the following rules: WebSocket connection count limit.

## URL PATH

/ws/v5/private (required login)

### Request Example

```
import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/private",
        useServerTime=False
    )
    await ws.start()
    args = [
        {
            "channel": "liquidation-warning",
            "instType": "ANY"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>liquidation-warning</code>
> instType	String	Yes	Instrument type <code>MARGIN</code> <code>SWAP</code> <code>FUTURES</code> <code>OPTION</code> <code>ANY</code>

Parameter	Type	Required	Description
> instFamily	String	No	Instrument family Applicable to <b>FUTURES</b> / <b>SWAP</b> / <b>OPTION</b>
> instId	String	No	Instrument ID

#### Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "liquidation-warning",
    "instType": "ANY"
  },
  "connId": "a4d3ae55"
}
```

#### Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\": \"liquidation-warning\", \"instType\": \"FUTURES\"}]}",
  "connId": "a4d3ae55"
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event <b>subscribe</b> <b>unsubscribe</b> <b>error</b>
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name <b>liquidation-warning</b>
> instType	String	Yes	Instrument type <b>OPTION</b> <b>FUTURES</b> <b>SWAP</b> <b>MARGIN</b> <b>ANY</b>
> instFamily	String	No	Instrument family
> instId	String	No	Instrument ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

#### Push Data Example

```
{
  "arg": {
    "channel": "liquidation-warning",
    "uid": "77982378738415879",
    "instType": "FUTURES"
  },
  "data": [
    {
      "cTime": "1619507758793",
      "ccy": "ETH",
      "instId": "ETH-USD-210430",
      "instType": "FUTURES",
      "lever": "10",
      "markPx": "2353.849",
      "mgnMode": "isolated",
      "mgnRatio": "11.731726509588816",
      "pTime": "1619507761462",
      "pos": "1",
      "posCcy": "",
      "posId": "307173036051017730",
      "posSide": "long",
      "uTime": "1619507761462",
    }
  ]
}
```

#### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> uid	String	User Identifier
> instType	String	Instrument type
> instFamily	String	Instrument family
> instId	String	Instrument ID
data	Array of objects	Subscribed data
> instType	String	Instrument type
> mgnMode	String	Margin mode, <code>cross</code> <code>isolated</code>
> posId	String	Position ID
> posSide	String	Position side <code>long</code> <code>short</code> <code>net</code> ( <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code> ): positive <code>pos</code> means long position and negative <code>pos</code> means short position. ( <code>MARGIN</code> : <code>pos</code> <code>ccy</code> being base currency means long position, <code>posCcy</code> being quote currency means short position.)
> pos	String	Quantity of positions
> posCcy	String	Position currency, only applicable to <code>MARGIN</code> positions
> instId	String	Instrument ID, e.g. <code>BTC-USDT-SWAP</code>
> lever	String	Leverage, not applicable to <code>OPTION</code> seller
> markPx	String	Mark price

Parameter	Type	Description
> mgnRatio	String	Maintenance margin ratio
> ccy	String	Currency used for margin
> cTime	String	Creation time, Unix timestamp format in milliseconds, e.g. 1597026383085.
> uTime	String	Latest time position was adjusted, Unix timestamp format in milliseconds, e.g. 1597026383085.
> pTime	String	Push time of positions information, Unix timestamp format in milliseconds, e.g. 1597026383085.

Trigger push logic: the trigger logic of the liquidation warning and the liquidation message is the same

#### ACCOUNT GREEKS CHANNEL

Retrieve account greeks information. Data will be pushed when triggered by events such as increase/decrease positions or cash balance in account, and will also be pushed in regular interval according to subscription granularity.

Concurrent connection to this channel will be restricted by the following rules: WebSocket connection count limit.

#### URL PATH

/ws/v5/private (required login)

Request Example

```
import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/private",
        useServerTime=False
    )
    await ws.start()
    args = [{
        "channel": "account-greeks"
    }]

    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.

Parameter	Type	Required	Description
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>account-greeks</code>
> ccy	String	No	Settlement currency When the user specifies a settlement currency, event push will only be triggered when the position of the same settlement currency changes. For example, when ccy=BTC, if the position of <code>BTC-USDT-SWAP</code> changes, no event push will be triggered.

#### Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "account-greeks"
  },
  "connId": "a4d3ae55"
}
```

#### Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \\"subscribe\\", \\"args\\": [{ \\"channel\\": \\"account-greeks\\\", \\"ccy\\": \\"BTC\\\" }]}",
  "connId": "a4d3ae55"
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code> <code>error</code>
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name, <code>account-greeks</code>
> ccy	String	No	Settlement currency
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

#### Push Data Example: single

```
{
  "arg": {
    "channel": "account-greeks",
    "ccy": "BTC"
  }
}
```

```

"ccy": "BTC",
"uid": "614488474791936"
},
"data": [
{
  "ccy": "BTC",
  "deltaBS": "1.1246665401944310",
  "deltaPA": "-0.0074076183688949",
  "gammaBS": "0.0000000000000000",
  "gammaPA": "0.0148152367377899",
  "thetaBS": "2.0356991946421226",
  "thetaPA": "-0.0000000200174309",
  "ts": "1729179082006",
  "vegaBS": "0.0000000000000000",
  "vegaPA": "0.0000000000000000"
}
]
}

```

## Push Data Example

```

{
  "arg": {
    "channel": "account-greeks",
    "uid": "614488474791936"
  },
  "data": [
    {
      "ccy": "BTC",
      "deltaBS": "1.1246665403011684",
      "deltaPA": "-0.0074021163991037",
      "gammaBS": "0.0000000000000000",
      "gammaPA": "0.0148042327982075",
      "thetaBS": "2.1342098201092528",
      "thetaPA": "-0.0000000200876441",
      "ts": "1729179001692",
      "vegaBS": "0.0000000000000000",
      "vegaPA": "0.0000000000000000"
    },
    {
      "ccy": "ETH",
      "deltaBS": "0.3810670161698570",
      "deltaPA": "-0.0688347042402955",
      "gammaBS": "-0.000000000230396",
      "gammaPA": "0.1376693483440320",
      "thetaBS": "0.3314776517141782",
      "thetaPA": "0.000000001316008",
      "ts": "1729179001692",
      "vegaBS": "-0.000000045069794",
      "vegaPA": "-0.00000000017267"
    }
  ]
}

```

## PUSH DATA PARAMETERS

Parameters	Types	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> uid	String	User Identifier
data	Array of objects	Subscribed data
> deltaBS	String	delta: Black-Scholes Greeks in dollars
> deltaPA	String	delta: Greeks in coins
> gammaBS	String	gamma: Black-Scholes Greeks in dollars, only applicable to OPTION cross

Parameters	Types	Description
> gammaPA	String	gamma: Greeks in coins, only applicable to OPTION cross
> thetaBS	String	theta: Black-Scholes Greeks in dollars, only applicable to OPTION cross
> thetaPA	String	theta: Greeks in coins, only applicable to OPTION cross
> vegaBS	String	vega: Black-Scholes Greeks in dollars, only applicable to OPTION cross
> vegaPA	String	vega: Greeks in coins, only applicable to OPTION cross
> ccy	String	Currency
> ts	String	Push time of account greeks, Unix timestamp format in milliseconds, e.g. 1597026383085

The account greeks data is sent on event basis and regular basis

- The event push is not pushed in real-time. It is aggregated and pushed at a fixed time interval, around 50ms. For example, if multiple events occur within a fixed time interval, the system will aggregate them into a single message and push it at the end of the fixed time interval. If the data volume is too large, it may be split into multiple messages.
- When the user specifies a settlement currency in the subscribe request, event push will only be triggered when the position of the same settlement currency changes. For example, when subscribe `ccy`=BTC, if the position of `BTC-USDT-SWAP` changes, no event push will be triggered.
- The regular push sends updates regardless of whether there are activities or not.

- Only currencies in the account will be pushed. If the data is too large to be sent in a single push message, it will be split into multiple messages.
- For example, when subscribing to account-greeks channel without specifying ccy and there are 5 currencies are with non-zero balance, all 5 currencies data will be pushed in initial snapshot and in regular interval. Subsequently when there is change in balance or equity of an token, only the incremental data of that currency will be pushed triggered by this change.

## Order Book Trading

### Trade

All `Trade` API endpoints require authentication.

#### POST / PLACE ORDER

You can place an order only if you have sufficient funds.

#### RATE LIMIT: 60 REQUESTS PER 2 SECONDS

#### RATE LIMIT OF LEAD TRADER LEAD INSTRUMENTS FOR COPY TRADING: 4 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE (EXCEPT OPTIONS): USER ID + INSTRUMENT ID

#### RATE LIMIT RULE (OPTIONS ONLY): USER ID + INSTRUMENT FAMILY

#### PERMISSION: TRADE

Rate limit of this endpoint will also be affected by the rules Sub-account rate limit and Fill ratio based sub-account rate limit.

#### HTTP REQUEST

`POST /api/v5/trade/order`

Request Example

```

import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Spot mode, limit order
result = tradeAPI.place_order(
    instId="BTC-USDT",
    tdMode="cash",
    clOrdId="b15",
    side="buy",
    ordType="limit",
    px="2.15",
    sz="2"
)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT</code>
tdMode	String	Yes	<p>Trade mode  Margin mode <code>cross</code> <code>isolated</code>  Non-Margin mode <code>cash</code>  <code>spot_isolated</code> (only applicable to SPOT lead trading, <code>tdMode</code> should be <code>spot_isolated</code> for SPOT lead trading.)</p> <p>Note: <code>isolated</code> is not available in multi-currency margin mode and portfolio margin mode.</p>
ccy	String	No	<p>Margin currency  Applicable to all <code>isolated</code> <code>MARGIN</code> orders and <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code>.</p>
clOrdId	String	No	<p>Client Order ID as assigned by the client  A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.  Only applicable to general order. It will not be posted to algold when placing TP/SL order after the general order is filled completely.</p>
tag	String	No	<p>Order tag  A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters.</p>
side	String	Yes	Order side, <code>buy</code> <code>sell</code>
posSide	String	Conditional	<p>Position side  The default is <code>net</code> in the <code>net</code> mode  It is required in the <code>long/short</code> mode, and can only be <code>long</code> or <code>short</code>.  Only applicable to <code>FUTURES</code>/<code>SWAP</code>.</p>
ordType	String	Yes	<p>Order type  <code>market</code>: Market order, only applicable to <code>SPOT/MARGIN/FUTURES/SWAP</code>  <code>limit</code>: Limit order  <code>post_only</code>: Post-only order  <code>fok</code>: Fill-or-kill order  <code>ioc</code>: Immediate-or-cancel order  <code>optimal_limit_ioc</code>: Market order with immediate-or-cancel order (applicable only to Expiry Futures and Perpetual Futures).  <code>mmp</code>: Market Maker Protection (only applicable to Option in Portfolio Margin mode)  <code>mmp_and_post_only</code>: Market Maker Protection and Post-only order(only applicable to Option in Portfolio Margin mode)</p>

Parameter	Type	Required	Description
sz	String	Yes	Quantity to buy or sell
px	String	Conditional	Order price. Only applicable to <code>limit</code> , <code>post_only</code> , <code>fok</code> , <code>ioc</code> , <code>mmp</code> , <code>mmp_and_post_only</code> order. When placing an option order, one of px/pxUsd/pxVol must be filled in, and only one can be filled in
pxUsd	String	Conditional	Place options orders in <code>USD</code> Only applicable to options When placing an option order, one of px/pxUsd/pxVol must be filled in, and only one can be filled in
pxVol	String	Conditional	Place options orders based on implied volatility, where 1 represents 100% Only applicable to options When placing an option order, one of px/pxUsd/pxVol must be filled in, and only one can be filled in
reduceOnly	Boolean	No	Whether orders can only reduce in position size. Valid options: <code>true</code> or <code>false</code> . The default value is <code>false</code> . Only applicable to <code>MARGIN</code> orders, and <code>FUTURES</code> / <code>SWAP</code> orders in <code>net</code> mode Only applicable to <code>Futures mode</code> and <code>Multi-currency margin</code>
tgtCcy	String	No	Whether the target currency uses the quote or base currency. <code>base_ccy</code> : Base currency, <code>quote_ccy</code> : Quote currency Only applicable to <code>SPOT</code> Market Orders Default is <code>quote_ccy</code> for buy, <code>base_ccy</code> for sell
banAmend	Boolean	No	Whether to disallow the system from amending the size of the SPOT Market Order. Valid options: <code>true</code> or <code>false</code> . The default value is <code>false</code> . If <code>true</code> , system will not amend and reject the market order if user does not have sufficient funds. Only applicable to SPOT Market Orders
pxAmendType	String	No	The price amendment type for orders <code>0</code> : Do not allow the system to amend to order price if <code>px</code> exceeds the price limit <code>1</code> : Allow the system to amend the price to the best available value within the price limit if <code>px</code> exceeds the price limit The default value is <code>0</code>
tradeQuoteCcy	String	No	The quote currency used for trading. Only applicable to <code>SPOT</code> . The default value is the quote currency of the <code>instId</code> , for example: for <code>BTC-USD</code> , the default is <code>USD</code> .
stpMode	String	No	Self trade prevention mode. <code>cancel_maker</code> , <code>cancel_taker</code> , <code>cancel_both</code> Cancel both does not support FOK
attachAlgoOrds	Array of objects	No	TP/SL information attached when placing order
> attachAlgoClOrddId	String	No	Client-supplied Algo ID when placing order attaching TP/SL A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. It will be posted to <code>algoClordId</code> when placing TP/SL order once the general order is filled completely.
> tpTriggerPx	String	Conditional	Take-profit trigger price For condition TP order, if you fill in this parameter, you should fill in the take-profit order price as well.

Parameter	Type	Required	Description
> tpTriggerRatio	String	Conditional	Take profit trigger ratio, 0.3 represents 30% Only one of <code>tpTriggerPx</code> and <code>tpTriggerPct</code> can be passed Only applicable to FUTURES and SWAP.
			Take-profit order price
> tpOrdPx	String	Conditional	For condition TP order, if you fill in this parameter, you should fill in the take-profit trigger price as well. For limit TP order, you need to fill in this parameter, but the take-profit trigger price doesn't need to be filled. If the price is -1, take-profit will be executed at the market price.
> tpOrdKind	String	No	TP order kind <code>condition</code> <code>limit</code> The default is <code>condition</code>
> slTriggerPx	String	Conditional	Stop-loss trigger price If you fill in this parameter, you should fill in the stop-loss order price.
> slTriggerRatio	String	Conditional	Stop profit trigger ratio, 0.3 represents 30% Only one of <code>slTriggerPx</code> and <code>slTriggerPct</code> can be passed Only applicable to FUTURES and SWAP.
> slOrdPx	String	Conditional	Stop-loss order price If you fill in this parameter, you should fill in the stop-loss trigger price. If the price is -1, stop-loss will be executed at the market price.
> tpTriggerPxType	String	No	Take-profit trigger price type <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price The default is <code>last</code>
> slTriggerPxType	String	No	Stop-loss trigger price type <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price The default is <code>last</code>
> sz	String	Conditional	Size. Only applicable to TP order of split TPs, and it is required for TP order of split TPs
> amendPxOnTriggerType	String	No	Whether to enable Cost-price SL. Only applicable to SL order of split TPs. Whether <code>slTriggerPx</code> will move to <code>avgPx</code> when the first TP order is triggered <code>0</code> : disable, the default value <code>1</code> : Enable

### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "c1ordId": "oktswap6",
      "ordId": "312269865356374016",
      "tag": "",
      "ts": "1695190491421",
      "sCode": "0",
      "sMsg": ""
    }
  ],
  "inTime": "1695190491421339",
  "outTime": "2023-09-04T14:21:33.999Z"
}
```

```
"outTime": "1695190491423240"  
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, 0 means success
msg	String	The error message, empty if the code is 0
data	Array of objects	Array of objects contains the response results
> ordId	String	Order ID
> clOrdId	String	Client Order ID as assigned by the client
> tag	String	Order tag
> ts	String	Timestamp when the order request processing is finished by our system, Unix timestamp format in milliseconds, e.g. 1597026383085
> sCode	String	The code of the event execution result, 0 means success.
> sMsg	String	Rejection or success message of event execution.
inTime	String	Timestamp at REST gateway when the request is received, Unix timestamp format in microseconds, e.g. 1597026383085 085123 The time is recorded after authentication.
outTime	String	Timestamp at REST gateway when the response is sent, Unix timestamp format in microseconds, e.g. 1597026383085 123

### tdMode

Trade Mode, when placing an order, you need to specify the trade mode.

#### Spot mode:

- SPOT and OPTION buyer: cash

#### Futures mode:

- Isolated MARGIN: isolated

- Cross MARGIN: cross

- SPOT: cash

- Cross FUTURES/SWAP/OPTION: cross

- Isolated FUTURES/SWAP/OPTION: isolated

#### Multi-currency margin mode:

- Cross SPOT: cross

- Cross FUTURES/SWAP/OPTION: cross

#### Portfolio margin:

- Cross SPOT: cross

- Cross FUTURES/SWAP/OPTION: cross

### clOrdId

clOrdId is a user-defined unique ID used to identify the order. It will be included in the response parameters if you have specified during order submission, and can be used as a request parameter to the endpoints to query, cancel and amend orders.

clOrdId must be unique among the clOrdIds of all pending orders.

### posSide

Position side, this parameter is not mandatory in **net** mode. If you pass it through, the only valid value is **net**.

In **long**/**short** mode, it is mandatory. Valid values are **long** or **short**.

In **long**/**short** mode, **side** and **posSide** need to be specified in the combinations below:

Open long: buy and open long (side: fill in buy; posSide: fill in long)

Open short: sell and open short (side: fill in sell; posSide: fill in short)

Close long: sell and close long (side: fill in sell; posSide: fill in long)

Close short: buy and close short (side: fill in buy; posSide: fill in short)

Portfolio margin mode: Expiry Futures and Perpetual Futures only support net mode

#### ordType

Order type. When creating a new order, you must specify the order type. The order type you specify will affect: 1) what order parameters are required, and 2) how the matching system executes your order. The following are valid order types:

limit: Limit order, which requires specified sz and px.

market: Market order. For SPOT and MARGIN, market order will be filled with market price (by swiping opposite order book). For Expiry Futures and Perpetual Futures, market order will be placed to order book with most aggressive price allowed by Price Limit Mechanism. For OPTION, market order is not supported yet. As the filled price for market orders cannot be determined in advance, OKX reserves/freezes your quote currency by an additional 5% for risk check.

post\_only: Post-only order, which the order can only provide liquidity to the market and be a maker. If the order would have executed on placement, it will be canceled instead.

fok: Fill or kill order. If the order cannot be fully filled, the order will be canceled. The order would not be partially filled.

ioc: Immediate or cancel order. Immediately execute the transaction at the order price, cancel the remaining unfilled quantity of the order, and the order quantity will not be displayed in the order book.

optimal\_limit\_ioc: Market order with ioc (immediate or cancel). Immediately execute the transaction of this market order, cancel the remaining unfilled quantity of the order, and the order quantity will not be displayed in the order book. Only applicable to Expiry Futures and Perpetual Futures.

#### sz

Quantity to buy or sell.

For SPOT/MARGIN Buy and Sell Limit Orders, it refers to the quantity in base currency.

For MARGIN Buy Market Orders, it refers to the quantity in quote currency.

For MARGIN Sell Market Orders, it refers to the quantity in base currency.

For SPOT Market Orders, it is set by tgtCcy.

For FUTURES/SWAP/OPTION orders, it refers to the number of contracts.

#### reduceOnly

When placing an order with this parameter set to true, it means that the order will reduce the size of the position only

For the same MARGIN instrument, the coin quantity of all reverse direction pending orders adds `sz` of new `reduceOnly` order cannot exceed the position assets. After the debt is paid off, if there is a remaining size of orders, the position will not be opened in reverse, but will be traded in SPOT.

For the same FUTURES/SWAP instrument, the sum of the current order size and all reverse direction reduce-only pending orders which's price-time priority is higher than the current order, cannot exceed the contract quantity of position.

Only applicable to `Futures mode` and `Multi-currency margin`

Only applicable to `MARGIN` orders, and `FUTURES`/`SWAP` orders in `net` mode

Notice: Under long/short mode of Expiry Futures and Perpetual Futures, all closing orders apply the reduce-only feature which is not affected by this parameter.

#### tgtCcy

This parameter is used to specify the order quantity in the order request is denominated in the quantity of base or quote currency. This is applicable to SPOT Market Orders only.

Base currency: base\_ccy

Quote currency: quote\_ccy

If you use the Base Currency quantity for buy market orders or the Quote Currency for sell market orders, please note:

1. If the quantity you enter is greater than what you can buy or sell, the system will execute the order according to your maximum buyable or sellable quantity. If you want to trade according to the specified quantity, you should use Limit orders.

2. When the market price is too volatile, the locked balance may not be sufficient to buy the Base Currency quantity or sell to receive the Quote Currency that you specified. We will change the quantity of the order to execute the order based on best effort principle based on your account balance. In addition, we will try to over lock a fraction of your balance to avoid changing the order quantity.

## 2.1 Example of base currency buy market order:

Taking the market order to buy 10 LTCs as an example, and the user can buy 11 LTC. At this time, if  $10 < 11$ , the order is accepted. When the LTC-USDT market price is 200, and the locked balance of the user is 3,000 USDT, as  $200*10 < 3,000$ , the market order of 10 LTC is fully executed; If the market is too volatile and the LTC-USDT market price becomes 400,  $400*10 > 3,000$ , the user's locked balance is not sufficient to buy using the specified amount of base currency, the user's maximum locked balance of 3,000 USDT will be used to settle the trade. Final transaction quantity becomes  $3,000/400 = 7.5$  LTC.

## 2.2 Example of quote currency sell market order:

Taking the market order to sell 1,000 USDT as an example, and the user can sell 1,200 USDT,  $1,000 < 1,200$ , the order is accepted. When the LTC-USDT market price is 200, and the locked balance of the user is 6 LTC, as  $1,000/200 < 6$ , the market order of 1,000 USDT is fully executed; If the market is too volatile and the LTC-USDT market price becomes 100,  $100*6 < 1,000$ , the user's locked balance is not sufficient to sell using the specified amount of quote currency, the user's maximum locked balance of 6 LTC will be used to settle the trade. Final transaction quantity becomes  $6 * 100 = 600$  USDT.

px

The value for px must be a multiple of tickSz for OPTION orders.

If not, the system will apply the rounding rules below. Using tickSz 0.0005 as an example:

The px will be rounded up to the nearest 0.0005 when the remainder of px to 0.0005 is more than 0.00025 or 'px' is less than 0.0005.

The px will be rounded down to the nearest 0.0005 when the remainder of px to 0.0005 is less than 0.00025 and 'px' is more than 0.0005.

For placing order with TP/SL:

1. TP/SL algo order will be generated only when this order is filled fully, or there is no TP/SL algo order generated.

2. Attaching TP/SL is neither supported for market buy with tgtCcy is base\_ccy or market sell with tgtCcy is quote\_ccy

3. If tpOrdKind is limit, and there is only one conditional TP order, attachAlgoClOrdId can be used as clOrdId for retrieving on "GET / Order details" endpoint.

4. For "split TPs", including condition TP order and limit TP order.

\* TP/SL orders in Split TPs only support one-way TP/SL. You can't use slTriggerPx&slOrdPx and tpTriggerPx&tpOrdPx at the same time, or error code 51076 will be thrown.

\* Take-profit trigger price types (tpTriggerPxType) must be the same in an order with Split TPs attached, or error code 51080 will be thrown.

\* Take-profit trigger prices (tpTriggerPx) cannot be the same in an order with Split TPs attached, or error code 51081 will be thrown.

\* The size of the TP order among split TPs attached cannot be empty, or error code 51089 will be thrown.

\* The total size of TP orders with Split TPs attached in a same order should equal the size of this order, or error code 51083 will be thrown.

\* The number of TP orders with Split TPs attached in a same order cannot exceed 10, or error code 51079 will be thrown.

\* Setting multiple TP and cost-price SL orders isn't supported for spot and margin trading, or error code 51077 will be thrown.

\* The number of SL orders with Split TPs attached in a same order cannot exceed 1, or error code 51084 will be thrown.

\* The number of TP orders cannot be less than 2 when cost-price SL is enabled (amendPxOnTriggerType set as 1) for Split TPs, or error code 51085 will be thrown.

\* All TP orders in one order must be of the same type, or error code 51091 will be thrown.

\* TP order prices (tpOrdPx) in one order must be different, or error code 51092 will be thrown.

\* TP limit order prices (tpOrdPx) in one order can't be -1 (market price), or error code 51093 will be thrown.

\* You can't place TP limit orders in spot, margin, or options trading. Otherwise, error code 51094 will be thrown.

## Mandatory self trade prevention (STP)

The trading platform imposes mandatory self trade prevention at master account level, which means the accounts under the same master account, including master account itself and all its affiliated sub-accounts, will be prevented from self trade. The account-level acctStpMode will be used to place orders by default. The default value of this field is 'cancel\_maker'. Users can log in to the webpage through the master account to modify this configuration. Users can also utilize the stpMode request parameter of the placing order endpoint to determine the stpMode of a certain order.

Mandatory self trade prevention will not lead to latency.

There are three STP modes. The STP mode is always taken based on the configuration in the taker order.

1. Cancel Maker: This is the default STP mode, which cancels the maker order to prevent self-trading. Then, the taker order continues to match with the next order based on the order book priority.

2. Cancel Taker: The taker order is canceled to prevent self-trading. If the user's own maker order is lower in the order book priority, the taker order is partially filled and then canceled. FOK orders are always honored and canceled if they would result in self-trading.
3. Cancel Both: Both taker and maker orders are canceled to prevent self-trading. If the user's own maker order is lower in the order book priority, the taker order is partially filled. Then, the remaining quantity of the taker order and the first maker order are canceled. FOK orders are not supported in this mode.

#### tradeQuoteCcy

For users in specific countries and regions, this parameter must be filled out for a successful order. Otherwise, the system will use the quote currency of instId as the default value, then error code 51000 will occur.

The value provided must be one of the enumerated values from tradeQuoteCcyList, which can be obtained from the endpoint Get instruments (GET /api/v5/account/instruments).

#### POST / PLACE MULTIPLE ORDERS

Place orders in batches. Maximum 20 orders can be placed per request.

Request parameters should be passed in the form of an array. Orders will be placed in turn

**RATE LIMIT: 300 ORDERS PER 2 SECONDS**

**RATE LIMIT OF LEAD TRADER LEAD INSTRUMENTS FOR COPY TRADING: 4 ORDERS PER 2 SECONDS**

**RATE LIMIT RULE (EXCEPT OPTIONS): USER ID + INSTRUMENT ID**

**RATE LIMIT RULE (OPTIONS ONLY): USER ID + INSTRUMENT FAMILY**

**PERMISSION: TRADE**

Rate limit of this endpoint will also be affected by the rules Sub-account rate limit and Fill ratio based sub-account rate limit.

Unlike other endpoints, the rate limit of this endpoint is determined by the number of orders. If there is only one order in the request, it will consume the rate limit of 'Place order'.

#### HTTP REQUEST

POST /api/v5/trade/batch-orders

##### Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Place multiple orders
place_orders_without_c10rId = [
    {"instId": "BTC-USDT", "tdMode": "cash", "c10rId": "b15", "side": "buy", "ordType": "limit", "px": "2.15", "sz": "2"}, 
    {"instId": "BTC-USDT", "tdMode": "cash", "c10rId": "b16", "side": "buy", "ordType": "limit", "px": "2.15", "sz": "2"} 
]

result = tradeAPI.place_multiple_orders(place_orders_without_c10rId)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT</code>
tdMode	String	Yes	Trade mode Margin mode <code>cross</code> <code>isolated</code>

Parameter	Type	Required	Description
			<p>Non-Margin mode <code>cash</code></p> <p><code>spot_isolated</code> (Only applicable to SPOT lead trading, <code>tdMode</code> should be <code>spot_isolated</code> for SPOT lead trading.)</p> <p>Note: <code>isolated</code> is not available in multi-currency margin mode and portfolio margin mode.</p>
ccy	String	No	<p>Margin currency</p> <p>Applicable to all <code>isolated</code> <code>MARGIN</code> orders and <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code>.</p>
clOrdId	String	No	<p>Client Order ID as assigned by the client</p> <p>A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.</p>
tag	String	No	<p>Order tag</p> <p>A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters.</p>
side	String	Yes	<p>Order side <code>buy</code> <code>sell</code></p>
posSide	String	Conditional	<p>Position side</p> <p>The default is <code>net</code> in the <code>net</code> mode</p> <p>It is required in the <code>long/short</code> mode, and can only be <code>long</code> or <code>short</code>.</p> <p>Only applicable to <code>FUTURES</code>/<code>SWAP</code>.</p>
ordType	String	Yes	<p>Order type</p> <p><code>market</code>: Market order, only applicable to <code>SPOT/MARGIN/FUTURES/SWAP</code></p> <p><code>limit</code>: Limit order</p> <p><code>post_only</code>: Post-only order</p> <p><code>fok</code>: Fill-or-kill order</p> <p><code>ioc</code>: Immediate-or-cancel order</p> <p><code>optimal_limit_ioc</code>: Market order with immediate-or-cancel order (applicable only to Expiry Futures and Perpetual Futures).</p> <p><code>mmp</code>: Market Maker Protection (only applicable to Option in Portfolio Margin mode)</p> <p><code>mmp_and_post_only</code>: Market Maker Protection and Post-only order(only applicable to Option in Portfolio Margin mode)</p>
sz	String	Yes	<p>Quantity to buy or sell</p>
px	String	Conditional	<p>Order price. Only applicable to <code>limit</code>, <code>post_only</code>, <code>fok</code>, <code>ioc</code>, <code>mmp</code>, <code>mmp_and_post_only</code> order.</p> <p>When placing an option order, one of px/pxUsd/pxVol must be filled in, and only one can be filled in</p>
pxUsd	String	Conditional	<p>Place options orders in <code>USD</code></p> <p>Only applicable to options</p> <p>When placing an option order, one of px/pxUsd/pxVol must be filled in, and only one can be filled in</p>
pxVol	String	Conditional	<p>Place options orders based on implied volatility, where 1 represents 100%</p> <p>Only applicable to options</p> <p>When placing an option order, one of px/pxUsd/pxVol must be filled in, and only one can be filled in</p>
reduceOnly	Boolean	No	<p>Whether the order can only reduce position size.</p> <p>Valid options: <code>true</code> or <code>false</code>. The default value is <code>false</code>.</p> <p>Only applicable to <code>MARGIN</code> orders, and <code>FUTURES</code>/<code>SWAP</code> orders in <code>net</code> mode</p> <p>Only applicable to <code>Futures mode</code> and <code>Multi-currency margin</code></p>
tgtCcy	String	No	<p>Order quantity unit setting for <code>sz</code></p> <p><code>base_ccy</code>: Base currency, <code>quote_ccy</code>: Quote currency</p> <p>Only applicable to <code>SPOT</code> Market Orders</p> <p>Default is <code>quote_ccy</code> for buy, <code>base_ccy</code> for sell</p>
banAmend	Boolean	No	<p>Whether to disallow the system from amending the size of the SPOT Market Order.</p> <p>Valid options: <code>true</code> or <code>false</code>. The default value is <code>false</code>.</p> <p>If <code>true</code>, system will not amend and reject the market order if user does not have sufficient</p>

Parameter	Type	Required	Description
			funds. Only applicable to SPOT Market Orders
pxAmendType	String	No	The price amendment type for orders <input type="radio"/> 0: Do not allow the system to amend to order price if <code>px</code> exceeds the price limit <input checked="" type="radio"/> 1: Allow the system to amend the price to the best available value within the price limit if <code>px</code> exceeds the price limit The default value is <code>0</code>
tradeQuoteCcy	String	No	The quote currency used for trading. Only applicable to <code>SPOT</code> . The default value is the quote currency of the <code>instId</code> , for example: for <code>BTC-USD</code> , the default is <code>USD</code> .
stpMode	String	No	Self trade prevention mode. <input type="radio"/> <code>cancel_maker</code> <input type="radio"/> <code>cancel_taker</code> <input type="radio"/> <code>cancel_both</code> Cancel both does not support FOK.
attachAlgoOrds	Array of objects	No	TP/SL information attached when placing order
> attachAlgoClOrdId	String	No	Client-supplied Algo ID when placing order attaching TP/SL A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. It will be posted to <code>algoClOrdId</code> when placing TP/SL order once the general order is filled completely.
> tpTriggerPx	String	Conditional	Take-profit trigger price For condition TP order, if you fill in this parameter, you should fill in the take-profit order price as well.
> tpTriggerRatio	String	Conditional	Take profit trigger ratio, 0.3 represents 30% Only one of <code>tpTriggerPx</code> and <code>tpTriggerPct</code> can be passed Only applicable to FUTURES and SWAP.
> tpOrdPx	String	Conditional	Take-profit order price For condition TP order, if you fill in this parameter, you should fill in the take-profit trigger price as well. For limit TP order, you need to fill in this parameter, take-profit trigger needn't to be filled. If the price is -1, take-profit will be executed at the market price.
> tpOrdKind	String	No	TP order kind <input type="radio"/> <code>condition</code> <input type="radio"/> <code>limit</code> The default is <code>condition</code>
> slTriggerPx	String	Conditional	Stop-loss trigger price If you fill in this parameter, you should fill in the stop-loss order price.
> slTriggerRatio	String	Conditional	Stop profit trigger ratio, 0.3 represents 30% Only one of <code>slTriggerPx</code> and <code>slTriggerPct</code> can be passed Only applicable to FUTURES and SWAP.
> slOrdPx	String	Conditional	Stop-loss order price If you fill in this parameter, you should fill in the stop-loss trigger price. If the price is -1, stop-loss will be executed at the market price.
> tpTriggerPxType	String	No	Take-profit trigger price type <input type="radio"/> <code>last</code> : last price <input type="radio"/> <code>index</code> : index price

Parameter	Type	Required	Description
			<p><code>mark</code>: mark price The default is last</p>
> slTriggerPxType	String	No	<p>Stop-loss trigger price type  <code>last</code>: last price  <code>index</code>: index price  <code>mark</code>: mark price The default is last</p>
> sz	String	Conditional	Size. Only applicable to TP order of split TPs, and it is required for TP order of split TPs
> amendPxOnTriggerType	String	No	<p>Whether to enable Cost-price SL. Only applicable to SL order of split TPs. Whether <code>slTriggerPxType</code> will move to <code>avgPx</code> when the first TP order is triggered  <code>0</code>: disable, the default value  <code>1</code>: Enable</p>

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "cl0rId": "oktswap6",
      "ordId": "12345689",
      "tag": "",
      "ts": "1695190491421",
      "sCode": "0",
      "sMsg": ""
    },
    {
      "cl0rId": "oktswap7",
      "ordId": "12344",
      "tag": "",
      "ts": "1695190491421",
      "sCode": "0",
      "sMsg": ""
    }
  ],
  "inTime": "1695190491421339",
  "outTime": "1695190491423240"
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, <code>0</code> means success
msg	String	The error message, empty if the code is 0
data	Array of objects	Array of objects contains the response results
> ordId	String	Order ID
> clOrdId	String	Client Order ID as assigned by the client
> tag	String	Order tag
> ts	String	Timestamp when the order request processing is finished by our system, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> sCode	String	The code of the event execution result, <code>0</code> means success.

Parameter	Type	Description
> sMsg	String	Rejection or success message of event execution.
inTime	String	<p>Timestamp at REST gateway when the request is received, Unix timestamp format in microseconds, e.g. <code>1597026383</code>  <code>085123</code></p> <p>The time is recorded after authentication.</p>
outTime	String	<p>Timestamp at REST gateway when the response is sent, Unix timestamp format in microseconds, e.g. <code>1597026383085</code>  <code>123</code></p>

In the `Portfolio Margin` account mode, either all orders are accepted by the system successfully, or all orders are rejected by the system.

### clOrdId

clOrdId is a user-defined unique ID used to identify the order. It will be included in the response parameters if you have specified during order submission, and can be used as a request parameter to the endpoints to query, cancel and amend orders.  
 clOrdId must be unique among all pending orders and the current request.

## POST / CANCEL ORDER

Cancel an incomplete order.

**RATE LIMIT: 60 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE (EXCEPT OPTIONS): USER ID + INSTRUMENT ID**

**RATE LIMIT RULE (OPTIONS ONLY): USER ID + INSTRUMENT FAMILY**

**PERMISSION: TRADE**

**HTTP REQUEST**

`POST /api/v5/trade/cancel-order`

### Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Cancel order
result = tradeAPI.cancel_order(instId="BTC-USDT", ordId="590908157585625111")
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT</code>
ordId	String	Conditional	Order ID Either <code>ordId</code> or <code>clOrdId</code> is required. If both are passed, <code>ordId</code> will be used.
clOrdId	String	Conditional	Client Order ID as assigned by the client

### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "cl0rId": "oktswap6",
      "ordId": "12345689",
      "ts": "1695190491421",
      "sCode": "0",
      "sMsg": ""
    }
  ],
  "inTime": "1695190491421339",
  "outTime": "1695190491423240"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, 0 means success
msg	String	The error message, empty if the code is 0
data	Array of objects	Array of objects contains the response results
> ordId	String	Order ID
> clOrdId	String	Client Order ID as assigned by the client
> ts	String	Timestamp when the order request processing is finished by our system, Unix timestamp format in milliseconds, e.g. 1597026383085
> sCode	String	The code of the event execution result, 0 means success.
> sMsg	String	Rejection message if the request is unsuccessful.
inTime	String	Timestamp at REST gateway when the request is received, Unix timestamp format in microseconds, e.g. 085123 The time is recorded after authentication.
outTime	String	Timestamp at REST gateway when the response is sent, Unix timestamp format in microseconds, e.g. 1597026383085 123

Cancel order returns with sCode equal to 0. It is not strictly considered that the order has been canceled. It only means that your cancellation request has been accepted by the system server. The result of the cancellation is subject to the state pushed by the order channel or the get order state.

## POST / CANCEL MULTIPLE ORDERS

Cancel incomplete orders in batches. Maximum 20 orders can be canceled per request. Request parameters should be passed in the form of an array.

**RATE LIMIT: 300 ORDERS PER 2 SECONDS**

**RATE LIMIT RULE (EXCEPT OPTIONS): USER ID + INSTRUMENT ID**

**RATE LIMIT RULE (OPTIONS ONLY): USER ID + INSTRUMENT FAMILY**

**PERMISSION: TRADE**

Unlike other endpoints, the rate limit of this endpoint is determined by the number of orders. If there is only one order in the request, it will consume the rate limit of 'Cancel order'.

## Request Example

```

import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Cancel multiple orders by ordId
cancel_orders_with_orderId = [
    {"instId": "BTC-USDT", "ordId": "590908157585625111"}, 
    {"instId": "BTC-USDT", "ordId": "590908544950571222"} 
]

result = tradeAPI.cancel_multiple_orders(cancel_orders_with_orderId)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT</code>
ordId	String	Conditional	Order ID Either <code>ordId</code> or <code>clOrdId</code> is required. If both are passed, <code>ordId</code> will be used.
clOrdId	String	Conditional	Client Order ID as assigned by the client

## Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "clOrdId": "oktswap6",
      "ordId": "12345689",
      "ts": "1695190491421",
      "sCode": "0",
      "sMsg": ""
    },
    {
      "clOrdId": "oktswap7",
      "ordId": "12344",
      "ts": "1695190491421",
      "sCode": "0",
      "sMsg": ""
    }
  ],
  "inTime": "1695190491421339",
  "outTime": "1695190491423240"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, <code>0</code> means success
msg	String	The error message, empty if the code is 0

Parameter	Type	Description
data	Array of objects	Array of objects contains the response results
> ordId	String	Order ID
> clOrdId	String	Client Order ID as assigned by the client
> ts	String	Timestamp when the order request processing is finished by our system, Unix timestamp format in milliseconds, e.g. 1597026383085
> sCode	String	The code of the event execution result, 0 means success.
> sMsg	String	Rejection message if the request is unsuccessful.
inTime	String	Timestamp at REST gateway when the request is received, Unix timestamp format in microseconds, e.g. 1597026383085 085123 The time is recorded after authentication.
outTime	String	Timestamp at REST gateway when the response is sent, Unix timestamp format in microseconds, e.g. 1597026383085 123

## POST / AMEND ORDER

Amend an incomplete order.

**RATE LIMIT: 60 REQUESTS PER 2 SECONDS**

**RATE LIMIT OF LEAD TRADER LEAD INSTRUMENTS FOR COPY TRADING: 4 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE (EXCEPT OPTIONS): USER ID + INSTRUMENT ID**

**RATE LIMIT RULE (OPTIONS ONLY): USER ID + INSTRUMENT FAMILY**

## PERMISSION: TRADE

Rate limit of this endpoint will also be affected by the rules Sub-account rate limit and Fill ratio based sub-account rate limit.

## HTTP REQUEST

POST /api/v5/trade/amend-order

Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Amend order
result = tradeAPI.amend_order(
    instId="BTC-USDT",
    ordId="590909145319051111",
    newSz="2"
)
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID

Parameter	Type	Required	Description
cxlOnFail	Boolean	No	Whether the order needs to be automatically canceled when the order amendment fails Valid options: <code>false</code> or <code>true</code> , the default is <code>false</code> .
ordId	String	Conditional	Order ID Either <code>ordId</code> or <code>clOrdId</code> is required. If both are passed, <code>ordId</code> will be used.
clOrdId	String	Conditional	Client Order ID as assigned by the client
reqId	String	No	Client Request ID as assigned by the client for order amendment A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. The response will include the corresponding <code>reqId</code> to help you identify the request if you provide it in the request.
newSz	String	Conditional	New quantity after amendment and it has to be larger than 0. When amending a partially-filled order, the <code>newSz</code> should include the amount that has been filled.
newPx	String	Conditional	New price after amendment. When modifying options orders, users can only fill in one of the following: <code>newPx</code> , <code>newPxUsd</code> , or <code>newPxVol</code> . It must be consistent with parameters when placing orders. For example, if users placed the order using <code>px</code> , they should use <code>newPx</code> when modifying the order.
newPxUsd	String	Conditional	Modify options orders using USD prices Only applicable to options. When modifying options orders, users can only fill in one of the following: <code>newPx</code> , <code>newPxUsd</code> , or <code>newPxVol</code> .
newPxVol	String	Conditional	Modify options orders based on implied volatility, where 1 represents 100% Only applicable to options. When modifying options orders, users can only fill in one of the following: <code>newPx</code> , <code>newPxUsd</code> , or <code>newPxVol</code> .
pxAmendType	String	No	The price amendment type for orders <code>0</code> : Do not allow the system to amend to order price if <code>newPx</code> exceeds the price limit <code>1</code> : Allow the system to amend the price to the best available value within the price limit if <code>newPx</code> exceeds the price limit The default value is <code>0</code>
attachAlgoOrds	Array of objects	No	TP/SL information attached when placing order
> attachAlgoid	String	Conditional	The order ID of attached TP/SL order. It is required to identify the TP/SL order when amending. It will not be posted to algoid when placing TP/SL order after the general order is filled completely.
> attachAlgoClOrdId	String	Conditional	Client-supplied Algo ID when placing order attaching TP/SL A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. It will be posted to <code>algoClOrdId</code> when placing TP/SL order once the general order is filled completely.
> newTpTriggerPx	String	Conditional	Take-profit trigger price. Either the take profit trigger price or order price is 0, it means that the take profit is deleted.
> newTpTriggerRatio	String	Conditional	Take profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.
> newTpOrdPx	String	Conditional	Take-profit order price If the price is -1, take-profit will be executed at the market price.
> newTpOrdKind	String	No	TP order kind <code>condition</code>

Parameter	Type	Required	Description
			<code>limit</code>
> newSITriggerPx	String	Conditional	Stop-loss trigger price Either the stop loss trigger price or order price is 0, it means that the stop loss is deleted.
> newSITriggerRatio	String	Conditional	Stop profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.
> newSIOrdPx	String	Conditional	Stop-loss order price If the price is -1, stop-loss will be executed at the market price.
> newTpTriggerPxType	String	Conditional	Take-profit trigger price type <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price Only applicable to <code>FUTURES</code> / <code>SWAP</code> If you want to add the take-profit, this parameter is required
> newSITriggerPxType	String	Conditional	Stop-loss trigger price type <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price Only applicable to <code>FUTURES</code> / <code>SWAP</code> If you want to add the stop-loss, this parameter is required
> sz	String	Conditional	New size. Only applicable to TP order of split TPs, and it is required for TP order of split TPs
> amendPxOnTriggerType	String	No	Whether to enable Cost-price SL. Only applicable to SL order of split TPs. <code>0</code> : disable, the default value <code>1</code> : Enable

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "c10rId": "",
      "ordId": "12344",
      "ts": "1695190491421",
      "reqId": "b12344",
      "sCode": "0",
      "sMsg": ""
    }
  ],
  "inTime": "1695190491421339",
  "outTime": "1695190491423240"
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, <code>0</code> means success
msg	String	The error message, empty if the code is 0
data	Array of objects	Array of objects contains the response results
> ordId	String	Order ID
> c10rId	String	Client Order ID as assigned by the client

Parameter	Type	Description
> ts	String	Timestamp when the order request processing is finished by our system, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> reqId	String	Client Request ID as assigned by the client for order amendment.
> sCode	String	The code of the event execution result, <code>0</code> means success.
> sMsg	String	Rejection message if the request is unsuccessful.
inTime	String	Timestamp at REST gateway when the request is received, Unix timestamp format in microseconds, e.g. <code>1597026383085123</code> The time is recorded after authentication.
outTime	String	Timestamp at REST gateway when the response is sent, Unix timestamp format in microseconds, e.g. <code>1597026383085123</code>

#### newSz

If the new quantity of the order is less than or equal to the filled quantity when you are amending a partially-filled order, the order status will be changed to filled.

The amend order returns sCode equal to 0. It is not strictly considered that the order has been amended. It only means that your amend order request has been accepted by the system server. The result of the amend is subject to the status pushed by the order channel or the order status query

#### POST / AMEND MULTIPLE ORDERS

Amend incomplete orders in batches. Maximum 20 orders can be amended per request. Request parameters should be passed in the form of an array.

#### RATE LIMIT: 300 ORDERS PER 2 SECONDS

#### RATE LIMIT OF LEAD TRADER LEAD INSTRUMENTS FOR COPY TRADING: 4 ORDERS PER 2 SECONDS

#### RATE LIMIT RULE (EXCEPT OPTIONS): USER ID + INSTRUMENT ID

#### RATE LIMIT RULE (OPTIONS ONLY): USER ID + INSTRUMENT FAMILY

#### PERMISSION: TRADE

Rate limit of this endpoint will also be affected by the rules Sub-account rate limit and Fill ratio based sub-account rate limit.

Unlike other endpoints, the rate limit of this endpoint is determined by the number of orders. If there is only one order in the request, it will consume the rate limit of 'Amend order'.

#### HTTP REQUEST

`POST /api/v5/trade/amend-batch-orders`

#### Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Amend incomplete orders in batches by orderId
amend_orders_with_orderId = [
```

```

    {"instId": "BTC-USDT", "ordId": "590909308792049444", "newSz": "2"},  

    {"instId": "BTC-USDT", "ordId": "590909308792049555", "newSz": "2"}  

]

```

```

result = tradeAPI.amend_multiple_orders(amend_orders_with_orderId)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID
cxlOnFail	Boolean	No	Whether the order needs to be automatically canceled when the order amendment fails false true, the default is false.
ordId	String	Conditional	Order ID Either ordId or clOrdId is required, if both are passed, ordId will be used.
clOrdId	String	Conditional	Client Order ID as assigned by the client
reqId	String	No	Client Request ID as assigned by the client for order amendment A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. The response will include the corresponding reqId to help you identify the request if you provide it in the request.
newSz	String	Conditional	New quantity after amendment and it has to be larger than 0. When amending a partially-filled order, the newSz should include the amount that has been filled.
newPx	String	Conditional	New price after amendment. When modifying options orders, users can only fill in one of the following: newPx, newPxUsd, or newPxVol. It must be consistent with parameters when placing orders. For example, if users placed the order using px, they should use newPx when modifying the order.
newPxUsd	String	Conditional	Modify options orders using USD prices Only applicable to options. When modifying options orders, users can only fill in one of the following: newPx, newPxUsd, or newPxVol.
newPxVol	String	Conditional	Modify options orders based on implied volatility, where 1 represents 100% Only applicable to options. When modifying options orders, users can only fill in one of the following: newPx, newPxUsd, or newPxVol.
pxAmendType	String	No	The price amendment type for orders 0: Do not allow the system to amend to order price if newPx exceeds the price limit 1: Allow the system to amend the price to the best available value within the price limit if newPx exceeds the price limit The default value is 0
attachAlgoOrds	Array of objects	No	TP/SL information attached when placing order
> attachAlgoid	String	Conditional	The order ID of attached TP/SL order. It is required to identify the TP/SL order when amending. It will not be posted to algoid when placing TP/SL order after the general order is filled completely.
> attachAlgoClOrdId	String	Conditional	Client-supplied Algo ID when placing order attaching TP/SL A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. It will be posted to algoclOrdId when placing TP/SL order once the general order is filled completely.

Parameter	Type	Required	Description
> newTpTriggerPx	String	Conditional	Take-profit trigger price. Either the take profit trigger price or order price is 0, it means that the take profit is deleted.
> newTpTriggerRatio	String	Conditional	Take profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.
> newTpOrdPx	String	Conditional	Take-profit order price If the price is -1, take-profit will be executed at the market price.
> newTpOrdKind	String	No	TP order kind condition limit
> newSITriggerPx	String	Conditional	Stop-loss trigger price Either the stop loss trigger price or order price is 0, it means that the stop loss is deleted.
> newSITriggerRatio	String	Conditional	Stop profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.
> newSIOrdPx	String	Conditional	Stop-loss order price If the price is -1, stop-loss will be executed at the market price.
> newTpTriggerPxType	String	Conditional	Take-profit trigger price type last: last price index: index price mark: mark price Only applicable to FUTURES/SWAP If you want to add the take-profit, this parameter is required
> newSITriggerPxType	String	Conditional	Stop-loss trigger price type last: last price index: index price mark: mark price Only applicable to FUTURES/SWAP If you want to add the stop-loss, this parameter is required
> sz	String	Conditional	New size. Only applicable to TP order of split TPs, and it is required for TP order of split TPs
> amendPxOnTriggerType	String	No	Whether to enable Cost-price SL. Only applicable to SL order of split TPs. 0: disable, the default value 1: Enable

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "cl0rdId": "oktswap6",
      "ordId": "12345689",
      "ts": "1695190491421",
      "reqId": "b12344",
      "sCode": "0",
      "sMsg": ""
    },
    {
      "cl0rdId": "oktswap7",
      "ordId": "12344",
      "ts": "1695190491421",
      "reqId": "b12344",
      "sCode": "0",
      "sMsg": ""
    }
  ]
}
```

```
],
  "inTime": "1695190491421339",
  "outTime": "1695190491423240"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, 0 means success
msg	String	The error message, empty if the code is 0
data	Array of objects	Array of objects contains the response results
> ordId	String	Order ID
> clOrdId	String	Client Order ID as assigned by the client
> ts	String	Timestamp when the order request processing is finished by our system, Unix timestamp format in milliseconds, e.g. 1597026383085
> reqId	String	Client Request ID as assigned by the client for order amendment.
> sCode	String	The code of the event execution result, 0 means success.
> sMsg	String	Rejection message if the request is unsuccessful.
inTime	String	Timestamp at REST gateway when the request is received, Unix timestamp format in microseconds, e.g. 1597026383085 085123 The time is recorded after authentication.
outTime	String	Timestamp at REST gateway when the response is sent, Unix timestamp format in microseconds, e.g. 1597026383085 123

### newSz

If the new quantity of the order is less than or equal to the filled quantity when you are amending a partially-filled order, the order status will be changed to filled.

## POST / CLOSE POSITIONS

Close the position of an instrument via a market order.

### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE (EXCEPT OPTIONS): USER ID + INSTRUMENT ID

### RATE LIMIT RULE (OPTIONS ONLY): USER ID + INSTRUMENT FAMILY

### PERMISSION: TRADE

### HTTP REQUEST

```
POST /api/v5/trade/close-position
```

### Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1
```

```

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Close the position of an instrument via a market order
result = tradeAPI.close_positions(
    instId="BTC-USDT-SWAP",
    mgnMode="cross"
)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID
posSide	String	Conditional	Position side This parameter can be omitted in <code>net</code> mode, and the default value is <code>net</code> . You can only fill with <code>net</code> . This parameter must be filled in under the <code>long/short</code> mode. Fill in <code>long</code> for close-long and <code>short</code> for close-short.
mgnMode	String	Yes	Margin mode <code>cross</code> <code>isolated</code>
ccy	String	Conditional	Margin currency, required in the case of closing <code>cross</code> <code>MARGIN</code> position for <code>Futures mode</code> .
autoCxl	Boolean	No	Whether any pending orders for closing out needs to be automatically canceled when close position via a market order. <code>false</code> or <code>true</code> , the default is <code>false</code> .
clOrdId	String	No	Client-supplied ID A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
tag	String	No	Order tag A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters.

## Response Example

```

{
  "code": "0",
  "data": [
    {
      "clOrdId": "",
      "instId": "BTC-USDT-SWAP",
      "posSide": "long",
      "tag": ""
    }
  ],
  "msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID
posSide	String	Position side
clOrdId	String	Client-supplied ID A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
tag	String	Order tag A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters.

if there are any pending orders for closing out and the orders do not need to be automatically canceled, it will return an error code and message to prompt users to cancel pending orders before closing the positions.

## GET / ORDER DETAILS

Retrieve order details.

**RATE LIMIT: 60 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE (EXCEPT OPTIONS): USER ID + INSTRUMENT ID**

**RATE LIMIT RULE (OPTIONS ONLY): USER ID + INSTRUMENT FAMILY**

**PERMISSION: READ**

### HTTP REQUEST

```
GET /api/v5/trade/order
```

#### Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Retrieve order details by ordId
result = tradeAPI.get_order(
    instId="BTC-USDT",
    ordId="680800019749904384"
)
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT</code> Only applicable to live instruments
ordId	String	Conditional	Order ID Either <code>ordId</code> or <code>c10rdId</code> is required, if both are passed, <code>ordId</code> will be used
c10rdId	String	Conditional	Client Order ID as assigned by the client If the <code>c10rdId</code> is associated with multiple orders, only the latest one will be returned.

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "accFillSz": "0.00192834",
      "algoC10rdId": "",
      "algoId": "",
      "attachAlgoC10rdId": "",
      "attachAlgoOrds": [],
      "avgPx": "51858",
      "cTime": "1708587373361",
      "cancelSource": "",
      "cancelSourceReason": "",
      "category": "normal",
      "ccy": "",
      "c10rdId": ""
```

```

"fee": "-0.0000192834",
"feeCcy": "BTC",
"fillPx": "51858",
"fillSz": "0.00192834",
"fillTime": "1708587373361",
"instId": "BTC-USDT",
"instType": "SPOT",
"isTpLimit": "false",
"lever": "",
"linkedAlgoOrd": {
  "algoid": ""
},
"ordId": "680800019749904384",
"ordType": "market",
"pnl": "0",
"posSide": "net",
"px": "",
"pxType": "",
"pxUsd": "",
"pxVol": "",
"quickMgnType": "",
"rebate": "0",
"rebateCcy": "USDT",
"reduceOnly": "false",
"side": "buy",
"slOrdPx": "",
"slTriggerPx": "",
"slTriggerPxType": "",
"source": "",
"state": "filled",
"stpId": "",
"stpMode": "",
"sz": "100",
"tag": "",
"tdMode": "cash",
"tgtCcy": "quote_ccy",
"tpOrdPx": "",
"tpTriggerPx": "",
"tpTriggerPxType": "",
"tradeId": "744876980",
"tradeQuoteCcy": "USDT",
"uTime": "1708587373362"
}
],
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type <div style="display: flex; justify-content: space-around; align-items: center;"> <span>SPOT</span> <span>MARGIN</span> <span>SWAP</span> <span>FUTURES</span> <span>OPTION</span> </div>
instId	String	Instrument ID
tgtCcy	String	Order quantity unit setting for <code>sz</code> <code>base_ccy</code> : Base currency, <code>quote_ccy</code> : Quote currency Only applicable to <code>SPOT</code> Market Orders Default is <code>quote_ccy</code> for buy, <code>base_ccy</code> for sell
ccy	String	Margin currency Applicable to all <code>isolated</code> <code>MARGIN</code> orders and <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code> , <code>FUTURES</code> and <code>SWAP</code> contracts.
ordId	String	Order ID
clOrdId	String	Client Order ID as assigned by the client

Parameter	Type	Description
tag	String	Order tag
px	String	Price For options, use coin as unit (e.g. BTC, ETH)
pxUsd	String	Options price in USDOnly applicable to options; return "" for other instrument types
pxVol	String	Implied volatility of the options orderOnly applicable to options; return "" for other instrument types
pxType	String	Price type of options <code>px</code> : Place an order based on price, in the unit of coin (the unit for the request parameter px is BTC or ETH) <code>pxVol</code> : Place an order based on pxVol <code>pxUsd</code> : Place an order based on pxUsd, in the unit of USD (the unit for the request parameter px is USD)
sz	String	Quantity to buy or sell
pnl	String	Profit and loss (excluding the fee). Applicable to orders which have a trade and aim to close position. It always is 0 in other conditions
ordType	String	Order type <code>market</code> : Market order <code>limit</code> : Limit order <code>post_only</code> : Post-only order <code>fok</code> : Fill-or-kill order <code>ioc</code> : Immediate-or-cancel order <code>optimal_limit_ioc</code> : Market order with immediate-or-cancel order <code>mmp</code> : Market Maker Protection (only applicable to Option in Portfolio Margin mode) <code>mmp_and_post_only</code> : Market Maker Protection and Post-only order(only applicable to Option in Portfolio Margin mode) <code>op_fok</code> : Simple options (fok)
side	String	Order side
posSide	String	Position side
tdMode	String	Trade mode
accFillSz	String	Accumulated fill quantity The unit is <code>base_ccy</code> for SPOT and MARGIN, e.g. BTC-USDT, the unit is BTC; For market orders, the unit both is <code>base_ccy</code> when the tgtCcy is <code>base_ccy</code> or <code>quote_ccy</code> ; The unit is contract for <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>
fillPx	String	Last filled price. If none is filled, it will return "".
tradeld	String	Last traded ID
fillSz	String	Last filled quantity The unit is <code>base_ccy</code> for SPOT and MARGIN, e.g. BTC-USDT, the unit is BTC; For market orders, the unit both is <code>base_ccy</code> when the tgtCcy is <code>base_ccy</code> or <code>quote_ccy</code> ; The unit is contract for <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>
fillTime	String	Last filled time
avgPx	String	Average filled price. If none is filled, it will return "".
state	String	State <code>canceled</code> <code>live</code> <code>partially_filled</code> <code>filled</code> <code>mmp_canceled</code>

Parameter	Type	Description
stpId	String	Self trade prevention ID Return "" if self trade prevention is not applicable (deprecated)
stpMode	String	Self trade prevention mode
lever	String	Leverage, from <code>0.01</code> to <code>125</code> . Only applicable to <code>MARGIN/FUTURES/SWAP</code>
attachAlgoClOrdId	String	Client-supplied Algo ID when placing order attaching TP/SL.
tpTriggerPx	String	Take-profit trigger price.
tpTriggerPxType	String	Take-profit trigger price type. <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price
tpOrdPx	String	Take-profit order price.
slTriggerPx	String	Stop-loss trigger price.
slTriggerPxType	String	Stop-loss trigger price type. <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price
slOrdPx	String	Stop-loss order price.
attachAlgoOrds	Array of objects	TP/SL information attached when placing order
> attachAlgId	String	The order ID of attached TP/SL order. It can be used to identify the TP/SL order when amending. It will not be posted to algId when placing TP/SL order after the general order is filled completely.
> attachAlgoClOrdId	String	Client-supplied Algo ID when placing order attaching TP/SL A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. It will be posted to <code>algoClOrdId</code> when placing TP/SL order once the general order is filled completely.
> tpOrdKind	String	TP order kind <code>condition</code> <code>limit</code>
> tpTriggerPx	String	Take-profit trigger price.
> tpTriggerRatio	String	Take profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.
> tpTriggerPxType	String	Take-profit trigger price type. <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price
> tpOrdPx	String	Take-profit order price.
> slTriggerPx	String	Stop-loss trigger price.
> slTriggerRatio	String	Stop profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.
> slTriggerPxType	String	Stop-loss trigger price type. <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price

Parameter	Type	Description
> slOrdPx	String	Stop-loss order price.
> sz	String	Size. Only applicable to TP order of split TPs
> amendPxOnTriggerType	String	Whether to enable Cost-price SL. Only applicable to SL order of split TPs. 0: disable, the default value 1: Enable
> amendPxOnTriggerType	String	Whether to enable Cost-price SL. Only applicable to SL order of split TPs. 0: disable, the default value 1: Enable
> failCode	String	The error code when failing to place TP/SL order, e.g. 51020 The default is ""
> failReason	String	The error reason when failing to place TP/SL order. The default is ""
linkedAlgoOrd	Object	Linked SL order detail, only applicable to the order that is placed by one-cancels-the-other (OCO) order that contains the TP limit order.
> algold	String	Algo ID
feeCcy	String	Fee currency For maker sell orders of Spot and Margin, this represents the quote currency. For all other cases, it represents the currency in which fees are charged.
fee	String	Fee amount For Spot and Margin (excluding maker sell orders): accumulated fee charged by the platform, always negative For maker sell orders in Spot and Margin, Expiry Futures, Perpetual Futures and Options: accumulated fee and rebate (always in quote currency for maker sell orders in Spot and Margin)
rebateCcy	String	Rebate currency For maker sell orders of Spot and Margin, this represents the base currency. For all other cases, it represents the currency in which rebates are paid.
rebate	String	Rebate amount, only applicable to Spot and Margin For maker sell orders: <del>Accumulated fee and</del> rebate amount in the unit of base currency. For all other cases, it represents the maker rebate amount, always positive, return "" if no rebate.
source	String	Order source 6: The normal order triggered by the <code>trigger order</code> 7: The normal order triggered by the <code>TP/SL order</code> 13: The normal order triggered by the algo order 25: The normal order triggered by the <code>trailing stop order</code> 34: The normal order triggered by the chase order
category	String	Category <code>normal</code> <code>twap</code> <code>ad1</code> <code>full_liquidation</code> <code>partial_liquidation</code> <code>delivery</code> ddh: Delta dynamic hedge <code>auto_conversion</code>
reduceOnly	String	Whether the order can only reduce the position size. Valid options: true or false.
isTpLimit	String	Whether it is TP limit order. true or false
cancelSource	String	Code of the cancellation source.

Parameter	Type	Description
cancelSourceReason	String	Reason for the cancellation.
quickMgnType	String	Quick Margin type, Only applicable to Quick Margin Mode of isolated margin <code>manual</code> , <code>auto_borrow</code> , <code>auto_repay</code>
algoClOrdId	String	Client-supplied Algo ID. There will be a value when algo order attaching <code>algoClOrdId</code> is triggered, or it will be "".
algold	String	Algo ID. There will be a value when algo order is triggered, or it will be "".
uTime	String	Update time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
cTime	String	Creation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
tradeQuoteCcy	String	The quote currency used for trading.

#### GET / ORDER LIST

Retrieve all incomplete orders under the current account.

**RATE LIMIT: 60 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

#### HTTP REQUEST

`GET /api/v5/trade/orders-pending`

Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Retrieve all incomplete orders
result = tradeAPI.get_order_list(
    instType="SPOT",
    ordType="post_only,fok,ioc"
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type <code>SPOT</code> <code>MARGIN</code> <code>SWAP</code> <code>FUTURES</code> <code>OPTION</code>
instFamily	String	No	Instrument family Applicable to <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>
instId	String	No	Instrument ID, e.g. <code>BTC-USD-200927</code>

Parameter	Type	Required	Description
ordType	String	No	Order type <input type="button" value="market"/> : Market order <input type="button" value="limit"/> : Limit order <input type="button" value="post_only"/> : Post-only order <input type="button" value="fok"/> : Fill-or-kill order <input type="button" value="ioc"/> : Immediate-or-cancel order <input type="button" value="optimal_limit_ioc"/> : Market order with immediate-or-cancel order <input type="button" value="mmp"/> : Market Maker Protection (only applicable to Option in Portfolio Margin mode) <input type="button" value="mmp_and_post_only"/> : Market Maker Protection and Post-only order(only applicable to Option in Portfolio Margin mode) <input type="button" value="op_fok"/> : Simple options (fok)
state	String	No	State <input type="button" value="live"/> <input type="button" value="partially_filled"/>
after	String	No	Pagination of data to return records earlier than the requested <input type="button" value="ordId"/>
before	String	No	Pagination of data to return records newer than the requested <input type="button" value="ordId"/>
limit	String	No	Number of results per request. The maximum is <input type="button" value="100"/> ; The default is <input type="button" value="100"/>

### Response Example

```
{
  "code": "0",
  "data": [
    {
      "accFillSz": "0",
      "algoClOrdId": "",
      "algoId": "",
      "attachAlgoClOrdId": "",
      "attachAlgoOrds": [],
      "avgPx": "",
      "cTime": "1724733617998",
      "cancelSource": "",
      "cancelSourceReason": "",
      "category": "normal",
      "ccy": "",
      "clOrdId": "",
      "fee": "0",
      "feeCcy": "BTC",
      "fillPx": "",
      "fillSz": "0",
      "fillTime": "",
      "instId": "BTC-USDT",
      "instType": "SPOT",
      "isTpLimit": "false",
      "lever": "",
      "linkedAlgoOrd": {
        "algoId": ""
      },
      "ordId": "1752588852617379840",
      "ordType": "post_only",
      "pnl": "0",
      "posSide": "net",
      "px": "13013.5",
      "pxType": "",
      "pxUsd": "",
      "pxVol": "",
      "quickMgnType": "",
      "rebate": "0",
      "rebateCcy": "USDT",
      "reduceOnly": "false",
      "side": "buy",
      "slOrdPx": "",
      "slTriggerPx": "",
      "slTriggerPxType": "",
      "source": ""
    }
  ]
}
```

```

        "state": "live",
        "stpId": "",
        "stpMode": "cancel_maker",
        "sz": "0.001",
        "tag": "",
        "tdMode": "cash",
        "tgtCcy": "",
        "tpOrdPx": "",
        "tpTriggerPx": "",
        "tpTriggerPxType": "",
        "tradeId": "",
        "tradeQuoteCcy": "USDT",
        "uTime": "1724733617998"
    }
],
"msg": ""
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type
instId	String	Instrument ID
tgtCcy	String	Order quantity unit setting for <code>sz</code> <code>base_ccy</code> : Base currency, <code>quote_ccy</code> : Quote currency Only applicable to <code>SPOT</code> Market Orders Default is <code>quote_ccy</code> for buy, <code>base_ccy</code> for sell
ccy	String	Margin currency Applicable to all <code>isolated</code> <code>MARGIN</code> orders and <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code> , <code>FUTURES</code> and <code>SWAP</code> contracts.
ordId	String	Order ID
clOrdId	String	Client Order ID as assigned by the client
tag	String	Order tag
px	String	Price For options, use coin as unit (e.g. BTC, ETH)
pxUsd	String	Options price in USDOnly applicable to options; return "" for other instrument types
pxVol	String	Implied volatility of the options orderOnly applicable to options; return "" for other instrument types
pxType	String	Price type of options <code>px</code> : Place an order based on price, in the unit of coin (the unit for the request parameter px is BTC or ETH) <code>pxVol</code> : Place an order based on pxVol <code>pxUsd</code> : Place an order based on pxUsd, in the unit of USD (the unit for the request parameter px is USD)
sz	String	Quantity to buy or sell
pnl	String	Profit and loss (excluding the fee). Applicable to orders which have a trade and aim to close position. It always is 0 in other conditions
ordType	String	Order type <code>market</code> : Market order <code>limit</code> : Limit order <code>post_only</code> : Post-only order <code>fok</code> : Fill-or-kill order <code>ioc</code> : Immediate-or-cancel order <code>optimal_limit_ioc</code> : Market order with immediate-or-cancel order <code>mmp</code> : Market Maker Protection (only applicable to Option in Portfolio Margin mode) <code>mmp_and_post_only</code> : Market Maker Protection and Post-only order(only applicable to Option in Portfolio

Parameter	Type	Description
		Margin mode) [op_fok]: Simple options (fok)
side	String	Order side
posSide	String	Position side
tdMode	String	Trade mode
accFillSz	String	Accumulated fill quantity
fillPx	String	Last filled price
tradId	String	Last trade ID
fillSz	String	Last filled quantity
fillTime	String	Last filled time
avgPx	String	Average filled price. If none is filled, it will return "".
state	String	State [live] [partially_filled]
lever	String	Leverage, from [0.01] to [125]. Only applicable to [MARGIN/FUTURES/SWAP]
attachAlgoClOrdId	String	Client-supplied Algo ID when placing order attaching TP/SL.
tpTriggerPx	String	Take-profit trigger price.
tpTriggerPxType	String	Take-profit trigger price type. [last]: last price [index]: index price [mark]: mark price
tpOrdPx	String	Take-profit order price.
slTriggerPx	String	Stop-loss trigger price.
slTriggerPxType	String	Stop-loss trigger price type. [last]: last price [index]: index price [mark]: mark price
slOrdPx	String	Stop-loss order price.
attachAlgoOrds	Array of objects	TP/SL information attached when placing order
> attachAlgId	String	The order ID of attached TP/SL order. It can be used to identify the TP/SL order when amending. It will not be posted to algId when placing TP/SL order after the general order is filled completely.
> attachAlgoClOrdId	String	Client-supplied Algo ID when placing order attaching TP/SL A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. It will be posted to [algoClOrdId] when placing TP/SL order once the general order is filled completely.
> tpOrdKind	String	TP order kind [condition] [limit]
> tpTriggerPx	String	Take-profit trigger price.

Parameter	Type	Description
> tpTriggerRatio	String	Take profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.
> tpTriggerPxType	String	Take-profit trigger price type. <input type="checkbox"/> <b>last</b> : last price <input type="checkbox"/> <b>index</b> : index price <input type="checkbox"/> <b>mark</b> : mark price
> tpOrdPx	String	Take-profit order price.
> slTriggerPx	String	Stop-loss trigger price.
> slTriggerRatio	String	Stop profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.
> slTriggerPxType	String	Stop-loss trigger price type. <input type="checkbox"/> <b>last</b> : last price <input type="checkbox"/> <b>index</b> : index price <input type="checkbox"/> <b>mark</b> : mark price
> slOrdPx	String	Stop-loss order price.
> sz	String	Size. Only applicable to TP order of split TPs
> amendPxOnTriggerType	String	Whether to enable Cost-price SL. Only applicable to SL order of split TPs. <input type="checkbox"/> <b>0</b> : disable, the default value <input type="checkbox"/> <b>1</b> : Enable
> failCode	String	The error code when failing to place TP/SL order, e.g. 51020 The default is ""
> failReason	String	The error reason when failing to place TP/SL order. The default is ""
linkedAlgoOrd	Object	Linked SL order detail, only applicable to the order that is placed by one-cancels-the-other (OCO) order that contains the TP limit order.
> algoid	String	Algo ID
stpId	String	<del>Self trade prevention ID Return "" if self trade prevention is not applicable (deprecated)</del>
stpMode	String	Self trade prevention mode
feeCcy	String	Fee currency For maker sell orders of Spot and Margin, this represents the quote currency. For all other cases, it represents the currency in which fees are charged.
fee	String	Fee amount For Spot and Margin (excluding maker sell orders): accumulated fee charged by the platform, always negative For maker sell orders in Spot and Margin, Expiry Futures, Perpetual Futures and Options: accumulated fee and rebate (always in quote currency for maker sell orders in Spot and Margin)
rebateCcy	String	Rebate currency For maker sell orders of Spot and Margin, this represents the base currency. For all other cases, it represents the currency in which rebates are paid.
rebate	String	Rebate amount, only applicable to Spot and Margin For maker sell orders: <del>Accumulated fee and</del> rebate amount in the unit of base currency. For all other cases, it represents the maker rebate amount, always positive, return "" if no rebate.
source	String	Order source <input type="checkbox"/> <b>6</b> : The normal order triggered by the <b>trigger order</b>

Parameter	Type	Description
		<p>7: The normal order triggered by the <a href="#">TP/SL order</a></p> <p>13: The normal order triggered by the algo order</p> <p>25: The normal order triggered by the <a href="#">trailing stop order</a></p> <p>34: The normal order triggered by the chase order</p>
category	String	Category <a href="#">normal</a>
reduceOnly	String	Whether the order can only reduce the position size. Valid options: true or false.
quickMgnType	String	Quick Margin type, Only applicable to Quick Margin Mode of isolated margin <a href="#">manual</a> , <a href="#">auto_borrow</a> , <a href="#">auto_repay</a>
algoClOrdId	String	Client-supplied Algo ID. There will be a value when algo order attaching <a href="#">algoClOrdId</a> is triggered, or it will be "".
algoid	String	Algo ID. There will be a value when algo order is triggered, or it will be "".
isTpLimit	String	Whether it is TP limit order. true or false
uTime	String	Update time, Unix timestamp format in milliseconds, e.g. <a href="#">1597026383085</a>
cTime	String	Creation time, Unix timestamp format in milliseconds, e.g. <a href="#">1597026383085</a>
cancelSource	String	Code of the cancellation source.
cancelSourceReason	String	Reason for the cancellation.
tradeQuoteCcy	String	The quote currency used for trading.

#### GET / ORDER HISTORY (LAST 7 DAYS)

Get completed orders which are placed in the last 7 days, including those placed 7 days ago but completed in the last 7 days.

The incomplete orders that have been canceled are only reserved for 2 hours.

**RATE LIMIT: 40 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

[GET /api/v5/trade/orders-history](#)

Request Example

```

import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Get completed SPOT orders which are placed in the last 7 days
# The incomplete orders that have been canceled are only reserved for 2 hours
result = tradeAPI.get_orders_history(
    instType="SPOT",
    ordType="post_only,fok,ioc"
)
print(result)

```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	yes	<p>Instrument type</p> <div style="display: flex; justify-content: space-around;"> <span>SPOT</span> <span>MARGIN</span> <span>SWAP</span> <span>FUTURES</span> <span>OPTION</span> </div>
instFamily	String	No	<p>Instrument family</p> <p>Applicable to <span>FUTURES</span>/<span>SWAP</span>/<span>OPTION</span></p>
instId	String	No	Instrument ID, e.g. <span>BTC-USDT</span>
ordType	String	No	<p>Order type</p> <p><span>market</span>: market order</p> <p><span>limit</span>: limit order</p> <p><span>post_only</span>: Post-only order</p> <p><span>fok</span>: Fill-or-kill order</p> <p><span>ioc</span>: Immediate-or-cancel order</p> <p><span>optimal_limit_ioc</span>: Market order with immediate-or-cancel order</p> <p><span>mmp</span>: Market Maker Protection (only applicable to Option in Portfolio Margin mode)</p> <p><span>mmp_and_post_only</span>: Market Maker Protection and Post-only order(only applicable to Option in Portfolio Margin mode)</p> <p><span>op_fok</span>: Simple options (fok)</p>
state	String	No	<p>State</p> <div style="display: flex; justify-content: space-around;"> <span>canceled</span> <span>filled</span> </div> <p><span>mmp_canceled</span>: Order canceled automatically due to Market Maker Protection</p>
category	String	No	<p>Category</p> <div style="display: flex; justify-content: space-around;"> <span>twap</span> <span>adl</span> </div> <div style="display: flex; justify-content: space-around;"> <span>full_liquidation</span> <span>partial_liquidation</span> </div> <div style="display: flex; justify-content: space-around;"> <span>delivery</span> <span>ddh</span>: Delta dynamic hedge       </div>
after	String	No	Pagination of data to return records earlier than the requested <span>ordId</span>
before	String	No	Pagination of data to return records newer than the requested <span>ordId</span>
begin	String	No	Filter with a begin timestamp <span>cTime</span> . Unix timestamp format in milliseconds, e.g. 1597026383085
end	String	No	Filter with an end timestamp <span>cTime</span> . Unix timestamp format in milliseconds, e.g. 1597026383085
limit	String	No	Number of results per request. The maximum is <span>100</span> ; The default is <span>100</span>

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "accFillSz": "0.00192834",
      "algoClOrdId": "",
      "algoId": "",
      "attachAlgoClOrdId": "",
      "attachAlgoOrds": [],
      "avgPx": "51858",
      "cTime": "1708587373361",
      "cancelSource": "",
      "cancelSourceReason": "",
      "category": "normal",
      "ccy": ""
    }
  ]
}
```

```

"clOrdId": "",  

"fee": "-0.00000192834",  

"feeCcy": "BTC",  

"fillPx": "51858",  

"fillSz": "0.00192834",  

"fillTime": "1708587373361",  

"instId": "BTC-USDT",  

"instType": "SPOT",  

"lever": "",  

"linkedAlgoOrd": {  

    "algoid": ""  

},  

"ordId": "680800019749904384",  

"ordType": "market",  

"pnl": "0",  

"posSide": "",  

"px": "",  

"pxType": "",  

"pxUsd": "",  

"pxVol": "",  

"quickMgnType": "",  

"rebate": "0",  

"rebateCcy": "USDT",  

"reduceOnly": "false",  

"side": "buy",  

"slOrdPx": "",  

"slTriggerPx": "",  

"slTriggerPxType": "",  

"source": "",  

"state": "filled",  

"stpId": "",  

"stpMode": "",  

"sz": "100",  

"tag": "",  

"tdMode": "cash",  

"tgtCcy": "quote_ccy",  

"tpOrdPx": "",  

"tpTriggerPx": "",  

"tpTriggerPxType": "",  

"tradeId": "744876980",  

"tradeQuoteCcy": "USDT",  

"uTime": "1708587373362",  

"isTpLimit": "false"  

}  

],  

"msg": ""  

}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type
instId	String	Instrument ID
tgtCcy	String	Order quantity unit setting for <code>sz</code> <code>base_ccy</code> : Base currency, <code>quote_ccy</code> : Quote currency Only applicable to <code>SPOT</code> Market Orders Default is <code>quote_ccy</code> for buy, <code>base_ccy</code> for sell
ccy	String	Margin currency Applicable to all <code>isolated</code> <code>MARGIN</code> orders and <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code> , <code>FUTURES</code> and <code>SWAP</code> contracts.
ordId	String	Order ID
clOrdId	String	Client Order ID as assigned by the client
tag	String	Order tag

Parameter	Type	Description
px	String	Price For options, use coin as unit (e.g. BTC, ETH)
pxUsd	String	Options price in USDOnly applicable to options; return "" for other instrument types
pxVol	String	Implied volatility of the options orderOnly applicable to options; return "" for other instrument types
pxType	String	Price type of options <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">px</div> : Place an order based on price, in the unit of coin (the unit for the request parameter px is BTC or ETH) <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">pxVol</div> : Place an order based on pxVol <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">pxUsd</div> : Place an order based on pxUsd, in the unit of USD (the unit for the request parameter px is USD)
sz	String	Quantity to buy or sell
ordType	String	Order type <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">market</div> : market order <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">limit</div> : limit order <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">post_only</div> : Post-only order <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">fok</div> : Fill-or-kill order <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">ioc</div> : Immediate-or-cancel order <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">optimal_limit_ioc</div> : Market order with immediate-or-cancel order <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">mmp</div> : Market Maker Protection (only applicable to Option in Portfolio Margin mode) <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">mmp_and_post_only</div> : Market Maker Protection and Post-only order(only applicable to Option in Portfolio Margin mode) <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">op_fok</div> : Simple options (fok)
side	String	Order side
posSide	String	Position side
tdMode	String	Trade mode
accFillSz	String	Accumulated fill quantity
fillPx	String	Last filled price. If none is filled, it will return "".
tradeld	String	Last trade ID
fillSz	String	Last filled quantity
fillTime	String	Last filled time
avgPx	String	Average filled price. If none is filled, it will return "".
state	String	State <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">canceled</div> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">filled</div> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">mmp_canceled</div>
lever	String	Leverage, from <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">0.01</div> to <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">125</div> . Only applicable to <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">MARGIN/FUTURES/SWAP</div>
attachAlgoCIOrld	String	Client-supplied Algo ID when placing order attaching TP/SL.
tpTriggerPx	String	Take-profit trigger price.
tpTriggerPxType	String	Take-profit trigger price type. <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">last</div> : last price <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">index</div> : index price <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">mark</div> : mark price

Parameter	Type	Description
tpOrdPx	String	Take-profit order price.
slTriggerPx	String	Stop-loss trigger price.
slTriggerPxType	String	Stop-loss trigger price type. <input type="checkbox"/> <code>last</code> : last price <input type="checkbox"/> <code>index</code> : index price <input type="checkbox"/> <code>mark</code> : mark price
slOrdPx	String	Stop-loss order price.
attachAlgoOrds	Array of objects	TP/SL information attached when placing order
> attachAlgId	String	The order ID of attached TP/SL order. It can be used to identify the TP/SL order when amending. It will not be posted to algId when placing TP/SL order after the general order is filled completely.
> attachAlgoClOrdId	String	Client-supplied Algo ID when placing order attaching TP/SL A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. It will be posted to <code>algoClOrdId</code> when placing TP/SL order once the general order is filled completely.
> tpOrdKind	String	TP order kind <input type="checkbox"/> <code>condition</code> <input type="checkbox"/> <code>limit</code>
> tpTriggerPx	String	Take-profit trigger price.
> tpTriggerRatio	String	Take profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.
> tpTriggerPxType	String	Take-profit trigger price type. <input type="checkbox"/> <code>last</code> : last price <input type="checkbox"/> <code>index</code> : index price <input type="checkbox"/> <code>mark</code> : mark price
> tpOrdPx	String	Take-profit order price.
> slTriggerPx	String	Stop-loss trigger price.
> slTriggerRatio	String	Stop profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.
> slTriggerPxType	String	Stop-loss trigger price type. <input type="checkbox"/> <code>last</code> : last price <input type="checkbox"/> <code>index</code> : index price <input type="checkbox"/> <code>mark</code> : mark price
> slOrdPx	String	Stop-loss order price.
> sz	String	Size. Only applicable to TP order of split TPs
> amendPxOnTriggerType	String	Whether to enable Cost-price SL. Only applicable to SL order of split TPs. <input type="checkbox"/> <code>0</code> : disable, the default value <input type="checkbox"/> <code>1</code> : Enable
> failCode	String	The error code when failing to place TP/SL order, e.g. 51020 The default is ""
> failReason	String	The error reason when failing to place TP/SL order. The default is ""
linkedAlgoOrd	Object	Linked SL order detail, only applicable to the order that is placed by one-cancels-the-other (OCO) order that contains the TP limit order.

Parameter	Type	Description
> algold	String	Algo ID
stpld	String	<b>Self trade prevention ID</b> Return "" if self trade prevention is not applicable (deprecated)
stpMode	String	Self trade prevention mode
feeCcy	String	Fee currency For maker sell orders of Spot and Margin, this represents the quote currency. For all other cases, it represents the currency in which fees are charged.
fee	String	Fee amount For Spot and Margin (excluding maker sell orders): accumulated fee charged by the platform, always negative For maker sell orders in Spot and Margin, Expiry Futures, Perpetual Futures and Options: accumulated fee and rebate (always in quote currency for maker sell orders in Spot and Margin)
rebateCcy	String	Rebate currency For maker sell orders of Spot and Margin, this represents the base currency. For all other cases, it represents the currency in which rebates are paid.
rebate	String	Rebate amount, only applicable to Spot and Margin For maker sell orders: <b>Accumulated fee and</b> rebate amount in the unit of base currency. For all other cases, it represents the maker rebate amount, always positive, return "" if no rebate.
source	String	Order source 6: The normal order triggered by the <b>trigger order</b> 7: The normal order triggered by the <b>TP/SL order</b> 13: The normal order triggered by the algo order 25: The normal order triggered by the <b>trailing stop order</b> 34: The normal order triggered by the chase order
pnl	String	Profit and loss (excluding the fee). Applicable to orders which have a trade and aim to close position. It always is 0 in other conditions
category	String	Category normal twap ad1 full_liquidation partial_liquidation delivery ddh: Delta dynamic hedge auto_conversion
reduceOnly	String	Whether the order can only reduce the position size. Valid options: true or false.
cancelSource	String	Code of the cancellation source.
cancelSourceReason	String	Reason for the cancellation.
algoClOrdId	String	Client-supplied Algo ID. There will be a value when algo order attaching <b>algoClOrdId</b> is triggered, or it will be "".
algold	String	Algo ID. There will be a value when algo order is triggered, or it will be "".
isTpLimit	String	Whether it is TP limit order. true or false
uTime	String	Update time, Unix timestamp format in milliseconds, e.g. <b>1597026383085</b>
cTime	String	Creation time, Unix timestamp format in milliseconds, e.g. <b>1597026383085</b>

Parameter	Type	Description
quickMgnType	String	Quick Margin type. Only applicable to Quick Margin Mode of isolated margin manual, auto_borrow, auto_repay (Deprecated)
tradeQuoteCcy	String	The quote currency used for trading.

#### GET / ORDER HISTORY (LAST 3 MONTHS)

Get completed orders which are placed in the last 3 months, including those placed 3 months ago but completed in the last 3 months.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

GET /api/v5/trade/orders-history-archive

Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Get completed SPOT orders which are placed in the last 3 months
result = tradeAPI.get_orders_history_archive(
    instType="SPOT",
    ordType="post_only,fok,ioc"
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	yes	Instrument type SPOT MARGIN SWAP FUTURES OPTION
instFamily	String	No	Instrument family Applicable to FUTURES/SWAP/OPTION
instId	String	No	Instrument ID, e.g. BTC-USD-200927
ordType	String	No	Order type market: Market order limit: Limit order post_only: Post-only order fok: Fill-or-kill order ioc: Immediate-or-cancel order optimal_limit_ioc: Market order with immediate-or-cancel order mmp: Market Maker Protection (only applicable to Option in Portfolio Margin mode) mmp_and_post_only: Market Maker Protection and Post-only order (only applicable to Option in Portfolio Margin mode) op_fok: Simple options (fok)

Parameter	Type	Required	Description
state	String	No	<p>State</p> <p><code>canceled</code></p> <p><code>filled</code></p> <p><code>mmp_canceled</code>: Order canceled automatically due to Market Maker Protection</p>
category	String	No	<p>Category</p> <p><code>twap</code></p> <p><code>adl</code></p> <p><code>full_liquidation</code></p> <p><code>partial_liquidation</code></p> <p><code>delivery</code></p> <p><code>ddh</code>: Delta dynamic hedge</p>
after	String	No	Pagination of data to return records earlier than the requested <code>ordId</code>
before	String	No	Pagination of data to return records newer than the requested <code>ordId</code>
begin	String	No	Filter with a begin timestamp <code>cTime</code> . Unix timestamp format in milliseconds, e.g. 1597026383085
end	String	No	Filter with an end timestamp <code>cTime</code> . Unix timestamp format in milliseconds, e.g. 1597026383085
limit	String	No	Number of results per request. The maximum is <code>100</code> ; The default is <code>100</code>

### Response Example

```
{
  "code": "0",
  "data": [
    {
      "accFillSz": "0.00192834",
      "algoClOrdId": "",
      "algoId": "",
      "attachAlgoClOrdId": "",
      "attachAlgoOrds": [],
      "avgPx": "51858",
      "cTime": "1708587373361",
      "cancelSource": "",
      "cancelSourceReason": "",
      "category": "normal",
      "ccy": "",
      "clOrdId": "",
      "fee": "-0.0000192834",
      "feeCcy": "BTC",
      "fillPx": "51858",
      "fillSz": "0.00192834",
      "fillTime": "1708587373361",
      "instId": "BTC-USDT",
      "instType": "SPOT",
      "lever": "",
      "ordId": "688800019749904384",
      "ordType": "market",
      "pnl": "0",
      "posSide": "",
      "px": "",
      "pxType": "",
      "pxUsd": "",
      "pxVol": "",
      "quickMgnType": "",
      "rebate": "0",
      "rebateCcy": "USDT",
      "reduceOnly": "false",
      "side": "buy",
      "slOrdPx": "",
      "slTriggerPx": "",
      "slTriggerPxType": "",
      "source": "",
      "state": "filled",
      "stpId": ""
    }
  ]
}
```

```

        "stpMode": "",
        "sz": "100",
        "tag": "",
        "tdMode": "cash",
        "tgtCcy": "quote_ccy",
        "tpOrdPx": "",
        "tpTriggerPx": "",
        "tpTriggerPxType": "",
        "tradeId": "744876980",
        "tradeQuoteCcy": "USDT",
        "uTime": "1708587373362",
        "isTpLimit": "false",
        "linkedAlgoOrd": {
            "algoId": ""
        }
    }
],
"msg": ""
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type
instId	String	Instrument ID
tgtCcy	String	Order quantity unit setting for <code>sz</code> <code>base_ccy</code> : Base currency, <code>quote_ccy</code> : Quote currency Only applicable to <code>SPOT</code> Market Orders Default is <code>quote_ccy</code> for buy, <code>base_ccy</code> for sell
ccy	String	Margin currency Applicable to all <code>isolated</code> <code>MARGIN</code> orders and <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code> , <code>FUTURES</code> and <code>SWAP</code> contracts.
ordId	String	Order ID
clOrdId	String	Client Order ID as assigned by the client
tag	String	Order tag
px	String	Price For options, use coin as unit (e.g. BTC, ETH)
pxUsd	String	Options price in USDOnly applicable to options; return "" for other instrument types
pxVol	String	Implied volatility of the options orderOnly applicable to options; return "" for other instrument types
pxType	String	Price type of options <code>px</code> : Place an order based on price, in the unit of coin (the unit for the request parameter px is BTC or ETH) <code>pxVol</code> : Place an order based on pxVol <code>pxUsd</code> : Place an order based on pxUsd, in the unit of USD (the unit for the request parameter px is USD)
sz	String	Quantity to buy or sell
ordType	String	Order type <code>market</code> : Market order <code>limit</code> : Limit order <code>post_only</code> : Post-only order <code>fok</code> : Fill-or-kill order <code>ioc</code> : Immediate-or-cancel order <code>optimal_limit_ioc</code> : Market order with immediate-or-cancel order <code>mmp</code> : Market Maker Protection (only applicable to Option in Portfolio Margin mode) <code>mmp_and_post_only</code> : Market Maker Protection and Post-only order(only applicable to Option in Portfolio

Parameter	Type	Description
		Margin mode) op_fok: Simple options (fok)
side	String	Order side
posSide	String	Position side
tdMode	String	Trade mode
accFillSz	String	Accumulated fill quantity
fillPx	String	Last filled price. If none is filled, it will return "".
tradeld	String	Last trade ID
fillSz	String	Last filled quantity
fillTime	String	Last filled time
avgPx	String	Average filled price. If none is filled, it will return "".
state	String	State canceled filled mmp_canceled
lever	String	Leverage, from 0.01 to 125. Only applicable to MARGIN/FUTURES/SWAP
attachAlgoClOrdId	String	Client-supplied Algo ID when placing order attaching TP/SL.
tpTriggerPx	String	Take-profit trigger price.
tpTriggerPxType	String	Take-profit trigger price type. last: last price index: index price mark: mark price
tpOrdPx	String	Take-profit order price.
slTriggerPx	String	Stop-loss trigger price.
slTriggerPxType	String	Stop-loss trigger price type. last: last price index: index price mark: mark price
slOrdPx	String	Stop-loss order price.
attachAlgoOrds	Array of objects	TP/SL information attached when placing order
> attachAlgId	String	The order ID of attached TP/SL order. It can be used to identify the TP/SL order when amending. It will not be posted to algold when placing TP/SL order after the general order is filled completely.
> attachAlgoClOrdId	String	Client-supplied Algo ID when placing order attaching TP/SL A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. It will be posted to algoclordId when placing TP/SL order once the general order is filled completely.
> tpOrdKind	String	TP order kind condition limit

Parameter	Type	Description
> tpTriggerPx	String	Take-profit trigger price.
> tpTriggerRatio	String	Take profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.
> tpTriggerPxType	String	Take-profit trigger price type. <input type="checkbox"/> <code>last</code> : last price <input type="checkbox"/> <code>index</code> : index price <input type="checkbox"/> <code>mark</code> : mark price
> tpOrdPx	String	Take-profit order price.
> slTriggerPx	String	Stop-loss trigger price.
> slTriggerRatio	String	Stop profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.
> slTriggerPxType	String	Stop-loss trigger price type. <input type="checkbox"/> <code>last</code> : last price <input type="checkbox"/> <code>index</code> : index price <input type="checkbox"/> <code>mark</code> : mark price
> slOrdPx	String	Stop-loss order price.
> sz	String	Size. Only applicable to TP order of split TPs
> amendPxOnTriggerType	String	Whether to enable Cost-price SL. Only applicable to SL order of split TPs. <input type="checkbox"/> <code>0</code> : disable, the default value <input type="checkbox"/> <code>1</code> : Enable
> failCode	String	The error code when failing to place TP/SL order, e.g. 51020 The default is ""
> failReason	String	The error reason when failing to place TP/SL order. The default is ""
linkedAlgoOrd	Object	Linked SL order detail, only applicable to the order that is placed by one-cancels-the-other (OCO) order that contains the TP limit order.
> algoid	String	Algo ID
stpld	String	<del>Self trade prevention ID</del> <del>Return "" if self trade prevention is not applicable (deprecated)</del>
stpMode	String	Self trade prevention mode
feeCcy	String	Fee currency For maker sell orders of Spot and Margin, this represents the quote currency. For all other cases, it represents the currency in which fees are charged.
fee	String	Fee amount For Spot and Margin (excluding maker sell orders): accumulated fee charged by the platform, always negative For maker sell orders in Spot and Margin, Expiry Futures, Perpetual Futures and Options: accumulated fee and rebate (always in quote currency for maker sell orders in Spot and Margin)
rebateCcy	String	Rebate currency For maker sell orders of Spot and Margin, this represents the base currency. For all other cases, it represents the currency in which rebates are paid.
rebate	String	Rebate amount, only applicable to Spot and Margin For maker sell orders: <del>Accumulated fee and</del> rebate amount in the unit of base currency. For all other cases, it represents the maker rebate amount, always positive, return "" if no rebate.

Parameter	Type	Description
source	String	Order source 6: The normal order triggered by the <a href="#">trigger order</a> 7: The normal order triggered by the <a href="#">TP/SL order</a> 13: The normal order triggered by the algo order 25: The normal order triggered by the <a href="#">trailing stop order</a> 34: The normal order triggered by the <a href="#">chase order</a>
pnl	String	Profit and loss (excluding the fee). Applicable to orders which have a trade and aim to close position. It always is 0 in other conditions
category	String	Category <a href="#">normal</a> <a href="#">twap</a> <a href="#">ad1</a> <a href="#">full_liquidation</a> <a href="#">partial_liquidation</a> <a href="#">delivery</a> ddh: Delta dynamic hedge <a href="#">auto_conversion</a>
reduceOnly	String	Whether the order can only reduce the position size. Valid options: true or false.
cancelSource	String	Code of the cancellation source.
cancelSourceReason	String	Reason for the cancellation.
algoClOrdId	String	Client-supplied Algo ID. There will be a value when algo order attaching <a href="#">algoClOrdId</a> is triggered, or it will be "".
algold	String	Algo ID. There will be a value when algo order is triggered, or it will be "".
isTpLimit	String	Whether it is TP limit order. true or false
uTime	String	Update time, Unix timestamp format in milliseconds, e.g. <a href="#">1597026383085</a>
cTime	String	Creation time, Unix timestamp format in milliseconds, e.g. <a href="#">1597026383085</a>
quickMgnType	String	Quick Margin type, Only applicable to Quick Margin Mode of isolated margin <a href="#">manual</a> - <a href="#">auto_borrow</a> - <a href="#">auto_repay</a> (Deprecated)
tradeQuoteCcy	String	The quote currency used for trading.

This interface does not contain the order data of the 'Canceled orders without any fills' type, which can be obtained through the 'Get Order History (last 7 days)' interface.

As far as OPTION orders that are complete, pxVol and pxUsd will update in time for px order, pxVol will update in time for pxUsd order, pxUsd will update in time for pxVol order.

#### GET / TRANSACTION DETAILS (LAST 3 DAYS)

Retrieve recently-filled transaction details in the last 3 day.

#### RATE LIMIT: 60 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: USER ID

#### PERMISSION: READ

#### HTTP REQUEST

GET /api/v5/trade/fills

## Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Retrieve recently-filled transaction details
result = tradeAPI.get_fills()
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type <input type="button" value="SPOT"/> <input type="button" value="MARGIN"/> <input type="button" value="SWAP"/> <input type="button" value="FUTURES"/> <input type="button" value="OPTION"/>
instFamily	String	No	Instrument family Applicable to <input type="button" value="FUTURES"/> / <input type="button" value="SWAP"/> / <input type="button" value="OPTION"/>
instId	String	No	Instrument ID, e.g. <input type="button" value="BTC-USDT"/>
ordId	String	No	Order ID
subType	String	No	Transaction type <input type="button" value="1: Buy"/> <input type="button" value="2: Sell"/> <input type="button" value="3: Open long"/> <input type="button" value="4: Open short"/> <input type="button" value="5: Close long"/> <input type="button" value="6: Close short"/> <input type="button" value="100: Partial liquidation close long"/> <input type="button" value="101: Partial liquidation close short"/> <input type="button" value="102: Partial liquidation buy"/> <input type="button" value="103: Partial liquidation sell"/> <input type="button" value="104: Liquidation long"/> <input type="button" value="105: Liquidation short"/> <input type="button" value="106: Liquidation buy"/> <input type="button" value="107: Liquidation sell"/> <input type="button" value="110: Liquidation transfer in"/> <input type="button" value="111: Liquidation transfer out"/> <input type="button" value="118: System token conversion transfer in"/> <input type="button" value="119: System token conversion transfer out"/> <input type="button" value="112: Delivery long"/> <input type="button" value="113: Delivery short"/> <input type="button" value="125: ADL close long"/> <input type="button" value="126: ADL close short"/> <input type="button" value="127: ADL buy"/> <input type="button" value="128: ADL sell"/> <input type="button" value="212: Auto borrow of quick margin"/> <input type="button" value="213: Auto repay of quick margin"/> <input type="button" value="204: block trade buy"/> <input type="button" value="205: block trade sell"/> <input type="button" value="206: block trade open long"/> <input type="button" value="207: block trade open short"/>

Parameter	Type	Required	Description
			<ul style="list-style-type: none"> <li>[208]: block trade close long</li> <li>[209]: block trade close short</li> <li>[236]: Easy convert in</li> <li>[237]: Easy convert out</li> <li>[270]: Spread trading buy</li> <li>[271]: Spread trading sell</li> <li>[272]: Spread trading open long</li> <li>[273]: Spread trading open short</li> <li>[274]: Spread trading close long</li> <li>[275]: Spread trading close short</li> <li>[324]: Move position buy</li> <li>[325]: Move position sell</li> <li>[326]: Move position open long</li> <li>[327]: Move position open short</li> <li>[328]: Move position close long</li> <li>[329]: Move position close short</li> <li>[376]: Collateralized borrowing auto conversion buy</li> <li>[377]: Collateralized borrowing auto conversion sell</li> </ul>
after	String	No	Pagination of data to return records earlier than the requested <code>billId</code>
before	String	No	Pagination of data to return records newer than the requested <code>billId</code>
begin	String	No	Filter with a begin timestamp <code>ts</code> . Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
end	String	No	Filter with an end timestamp <code>ts</code> . Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. The maximum is <code>100</code> ; The default is <code>100</code>

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "side": "buy",
      "fillSz": "0.00192834",
      "fillPx": "51858",
      "fillPxVol": "",
      "fillFwdPx": "",
      "fee": "-0.0000192834",
      "fillPnl": "0",
      "ordId": "680800019749904384",
      "feeRate": "-0.001",
      "instType": "SPOT",
      "fillPxUsd": "",
      "instId": "BTC-USDT",
      "clOrdId": "",
      "posSide": "net",
      "billId": "680800019754098688",
      "subType": "1",
      "fillMarkVol": "",
      "tag": "",
      "fillTime": "1708587373361",
      "execType": "T",
      "fillIdxPx": "",
      "tradeId": "744876980",
      "fillMarkPx": "",
      "feeCcy": "BTC",
      "ts": "1708587373362",
      "tradeQuoteCcy": "USDT"
    }
  ],
  "msg": ""
}
```

Parameter	Type	Description
instType	String	Instrument type
instId	String	Instrument ID
tradId	String	Last trade ID
ordId	String	Order ID
clOrdId	String	Client Order ID as assigned by the client
billId	String	Bill ID
subType	String	Transaction type
tag	String	Order tag
fillPx	String	Last filled price. It is the same as the px from "Get bills details".
fillSz	String	Last filled quantity
fillIdxPx	String	Index price at the moment of trade execution For cross currency spot pairs, it returns baseCcy-USDT index price. For example, for LTC-ETH, this field returns the index price of LTC-USDT.
fillPnl	String	Last filled profit and loss, applicable to orders which have a trade and aim to close position. It always is 0 in other conditions
fillPxVol	String	Implied volatility when filled Only applicable to options; return "" for other instrument types
fillPxUsd	String	Options price when filled, in the unit of USD Only applicable to options; return "" for other instrument types
fillMarkVol	String	Mark volatility when filled Only applicable to options; return "" for other instrument types
fillFwdPx	String	Forward price when filled Only applicable to options; return "" for other instrument types
fillMarkPx	String	Mark price when filled Applicable to <b>FUTURES</b> , <b>SWAP</b> , <b>OPTION</b>
side	String	Order side, <b>buy</b> <b>sell</b>
posSide	String	Position side <b>long</b> <b>short</b> it returns <b>net</b> in <b>net</b> mode.
execType	String	Liquidity taker or maker <b>T</b> : taker <b>M</b> : maker Not applicable to system orders such as ADL and liquidation
feeCcy	String	Trading fee or rebate currency
fee	String	The amount of trading fee or rebate. The trading fee deduction is negative, such as '-0.01'; the rebate is positive, such as '0.01'.
ts	String	Data generation time, Unix timestamp format in milliseconds, e.g. <b>1597026383085</b> .
fillTime	String	Trade time which is the same as <b>fillTime</b> for the order channel.
feeRate	String	Fee rate. This field is returned for <b>SPOT</b> and <b>MARGIN</b> only

Parameter	Type	Description
tradeQuoteCcy	String	The quote currency for trading.

tradeld

For partial\_liquidation, full\_liquidation, or adl, when it comes to fill information, this field will be assigned a negative value to distinguish it from other matching transaction scenarios, when it comes to order information, this field will be 0.

ordId

Order ID, always "" for block trading.

clOrdId

Client-supplied order ID, always "" for block trading.

#### GET / TRANSACTION DETAILS (LAST 3 MONTHS)

This endpoint can retrieve data from the last 3 months.

**RATE LIMIT: 10 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

GET /api/v5/trade/fills-history

Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Retrieve SPOT transaction details in the last 3 months.
result = tradeAPI.get_fills_history(
    instType="SPOT"
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	YES	<p>Instrument type</p> <div style="display: flex; justify-content: space-around;"> <span>SPOT</span> <span>MARGIN</span> <span>SWAP</span> <span>FUTURES</span> <span>OPTION</span> </div>
instFamily	String	No	<p>Instrument family</p> <p>Applicable to <span>FUTURES</span> / <span>SWAP</span> / <span>OPTION</span></p>
instId	String	No	Instrument ID, e.g. <span>BTC-USDT</span>

Parameter	Type	Required	Description
ordId	String	No	Order ID
			Transaction type <input type="radio"/> 1: Buy <input type="radio"/> 2: Sell <input type="radio"/> 3: Open long <input type="radio"/> 4: Open short <input type="radio"/> 5: Close long <input type="radio"/> 6: Close short <input type="radio"/> 100: Partial liquidation close long <input type="radio"/> 101: Partial liquidation close short <input type="radio"/> 102: Partial liquidation buy <input type="radio"/> 103: Partial liquidation sell <input type="radio"/> 104: Liquidation long <input type="radio"/> 105: Liquidation short <input type="radio"/> 106: Liquidation buy <input type="radio"/> 107: Liquidation sell <input type="radio"/> 110: Liquidation transfer in <input type="radio"/> 111: Liquidation transfer out <input type="radio"/> 118: System token conversion transfer in <input type="radio"/> 119: System token conversion transfer out <input type="radio"/> 112: Delivery long <input type="radio"/> 113: Delivery short <input type="radio"/> 125: ADL close long <input type="radio"/> 126: ADL close short <input type="radio"/> 127: ADL buy <input type="radio"/> 128: ADL sell <input type="radio"/> 212: Auto borrow of quick margin <input type="radio"/> 213: Auto repay of quick margin <input type="radio"/> 204: block trade buy <input type="radio"/> 205: block trade sell <input type="radio"/> 206: block trade open long <input type="radio"/> 207: block trade open short <input type="radio"/> 208: block trade close long <input type="radio"/> 209: block trade close short <input type="radio"/> 236: Easy convert in <input type="radio"/> 237: Easy convert out <input type="radio"/> 270: Spread trading buy <input type="radio"/> 271: Spread trading sell <input type="radio"/> 272: Spread trading open long <input type="radio"/> 273: Spread trading open short <input type="radio"/> 274: Spread trading close long <input type="radio"/> 275: Spread trading close short <input type="radio"/> 324: Move position buy <input type="radio"/> 325: Move position sell <input type="radio"/> 326: Move position open long <input type="radio"/> 327: Move position open short <input type="radio"/> 328: Move position close long <input type="radio"/> 329: Move position close short <input type="radio"/> 376: Collateralized borrowing auto conversion buy <input type="radio"/> 377: Collateralized borrowing auto conversion sell
subType	String	No	
after	String	No	Pagination of data to return records earlier than the requested <code>billId</code>
before	String	No	Pagination of data to return records newer than the requested <code>billId</code>
begin	String	No	Filter with a begin timestamp <code>ts</code> . Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
end	String	No	Filter with an end timestamp <code>ts</code> . Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. The maximum is <code>100</code> ; The default is <code>100</code>

## Response Example

```
{  
  "code": "0",  
  "data": [  
    {  
      "side": "buy",  
      "fillSz": "0.00192834",  
      "fillPx": "51858",  
      "fillPxVol": "",  
      "fillFwdPx": "",  
      "fee": "-0.0000192834",  
      "fillPnl": "0",  
      "ordId": "680800019749904384",  
      "feeRate": "-0.001",  
      "instType": "SPOT",  
      "fillPxUsd": "",  
      "instId": "BTC-USDT",  
      "clOrdId": "",  
      "posSide": "net",  
      "billId": "680800019754098688",  
      "subType": "1",  
      "fillMarkVol": "",  
      "tag": "",  
      "fillTime": "1708587373361",  
      "execType": "T",  
      "fillIdxPx": "",  
      "tradeId": "744876980",  
      "fillMarkPx": "",  
      "feeCcy": "BTC",  
      "ts": "1708587373362",  
      "tradeQuoteCcy": "USDT"  
    }  
  ],  
  "msg": ""  
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type
instId	String	Instrument ID
tradeId	String	Last trade ID
ordId	String	Order ID
clOrdId	String	Client Order ID as assigned by the client
billId	String	Bill ID
subType	String	Transaction type
tag	String	Order tag
fillPx	String	Last filled price
fillSz	String	Last filled quantity
fillIdxPx	String	Index price at the moment of trade execution For cross currency spot pairs, it returns baseCcy-USDT index price. For example, for LTC-ETH, this field returns the index price of LTC-USDT.
fillPnl	String	Last filled profit and loss, applicable to orders which have a trade and aim to close position. It always is 0 in other conditions
fillPxVol	String	Implied volatility when filled Only applicable to options; return "" for other instrument types

Parameter	Type	Description
fillPxUsd	String	Options price when filled, in the unit of USD Only applicable to options; return "" for other instrument types
fillMarkVol	String	Mark volatility when filled Only applicable to options; return "" for other instrument types
fillFwdPx	String	Forward price when filled Only applicable to options; return "" for other instrument types
fillMarkPx	String	Mark price when filled Applicable to <b>FUTURES</b> , <b>SWAP</b> , <b>OPTION</b>
side	String	Order side <b>buy</b> <b>sell</b>
posSide	String	Position side <b>long</b> <b>short</b> it returns <b>net</b> in <b>net</b> mode.
execType	String	Liquidity taker or maker <b>T</b> : taker <b>M</b> : maker Not applicable to system orders such as ADL and liquidation
feeCcy	String	Trading fee or rebate currency
fee	String	The amount of trading fee or rebate. The trading fee deduction is negative, such as '-0.01'; the rebate is positive, such as '0.01'.
ts	String	Data generation time, Unix timestamp format in milliseconds, e.g. <b>1597026383085</b> .
fillTime	String	Trade time which is the same as <b>fillTime</b> for the order channel.
feeRate	String	Fee rate. This field is returned for <b>SPOT</b> and <b>MARGIN</b> only
tradeQuoteCcy	String	The quote currency for trading.

#### tradId

When the order category to which the transaction details belong is `partial_liquidation`, `full_liquidation`, or `adl`, this field will be assigned a negative value to distinguish it from other matching transaction scenarios.

#### ordId

Order ID, always "" for block trading.

#### c1OrdId

Client-supplied order ID, always "" for block trading.

We advise you to use Get Transaction details (last 3 days) when you request data for recent 3 days.

#### GET / EASY CONVERT CURRENCY LIST

Get list of small convertibles and mainstream currencies. Only applicable to the crypto balance less than \$10.

#### RATE LIMIT: 1 REQUEST PER 2 SECONDS

## PERMISSION: READ

## HTTP REQUEST

GET /api/v5/trade/easy-convert-currency-list

## Request Example

```

import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Get list of small convertibles and mainstream currencies
result = tradeAPI.get_easy_convert_currency_list()
print(result)

```

## REQUEST PARAMETERS

Parameters	Type	Required	Description
source	String	No	Funding source 1: Trading account 2: Funding account The default is 1.

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "fromData": [
        {
          "fromAmt": "6.580712708344864",
          "fromCcy": "ADA"
        },
        {
          "fromAmt": "2.9970000013055097",
          "fromCcy": "USDC"
        }
      ],
      "toCcy": [
        "USDT",
        "BTC",
        "ETH",
        "OKB"
      ]
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
fromData	Array of objects	Currently owned and convertible small currency list
> fromCcy	String	Type of small payment currency convert from, e.g. BTC
> fromAmt	String	Amount of small payment currency convert from

Parameter	Type	Description
toCcy	Array of strings	Type of mainstream currency convert to, e.g. <code>USDT</code>

#### POST / PLACE EASY CONVERT

Convert small currencies to mainstream currencies.

**RATE LIMIT: 1 REQUEST PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

`POST /api/v5/trade/easy-convert`

#### Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Convert small currencies to mainstream currencies
result = tradeAPI.easy_convert(
    fromCcy=["ADA", "USDC"],
    toCcy="OKB"
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
fromCcy	Array of strings	Yes	Type of small payment currency convert from Maximum 5 currencies can be selected in one order. If there are multiple currencies, separate them with commas.
toCcy	String	Yes	Type of mainstream currency convert to Only one receiving currency type can be selected in one order and cannot be the same as the small payment currencies.
source	String	No	Funding source ①: Trading account ②: Funding account The default is ①.

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "fillFromSz": "6.5807127",
      "fillToSz": "0.17171580105126",
      "fromCcy": "ADA",
      "status": "running",
      "toCcy": "OKB",
      "uTime": "1661419684687"
    },
    {
      "fillFromSz": "2.997",
      "fillToSz": "0.1683755161661844",
      "fromCcy": "ADA",
      "status": "running",
      "toCcy": "OKB",
      "uTime": "1661419684687"
    }
  ]
}
```

```

        "fromCcy": "USDC",
        "status": "running",
        "toCcy": "OKB",
        "uTime": "1661419684687"
    },
],
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
status	String	Current status of easy convert <span>running</span> : Running <span>filled</span> : Filled <span>failed</span> : Failed
fromCcy	String	Type of small payment currency convert from
toCcy	String	Type of mainstream currency convert to
fillFromSz	String	Filled amount of small payment currency convert from
fillToSz	String	Filled amount of mainstream currency convert to
uTime	String	Trade time, Unix timestamp format in milliseconds, e.g. 1597026383085

## GET / EASY CONVERT HISTORY

Get the history and status of easy convert trades in the past 7 days.

**RATE LIMIT: 1 REQUEST PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

### HTTP REQUEST

```
GET /api/v5/trade/easy-convert-history
```

#### Request Example

```

import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Get the history of easy convert trades
result = tradeAPI.get_easy_convert_history()
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
after	String	No	Pagination of data to return records earlier than the requested time (exclude), Unix timestamp format in milliseconds, e.g. <span>1597026383085</span>
before	String	No	Pagination of data to return records newer than the requested time (exclude), Unix timestamp format in milliseconds, e.g. <span>1597026383085</span>

Parameter	Type	Required	Description
limit	String	No	Number of results per request. The maximum is 100. The default is 100.

Response Example

```
{
  "code": "0",
  "data": [
    {
      "fillFromSz": "0.1761712511667539",
      "fillToSz": "6.7342205900000000",
      "fromCcy": "OKB",
      "status": "filled",
      "toCcy": "ADA",
      "acct": "18",
      "uTime": "1661313307979"
    },
    {
      "fillFromSz": "0.1722106121112177",
      "fillToSz": "2.9971018300000000",
      "fromCcy": "OKB",
      "status": "filled",
      "toCcy": "USDC",
      "acct": "18",
      "uTime": "1661313307979"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
fromCcy	String	Type of small payment currency convert from
fillFromSz	String	Amount of small payment currency convert from
toCcy	String	Type of mainstream currency convert to
fillToSz	String	Amount of mainstream currency convert to
acct	String	The account where the mainstream currency is located ⑥: Funding account ⑯: Trading account
status	String	Current status of easy convert ⑥: Running ⑦: Filled ⑧: Failed
uTime	String	Trade time, Unix timestamp format in milliseconds, e.g. 1597026383085

## GET / ONE-CLICK REPAY CURRENCY LIST

Get list of debt currency data and repay currencies. Debt currencies include both cross and isolated debts. Only applicable to [Multi-currency margin](#) / [Portfolio margin](#).

### RATE LIMIT: 1 REQUEST PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: READ

### HTTP REQUEST

```
GET /api/v5/trade/one-click-repay-currency-list
```

## Request Example

```

import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Get list of debt currency data and repay currencies
result = tradeAPI.get_oneclick_repay_list()
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
debtType	String	No	Debt type cross: cross isolated: isolated

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "debtData": [
        {
          "debtAmt": "29.653478",
          "debtCcy": "LTC"
        },
        {
          "debtAmt": "237803.6828295906051002",
          "debtCcy": "USDT"
        }
      ],
      "debtType": "cross",
      "repayData": [
        {
          "repayAmt": "0.4978335419825104",
          "repayCcy": "ETH"
        }
      ]
    },
    "msg": ""
  }
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
debtData	Array of objects	Debt currency data list
> debtCcy	String	Debt currency
> debtAmt	String	Debt currency amount Including principal and interest
debtType	String	Debt type cross: cross isolated: isolated
repayData	Array of objects	Repay currency data list
> repayCcy	String	Repay currency

Parameter	Type	Description
> repayAmt	String	Repay currency's available balance amount

#### POST / TRADE ONE-CLICK REPAY

Trade one-click repay to repay cross debts. Isolated debts are not applicable. The maximum repayment amount is based on the remaining available balance of funding and trading accounts. Only applicable to [Multi-currency margin](#)/[Portfolio margin](#).

**RATE LIMIT: 1 REQUEST PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

`POST /api/v5/trade/one-click-repay`

Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Trade one-click repay to repay cross debts
result = tradeAPI.oneclick_repay(
    debtCcy=["ETH", "BTC"],
    repayCcy="USDT"
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
debtCcy	Array of strings	Yes	Debt currency type Maximum 5 currencies can be selected in one order. If there are multiple currencies, separate them with commas.
repayCcy	String	Yes	Repay currency type Only one receiving currency type can be selected in one order and cannot be the same as the small payment currencies.

Response Example

```
{
  "code": "0",
  "data": [
    {
      "debtCcy": "ETH",
      "fillDebtSz": "0.01023052",
      "fillRepaySz": "30",
      "repayCcy": "USDT",
      "status": "filled",
      "uTime": "1646188520338"
    },
    {
      "debtCcy": "BTC",
      "fillFromSz": "3",
      "fillToSz": "60,221.15910001",
      "repayCcy": "USDT",
      "status": "filled",
      "uTime": "1646188520338"
    }
  ]
}
```

```
],
"msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
status	String	Current status of one-click repay <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">running</div> : Running <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">filled</div> : Filled <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">failed</div> : Failed
debtCcy	String	Debt currency type
repayCcy	String	Repay currency type
fillDebtSz	String	Filled amount of debt currency
fillRepaySz	String	Filled amount of repay currency
uTime	String	Trade time, Unix timestamp format in milliseconds, e.g. 1597026383085

## GET / ONE-CLICK REPAY HISTORY

Get the history and status of one-click repay trades in the past 7 days. Only applicable to [Multi-currency margin](#)/[Portfolio margin](#).

**RATE LIMIT: 1 REQUEST PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/trade/one-click-repay-history`

Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Get the history of one-click repay trades
result = tradeAPI.oneclick_repay_history()
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
after	String	No	Pagination of data to return records earlier than the requested time, Unix timestamp format in milliseconds, e.g. 1597026383085
before	String	No	Pagination of data to return records newer than the requested time, Unix timestamp format in milliseconds, e.g. 1597026383085
limit	String	No	Number of results per request. The maximum is 100. The default is 100.

Response Example

```
{
  "code": "0",
  "data": [
    {
      "debtCcy": "USDC",
      "fillDebtSz": "6950.4865447900000000",
      "fillRepaySz": "4.3067975995094930",
      "repayCcy": "ETH",
      "status": "filled",
      "uTime": "1661256148746"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
debtCcy	String	Debt currency type
fillDebtSz	String	Amount of debt currency transacted
repayCcy	String	Repay currency type
fillRepaySz	String	Amount of repay currency transacted
status	String	Current status of one-click repay running: Running filled: Filled failed: Failed
uTime	String	Trade time, Unix timestamp format in milliseconds, e.g. 1597026383085

## GET / ONE-CLICK REPAY CURRENCY LIST (NEW)

Get list of debt currency data and repay currencies. Only applicable to [SPOT mode](#).

**RATE LIMIT: 1 REQUEST PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/trade/one-click-repay-currency-list-v2`

Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"
flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag, debug=True)
result = tradeAPI.get_oneclick_repay_list_v2()
print(result)
```

Response Example

```
{
  "code": "0",
  "data": [
    {
      "debtData": [
        {

```

```

        "debtAmt": "100",
        "debtCcy": "USDC"
    },
    ],
    "repayData": [
        {
            "repayAmt": "1.000022977",
            "repayCcy": "BTC"
        },
        {
            "repayAmt": "4998.0002397",
            "repayCcy": "USDT"
        },
        {
            "repayAmt": "100",
            "repayCcy": "OKB"
        },
        {
            "repayAmt": "1",
            "repayCcy": "ETH"
        },
        {
            "repayAmt": "100",
            "repayCcy": "USDC"
        }
    ]
},
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
debtData	Array of objects	Debt currency data list
> debtCcy	String	Debt currency
> debtAmt	String	Debt currency amount Including principal and interest
repayData	Array of objects	Repay currency data list
> repayCcy	String	Repay currency
> repayAmt	String	Repay currency's available balance amount

## POST / TRADE ONE-CLICK REPAY (NEW)

Trade one-click repay to repay debts. Only applicable to `SPOT mode`.

### RATE LIMIT: 1 REQUEST PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: TRADE

### HTTP REQUEST

`POST /api/v5/trade/one-click-repay-v2`

#### Request Example

```

import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"
flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag, debug=True)

```

```
result = tradeAPI.oneclick_repay_v2("USDC", ["USDC", "BTC"])
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
debtCcy	String	Yes	Debt currency
repayCcyList	Array of strings	Yes	Repay currency list, e.g. ["USDC", "BTC"] The priority of currency to repay is consistent with the order in the array. (The first item has the highest priority)

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "debtCcy": "USDC",
      "repayCcyList": [
        "USDC",
        "BTC"
      ],
      "ts": "1742192217514"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
debtCcy	String	Debt currency
repayCcyList	Array of strings	Repay currency list, e.g. ["USDC", "BTC"]
ts	String	Request time, Unix timestamp format in milliseconds, e.g. 1597026383085

## GET / ONE-CLICK REPAY HISTORY (NEW)

Get the history and status of one-click repay trades in the past 7 days. Only applicable to SPOT mode.

### RATE LIMIT: 1 REQUEST PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: READ

### HTTP REQUEST

```
GET /api/v5/trade/one-click-repay-history-v2
```

## Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"
flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)
result = tradeAPI.oneclick_repay_history_v2()
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
after	String	No	Pagination of data to return records earlier than (included) the requested time <code>ts</code> , Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
before	String	No	Pagination of data to return records newer than (included) the requested time <code>ts</code> , Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. The maximum is 100. The default is 100.

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "debtCcy": "USDC",
      "fillDebtSz": "9.079631989",
      "ordIdInfo": [
        {
          "cTime": "1742194485439",
          "fillPx": "1",
          "fillSz": "9.088651",
          "instId": "USDC-USDT",
          "ordId": "2338478342062235648",
          "ordType": "ioc",
          "px": "1.0049",
          "side": "buy",
          "state": "filled",
          "sz": "9.0886514537313433"
        },
        {
          "cTime": "1742194482326",
          "fillPx": "83271.9",
          "fillSz": "0.00010969",
          "instId": "BTC-USDT",
          "ordId": "2338478237607288832",
          "ordType": "ioc",
          "px": "82856.7",
          "side": "sell",
          "state": "filled",
          "sz": "0.000109696512171"
        }
      ],
      "repayCcyList": [
        "USDC",
        "BTC"
      ],
      "status": "filled",
      "ts": "1742194481852"
    },
    {
      "debtCcy": "USDC",
      "fillDebtSz": "100",
      "ordIdInfo": [],
      "repayCcyList": [
        "USDC",
        "BTC"
      ],
      "status": "filled",
      "ts": "1742192217511"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
debtCcy	String	Debt currency

Parameter	Type	Description
repayCcyList	Array of strings	Repay currency list, e.g. ["USDC", "BTC"]
fillDebtSz	String	Amount of debt currency transacted
status	String	Current status of one-click repay <span>running</span> : Running <span>filled</span> : Filled <span>failed</span> : Failed
ordIdInfo	Array of objects	Order info
> ordId	String	Order ID
> instId	String	Instrument ID, e.g. <span>BTC-USDT</span>
> ordType	String	Order type <span>ioc</span> : Immediate-or-cancel order
> side	String	Side <span>buy</span> <span>sell</span>
> px	String	Price
> sz	String	Quantity to buy or sell
> fillPx	String	Last filled price. If none is filled, it will return "".
> fillSz	String	Last filled quantity
> state	String	State <span>filled</span> <span>cancelled</span>
> cTime	String	Creation time for order, Unix timestamp format in milliseconds, e.g. <span>1597026383085</span>
ts	String	Request time, Unix timestamp format in milliseconds, e.g. <span>1597026383085</span>

#### POST / MASS CANCEL ORDER

Cancel all the MMP pending orders of an instrument family.

Only applicable to Option in Portfolio Margin mode, and MMP privilege is required.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

POST /api/v5/trade/mass-cancel

Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	Yes	Instrument type <span>OPTION</span>
instFamily	String	Yes	Instrument family

Parameter	Type	Required	Description
lockInterval	String	No	Lock interval(ms) The range should be [0, 10 000] The default is 0. You can set it as "0" if you want to unlock it immediately. Error 54008 will be returned when placing order during lock interval, it is different from 51034 which is thrown when MMP is triggered

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "result": true
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
result	Boolean	Result of the request <code>true</code> , <code>false</code>

#### POST / CANCEL ALL AFTER

Cancel all pending orders after the countdown timeout. Applicable to all trading symbols through order book (except Spread trading)

**RATE LIMIT: 1 REQUEST PER SECOND**

**RATE LIMIT RULE: USER ID + TAG**

**PERMISSION: TRADE**

**HTTP REQUEST**

`POST /api/v5/trade/cancel-all-after`

#### Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Set cancel all after
result = tradeAPI.cancel_all_after(
  timeOut="10"
)

print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
timeOut	String	Yes	The countdown for order cancellation, with second as the unit. Range of value can be 0, [10, 120]. Setting timeOut to 0 disables Cancel All After.

Parameter	Type	Required	Description
tag	String	No	CAA order tag A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters.

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "triggerTime": "1587971460",
      "tag": "",
      "ts": "1587971400"
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
triggerTime	String	The time the cancellation is triggered. triggerTime=0 means Cancel All After is disabled.
tag	String	CAA order tag
ts	String	The time the request is received.

Users are recommended to send heartbeat to the exchange every second. When the cancel all after is triggered, the trading engine will cancel orders on behalf of the client one by one and this operation may take up to a few seconds. This feature is intended as a protection mechanism for clients only and clients should not use this feature as part of their trading strategies.

To use tag level CAA, first, users need to set tags for their orders using the `tag` request parameter in the placing orders endpoint. When calling the CAA endpoint, if the `tag` request parameter is not provided, the default will be to set CAA at the account level. In this case, all pending orders for all order book trading symbols under that sub-account will be cancelled when CAA triggers, consistent with the existing logic. If the `tag` request parameter is provided, CAA will be set at the order tag level. When triggered, only pending orders of order book trading symbols with the specified tag will be canceled, while orders with other tags or no tags will remain unaffected.

Users can run a maximum of 20 tag level CAAs simultaneously under the same sub-account. The system will only count live tag level CAAs. CAAs that have been triggered or revoked by the user will not be counted. The user will receive error code 51071 when exceeding the limit.

#### GET / ACCOUNT RATE LIMIT

Get account rate limit related information.

Only new order requests and amendment order requests will be counted towards this limit. For batch order requests consisting of multiple orders, each order will be counted individually.

For details, please refer to Fill ratio based sub-account rate limit

#### RATE LIMIT: 1 REQUEST PER SECOND

#### RATE LIMIT RULE: USER ID

#### HTTP REQUEST

```
GET /api/v5/trade/account-rate-limit
```

#### Request Example

#### REQUEST PARAMETERS

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "accRateLimit": "2000",
      "fillRatio": "0.1234",
      "mainFillRatio": "0.1234",
      "nextAccRateLimit": "2000",
      "ts": "123456789000"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
fillRatio	String	Sub account fill ratio during the monitoring period Applicable for users with trading fee level >= VIP 5 and return "" for others For accounts with no trading volume during the monitoring period, return "0". For accounts with trading volume but no order count due to our counting logic, return "9999".
mainFillRatio	String	Master account aggregated fill ratio during the monitoring period Applicable for users with trading fee level >= VIP 5 and return "" for others For accounts with no trading volume during the monitoring period, return "0"
accRateLimit	String	Current sub-account rate limit per two seconds
nextAccRateLimit	String	Expected sub-account rate limit (per two seconds) in the next period Applicable for users with trading fee level >= VIP 5 and return "" for others
ts	String	Data update time For users with trading fee level >= VIP 5, the data will be generated at 08:00 am (UTC) For users with trading fee level < VIP 5, return the current timestamp

## POST / ORDER PRECHECK

This endpoint is used to precheck the account information before and after placing the order.

Only applicable to [Multi-currency margin mode](#), and [Portfolio margin mode](#).

## RATE LIMIT: 5 REQUESTS PER 2 SECONDS

## RATE LIMIT RULE: USER ID

## PERMISSION: TRADE

## HTTP REQUEST

`POST /api/v5/trade/order-precheck`

## Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <a href="#">BTC-USDT</a>
tdMode	String	Yes	Trade mode Margin mode <a href="#">cross</a> <a href="#">isolated</a> Non-Margin mode <a href="#">cash</a> <a href="#">spot_isolated</a> (only applicable to SPOT lead trading, <code>tdMode</code> should be <a href="#">spot_isolated</a> for <a href="#">POT</a> lead trading.)

Parameter	Type	Required	Description
side	String	Yes	Order side, <code>buy</code> <code>sell</code>
posSide	String	Conditional	<p>Position side The default is <code>net</code> in the <code>net</code> mode It is required in the <code>long/short</code> mode, and can only be <code>long</code> or <code>short</code>. Only applicable to <code>FUTURES</code>/<code>SWAP</code>.</p>
ordType	String	Yes	<p>Order type  <code>market</code>: Market order  <code>limit</code>: Limit order  <code>post_only</code>: Post-only order  <code>fok</code>: Fill-or-kill order  <code>ioc</code>: Immediate-or-cancel order  <code>optimal_limit_ioc</code>: Market order with immediate-or-cancel order (applicable only to Expiry Futures and Perpetual Futures).</p>
sz	String	Yes	Quantity to buy or sell
px	String	Conditional	Order price. Only applicable to <code>limit</code> , <code>post_only</code> , <code>fok</code> , <code>ioc</code> , <code>mmp</code> , <code>mmp_and_post_only</code> order.
reduceOnly	Boolean	No	<p>Whether orders can only reduce in position size. Valid options: <code>true</code> or <code>false</code>. The default value is <code>false</code>. Only applicable to <code>MARGIN</code> orders, and <code>FUTURES</code>/<code>SWAP</code> orders in <code>net</code> mode Only applicable to <code>Futures mode</code> and <code>Multi-currency margin</code></p>
tgtCcy	String	No	<p>Whether the target currency uses the quote or base currency.  <code>base_ccy</code>: Base currency, <code>quote_ccy</code>: Quote currency Only applicable to <code>SPOT</code> Market Orders Default is <code>quote_ccy</code> for buy, <code>base_ccy</code> for sell</p>
attachAlgoOrds	Array of objects	No	TP/SL information attached when placing order
> attachAlgoClOrdId	String	No	<p>Client-supplied Algo ID when placing order attaching TP/SL A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. It will be posted to <code>algoClOrdId</code> when placing TP/SL order once the general order is filled completely.</p>
> tpTriggerPx	String	Conditional	<p>Take-profit trigger price For condition TP order, if you fill in this parameter, you should fill in the take-profit order price as well.</p>
> tpOrdPx	String	Conditional	<p>Take-profit order price For condition TP order, if you fill in this parameter, you should fill in the take-profit trigger price as well. For limit TP order, you need to fill in this parameter, take-profit trigger needn't to be filled. If the price is -1, take-profit will be executed at the market price.</p>
> tpOrdKind	String	No	<p>TP order kind  <code>condition</code>  <code>limit</code> The default is <code>condition</code></p>
> slTriggerPx	String	Conditional	<p>Stop-loss trigger price If you fill in this parameter, you should fill in the stop-loss order price.</p>
> slOrdPx	String	Conditional	<p>Stop-loss order price If you fill in this parameter, you should fill in the stop-loss trigger price. If the price is -1, stop-loss will be executed at the market price.</p>

Parameter	Type	Required	Description
> tpTriggerPxType	String	No	<p>Take-profit trigger price type</p> <p><code>last</code>: last price  <code>index</code>: index price  <code>mark</code>: mark price</p> <p>The default is last</p>
> slTriggerPxType	String	No	<p>Stop-loss trigger price type</p> <p><code>last</code>: last price  <code>index</code>: index price  <code>mark</code>: mark price</p> <p>The default is last</p>
> sz	String	Conditional	Size. Only applicable to TP order of split TPs, and it is required for TP order of split TPs

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "adjEq": "41.94347460746277",
      "adjEqChg": "-226.05616481626",
      "availBal": "0",
      "availBalChg": "0",
      "imr": "0",
      "imrChg": "57.74709688430927",
      "liab": "0",
      "liabChg": "0",
      "liabChgCcy": "",
      "liqPx": "6764.8556232031115",
      "liqPxDiff": "-57693.04437679688536773622035980224609375",
      "liqPxDiffRatio": "-0.8950500152315991",
      "mgnRatio": "0",
      "mgnRatioChg": "0",
      "mmr": "0",
      "mmrChg": "0",
      "posBal": "",
      "posBalChg": "",
      "type": ""
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
adjEq	String	Current adjusted / Effective equity in <code>USD</code>
adjEqChg	String	After placing order, changed quantity of adjusted / Effective equity in <code>USD</code>
imr	String	Current initial margin requirement in <code>USD</code>
imrChg	String	After placing order, changed quantity of initial margin requirement in <code>USD</code>
mmr	String	Current Maintenance margin requirement in <code>USD</code>
mmrChg	String	After placing order, changed quantity of maintenance margin requirement in <code>USD</code>
mgnRatio	String	Current Maintenance margin ratio in <code>USD</code>
mgnRatioChg	String	After placing order, changed quantity of Maintenance margin ratio in <code>USD</code>
availBal	String	Current available balance in margin coin currency, only applicable to turn auto borrow off

Parameter	Type	Description
availBalChg	String	After placing order, changed quantity of available balance after placing order, only applicable to turn auto borrow off
liqPx	String	Current estimated liquidation price
liqPxDiff	String	After placing order, the distance between estimated liquidation price and mark price
liqPxDiffRatio	String	After placing order, the distance rate between estimated liquidation price and mark price
posBal	String	Current positive asset, only applicable to margin isolated position
posBalChg	String	After placing order, positive asset of margin isolated, only applicable to margin isolated position
liab	String	Current liabilities of currency For cross, it is cross liabilities For isolated position, it is isolated liabilities
liabChg	String	After placing order, changed quantity of liabilities For cross, it is cross liabilities For isolated position, it is isolated liabilities
liabChgCcy	String	After placing order, the unit of changed liabilities quantity only applicable cross and in auto borrow
type	String	Unit type of positive asset, only applicable to margin isolated position ①: it is both base currency before and after placing order ②: before placing order, it is base currency. after placing order, it is quota currency. ③: before placing order, it is quota currency. after placing order, it is base currency ④: it is both quota currency before and after placing order

## WS / ORDER CHANNEL

Retrieve order information. Data will not be pushed when first subscribed. Data will only be pushed when there are new orders or order updates.

Concurrent connection to this channel will be restricted by the following rules: WebSocket connection count limit.

### URL PATH

/ws/v5/private (required login)

Request Example : single

```

import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/private",
        useServerTime=False
    )
    await ws.start()
    args = [
        {
            "channel": "orders",
            "instType": "FUTURES",
            "instId": "BTC-USD-200329"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)

```

```

await asyncio.sleep(10)

await ws.unsubscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

asyncio.run(main())

```

## Request Example

```

import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/private",
        useServerTime=False
    )
    await ws.start()
    args = [
        {
            "channel": "orders",
            "instType": "FUTURES",
            "instFamily": "BTC-USD"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>orders</code>
> instType	String	Yes	Instrument type <code>SPOT</code> <code>MARGIN</code> <code>SWAP</code> <code>FUTURES</code> <code>OPTION</code> <code>ANY</code>

Parameter	Type	Required	Description
> instFamily	String	No	Instrument family Applicable to <b>FUTURES</b> / <b>SWAP</b> / <b>OPTION</b>
> instId	String	No	Instrument ID

Successful Response Example : single

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "orders",
    "instType": "FUTURES",
    "instId": "BTC-USD-200329"
  },
  "connId": "a4d3ae55"
}
```

Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "orders",
    "instType": "FUTURES",
    "instFamily": "BTC-USD"
  },
  "connId": "a4d3ae55"
}
```

Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \"subscribe\", \\"args\\\": [{ \\"channel\\\" : \"orders\", \\"instType\\\" : \"FUTURES\"}]}",
  "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event <b>subscribe</b> <b>unsubscribe</b> <b>error</b>
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> instType	String	Yes	Instrument type <b>SPOT</b> <b>MARGIN</b> <b>SWAP</b> <b>FUTURES</b> <b>OPTION</b> <b>ANY</b>

Parameter	Type	Required	Description
> instFamily	String	No	Instrument family
> instId	String	No	Instrument ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

### Push Data Example

```
{
  "arg": {
    "channel": "orders",
    "instType": "SPOT",
    "instId": "BTC-USDT",
    "uid": "614488474791936"
  },
  "data": [
    {
      "accFillSz": "0.001",
      "algoClOrdId": "",
      "algoId": "",
      "amendResult": "",
      "amendSource": "",
      "avgPx": "31527.1",
      "cancelSource": "",
      "category": "normal",
      "ccy": "",
      "clOrdId": "",
      "code": "0",
      "cTime": "1654084334977",
      "execType": "M",
      "fee": "-0.02522168",
      "feeCcy": "USDT",
      "fillFee": "-0.02522168",
      "fillFeeCcy": "USDT",
      "fillNotionalUsd": "31.50818374",
      "fillPx": "31527.1",
      "fillSz": "0.001",
      "fillPnl": "0.01",
      "fillTime": "1654084353263",
      "fillPxVol": "",
      "fillPxUsd": "",
      "fillMarkVol": "",
      "fillFwdPx": "",
      "fillMarkPx": "",
      "fillIdxPx": "",
      "instId": "BTC-USDT",
      "instType": "SPOT",
      "lever": "0",
      "msg": "",
      "notionalUsd": "31.50818374",
      "ordId": "452197707845865472",
      "ordType": "limit",
      "pnl": "0",
      "posSide": "",
      "px": "31527.1",
      "pxUsd": "",
      "pxVol": "",
      "pxType": "",
      "quickMgnType": "",
      "rebate": "0",
      "rebateCcy": "BTC",
      "reduceOnly": "false",
      "reqId": "",
      "side": "sell",
      "attachAlgoClOrdId": "",
      "slOrdPx": "",
      "slTriggerPx": ""
    }
  ]
}
```

```

    "slTriggerPxType": "last",
    "source": "",
    "state": "filled",
    "stpId": "",
    "stpMode": "",
    "sz": "0.001",
    "tag": "",
    "tdMode": "cash",
    "tgtCcy": "",
    "tpOrdPx": "",
    "tpTriggerPx": "",
    "tpTriggerPxType": "last",
    "attachAlgoOrds": [],
    "tradeId": "242589207",
    "tradeQuoteCcy": "USDT",
    "lastPx": "38892.2",
    "uTime": "1654084353264",
    "isTpLimit": "false",
    "linkedAlgoOrd": {
        "algoId": ""
    }
}
]
}

```

#### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> uid	String	User Identifier
> instType	String	Instrument type
> instFamily	String	Instrument family
> instId	String	Instrument ID
data	Array of objects	Subscribed data
> instType	String	Instrument type
> instId	String	Instrument ID
> tgtCcy	String	Order quantity unit setting for <code>sz</code> <code>base_ccy</code> : Base currency, <code>quote_ccy</code> : Quote currency Only applicable to <code>SPOT</code> Market orders. Default is <code>quote_ccy</code> for buy, <code>base_ccy</code> for sell
> ccy	String	Margin currency Applicable to all <code>isolated</code> <code>MARGIN</code> orders and <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code> , <code>FUTURES</code> and <code>SWAP</code> contracts.
> ordId	String	Order ID
> clOrdId	String	Client Order ID as assigned by the client
> tag	String	Order tag
> px	String	Price For options, use coin as unit (e.g. BTC, ETH)
> pxUsd	String	Options price in USDOnly applicable to options; return "" for other instrument types
> pxVol	String	Implied volatility of the options orderOnly applicable to options; return "" for other instrument types

Parameter	Type	Description
> pxType	String	<p>Price type of options</p> <p><code>px</code>: Place an order based on price, in the unit of coin (the unit for the request parameter px is BTC or ETH)</p> <p><code>pxVol</code>: Place an order based on pxVol</p> <p><code>pxUsd</code>: Place an order based on pxUsd, in the unit of USD (the unit for the request parameter px is USD)</p>
> sz	String	The original order quantity, <code>SPOT</code> / <code>MARGIN</code> , in the unit of currency; <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code> , in the unit of contract
> notionalUsd	String	Estimated national value in <code>USD</code> of order
> ordType	String	<p>Order type</p> <p><code>market</code>: market order</p> <p><code>limit</code>: limit order</p> <p><code>post_only</code>: Post-only order</p> <p><code>fok</code>: Fill-or-kill order</p> <p><code>ioc</code>: Immediate-or-cancel order</p> <p><code>optimal_limit_ioc</code>: Market order with immediate-or-cancel order (applicable only to Expiry Futures and Perpetual Futures)</p> <p><code>mmp</code>: Market Maker Protection (only applicable to Option in Portfolio Margin mode)</p> <p><code>mmp_and_post_only</code>: Market Maker Protection and Post-only order(only applicable to Option in Portfolio Margin mode).</p> <p><code>op_fok</code>: Simple options (fok)</p>
> side	String	Order side, <code>buy</code> <code>sell</code>
> posSide	String	<p>Position side</p> <p><code>net</code></p> <p><code>long</code> or <code>short</code> Only applicable to <code>FUTURES</code>/<code>SWAP</code></p>
> tdMode	String	Trade mode, <code>cross</code> : cross <code>isolated</code> : isolated <code>cash</code> : cash
> fillPx	String	Filled price for the current update.
> tradeId	String	Trade ID for the current update.
> fillSz	String	<p>Filled quantity for the current update.</p> <p>The unit is <code>base_ccy</code> for SPOT and MARGIN, e.g. BTC-USDT, the unit is BTC; For market orders, the unit both is <code>base_ccy</code> when the tgtCcy is <code>base_ccy</code> or <code>quote_ccy</code>;</p> <p>The unit is contract for <code>FUTURES</code>/<code>SWAP</code>/<code>OPTION</code></p>
> fillPnl	String	Filled profit and loss for the current update, applicable to orders which have a trade and aim to close position. It always is 0 in other conditions
> fillTime	String	Filled time for the current update.
> fillFee	String	<p>Filled fee amount or rebate amount for the current update. :</p> <p>Negative number represents the user transaction fee charged by the platform;</p> <p>Positive number represents rebate</p>
> fillFeeCcy	String	<p>Filled fee currency or rebate currency for the current update..</p> <p>It is fee currency when fillFee is less than 0; It is rebate currency when fillFee&gt;=0.</p>
> fillPxVol	String	<p>Implied volatility when filled</p> <p>Only applicable to options; return "" for other instrument types</p>
> fillPxUsd	String	<p>Options price when filled, in the unit of USD</p> <p>Only applicable to options; return "" for other instrument types</p>
> fillMarkVol	String	<p>Mark volatility when filled</p> <p>Only applicable to options; return "" for other instrument types</p>

Parameter	Type	Description
> fillFwdPx	String	Forward price when filled Only applicable to options; return "" for other instrument types
> fillMarkPx	String	Mark price when filled Applicable to <b>FUTURES</b> , <b>SWAP</b> , <b>OPTION</b>
> fillIdxPx	String	Index price at the moment of trade execution For cross currency spot pairs, it returns baseCcy-USDT index price. For example, for LTC-ETH, this field returns the index price of LTC-USDT.
> execType	String	Liquidity taker or maker for the current update, T: taker M: maker
> accFillSz	String	Accumulated fill quantity The unit is <b>base_ccy</b> for SPOT and MARGIN, e.g. BTC-USDT, the unit is BTC; For market orders, the unit both is <b>base_ccy</b> when the tgtCcy is <b>base_ccy</b> or <b>quote_ccy</b> ; The unit is contract for <b>FUTURES</b> / <b>SWAP</b> / <b>OPTION</b>
> fillNotionalUsd	String	Filled notional value in <b>USD</b> of order
> avgPx	String	Average filled price. If none is filled, it will return <b>0</b> .
> state	String	Order state <b>canceled</b> <b>live</b> <b>partially_filled</b> <b>filled</b> <b>mmp_canceled</b>
> lever	String	Leverage, from <b>0.01</b> to <b>125</b> . Only applicable to <b>MARGIN/FUTURES/SWAP</b>
> attachAlgoClOrdId	String	Client-supplied Algo ID when placing order attaching TP/SL.
> tpTriggerPx	String	Take-profit trigger price, it
> tpTriggerPxType	String	Take-profit trigger price type. <b>last</b> : last price <b>index</b> : index price <b>mark</b> : mark price
> tpOrdPx	String	Take-profit order price, it
> slTriggerPx	String	Stop-loss trigger price, it
> slTriggerPxType	String	Stop-loss trigger price type. <b>last</b> : last price <b>index</b> : index price <b>mark</b> : mark price
> slOrdPx	String	Stop-loss order price, it
> attachAlgoOrds	Array of objects	TP/SL information attached when placing order
>> attachAlgoid	String	The order ID of attached TP/SL order. It can be used to identify the TP/SL order when amending. It will not be posted to algoid when placing TP/SL order after the general order is filled completely.
>> attachAlgoClOrdId	String	Client-supplied Algo ID when placing order attaching TP/SL A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. It will be posted to <b>algoClOrdId</b> when placing TP/SL order once the general order is filled completely.
>> tpOrdKind	String	TP order kind <b>condition</b>

Parameter	Type	Description
		<code>limit</code>
>> tpTriggerPx	String	Take-profit trigger price.
>> tpTriggerRatio	String	Take-profit trigger ratio, 0.3 represents 30%. Only applicable to <code>FUTURES</code> / <code>SWAP</code> contracts.
>> tpTriggerPxType	String	Take-profit trigger price type. <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price
>> tpOrdPx	String	Take-profit order price.
>> slTriggerPx	String	Stop-loss trigger price.
>> slTriggerRatio	String	Stop-loss trigger ratio, 0.3 represents 30%. Only applicable to <code>FUTURES</code> / <code>SWAP</code> contracts.
>> slTriggerPxType	String	Stop-loss trigger price type. <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price
>> slOrdPx	String	Stop-loss order price.
>> sz	String	Size. Only applicable to TP order of split TPs
>> amendPxOnTriggerType	String	Whether to enable Cost-price SL. Only applicable to SL order of split TPs. <code>0</code> : disable, the default value <code>1</code> : Enable
> linkedAlgoOrd	Object	Linked SL order detail, only applicable to TP limit order of one-cancels-the-other order(oco)
>> algoid	Object	Algo ID
> stpld	String	<del>Self trade prevention ID</del> Return "" if self trade prevention is not applicable (deprecated)
> stpMode	String	Self trade prevention mode
> feeCcy	String	Fee currency For maker sell orders of Spot and Margin, this represents the quote currency. For all other cases, it represents the currency in which fees are charged.
> fee	String	Fee amount For Spot and Margin (excluding maker sell orders): accumulated fee charged by the platform, always negative For maker sell orders in Spot and Margin, Expiry Futures, Perpetual Futures and Options: accumulated fee and rebate (always in quote currency for maker sell orders in Spot and Margin)
> rebateCcy	String	Rebate currency For maker sell orders of Spot and Margin, this represents the base currency. For all other cases, it represents the currency in which rebates are paid.
rebate	String	Rebate amount, only applicable to Spot and Margin For maker sell orders: <del>Accumulated fee and</del> rebate amount in the unit of base currency. For all other cases, it represents the maker rebate amount, always positive, return "" if no rebate.
> pnl	String	Profit and loss (excluding the fee). applicable to orders which have a trade and aim to close position. It always is 0 in other conditions. For liquidation under cross margin mode, it will include liquidation penalties.
> source	String	Order source <code>6</code> : The normal order triggered by the <code>trigger order</code> <code>7</code> : The normal order triggered by the <code>TP/SL order</code>

Parameter	Type	Description
		<p>13: The normal order triggered by the algo order</p> <p>25: The normal order triggered by the <code>trailing_stop_order</code></p> <p>34: The normal order triggered by the chase order</p>
> cancelSource	String	<p>Source of the order cancellation.</p> <p>Valid values and the corresponding meanings are:</p> <p>0: Order canceled by system</p> <p>1: Order canceled by user</p> <p>2: Order canceled: Pre reduce-only order canceled, due to insufficient margin in user position</p> <p>3: Order canceled: Risk cancellation was triggered. Pending order was canceled due to insufficient maintenance margin ratio and forced-liquidation risk.</p> <p>4: Order canceled: Borrowings of crypto reached hard cap, order was canceled by system.</p> <p>6: Order canceled: ADL order cancellation was triggered. Pending order was canceled due to a low margin ratio and forced-liquidation risk.</p> <p>7: Order canceled: Futures contract delivery.</p> <p>9: Order canceled: Insufficient balance after funding fees deducted.</p> <p>10: Order canceled: Option contract expiration.</p> <p>13: Order canceled: FOK order was canceled due to incompletely filled.</p> <p>14: Order canceled: IOC order was partially canceled due to incompletely filled.</p> <p>15: Order canceled: The order price is beyond the limit</p> <p>17: Order canceled: Close order was canceled, due to the position was already closed at market price.</p> <p>20: Cancel all after triggered</p> <p>21: Order canceled: The TP/SL order was canceled because the position had been closed</p> <p>22: Order canceled: Due to a better price was available for the order in the same direction, the current operation reduce-only order was automatically canceled</p> <p>23: Order canceled: Due to a better price was available for the order in the same direction, the existing reduce-only order was automatically canceled</p> <p>27: Order canceled: Price limit verification failed because the price difference between counterparties exceeds 5%</p> <p>31: The post-only order will take liquidity in taker orders</p> <p>32: Self trade prevention</p> <p>33: The order exceeds the maximum number of order matches per taker order</p> <p>36: Your TP limit order was canceled because the corresponding SL order was triggered.</p> <p>37: Your TP limit order was canceled because the corresponding SL order was canceled.</p> <p>38: You have canceled market maker protection (MMP) orders.</p> <p>39: Your order was canceled because market maker protection (MMP) was triggered.</p> <p>42: Your order was canceled because the difference between the initial and current best bid or ask prices reached the maximum chase difference.</p> <p>43: Order cancelled because the buy order price is higher than the index price or the sell order price is lower than the index price.</p> <p>44: Your order was canceled because your available balance of this crypto was insufficient for auto conversion. Auto conversion was triggered when the total collateralized liabilities for this crypto reached the platform's risk control limit.</p> <p>46: delta reducing cancel orders</p>
> amendSource	String	<p>Source of the order amendment.</p> <p>1: Order amended by user</p> <p>2: Order amended by user, but the order quantity is overriden by system due to reduce-only</p> <p>3: New order placed by user, but the order quantity is overriden by system due to reduce-only</p> <p>4: Order amended by system due to other pending orders</p> <p>5: Order modification due to changes in options px, pxVol, or pxUsd as a result of following variations. For example, when iv = 60, USD and px are anchored at iv = 60, the changes in USD or px lead to modification.</p>
> category	String	<p>Category</p> <p>normal</p> <p>twap</p> <p>adl</p> <p>full_liquidation</p> <p>partial_liquidation</p> <p>delivery</p> <p>ddh: Delta dynamic hedge</p> <p>auto_conversion</p>

Parameter	Type	Description
> isTpLimit	String	Whether it is TP limit order. true or false
> uTime	String	Update time, Unix timestamp format in milliseconds, e.g. 1597026383085
> cTime	String	Creation time, Unix timestamp format in milliseconds, e.g. 1597026383085
> reqId	String	Client Request ID as assigned by the client for order amendment. "" will be returned if there is no order amendment.
> amendResult	String	<p>The result of amending the order</p> <p>-1: failure 0: success 1: Automatic cancel (amendment request returned success but amendment subsequently failed then automatically canceled by the system) 2: Automatic amendment successfully, only applicable to pxVol and pxUsd orders of Option.</p> <p>When amending the order through API and <code>cx10nFail</code> is set to <code>true</code> in the order amendment request but the amendment is rejected, "" is returned.</p> <p>When amending the order through API, the order amendment acknowledgement returns success and the amendment subsequently failed, -1 will be returned if <code>cx10nFail</code> is set to <code>false</code>, 1 will be returned if <code>cx10nFail</code> is set to <code>true</code>.</p> <p>When amending the order through Web/APP and the amendment failed, 1 will be returned.</p>
> reduceOnly	String	Whether the order can only reduce the position size. Valid options: <code>true</code> or <code>false</code> .
> quickMgnType	String	Quick Margin type, Only applicable to Quick Margin Mode of isolated margin <code>manual</code> , <code>auto_borrow</code> , <code>auto_repay</code>
> algoClOrdId	String	Client-supplied Algo ID. There will be a value when algo order attaching <code>algoClOrdId</code> is triggered, or it will be "".
> algoid	String	Algo ID. There will be a value when algo order is triggered, or it will be "".
> lastPx	String	Last price
> code	String	Error Code, the default is 0
> msg	String	Error Message, The default is ""
> tradeQuoteCcy	String	The quote currency used for trading.

For market orders, it's likely the orders channel will show order state as "filled" while showing the "last filled quantity (fillSz)" as 0.

In exceptional cases, the same message may be sent multiple times (perhaps with the different uTime) . The following guidelines are advised:

1. If a `tradeld` is present, it means a fill. Each `tradeld` should only be returned once per instrument ID, and the later messages that have the same `tradeld` should be discarded.
2. If `tradeld` is absent and the `state` is "filled," it means that the `SPOT`/`MARGIN` market order is fully filled. For messages with the same `ordId`, process only the first filled message and discard any subsequent messages. State = filled is the terminal state of an order.
3. If the state is `canceled` or `mmp\_canceled`, it indicates that the order has been canceled. For cancellation messages with the same `ordId`, process the first one and discard later messages. State = canceled / mmp\_canceled is the terminal state of an order.
4. If `reqId` is present, it indicates a response to a user-requested order modification. It is recommended to use a unique `reqId` for each modification request. For modification messages with the same `reqId`, process only the first message received and discard subsequent messages.

The definitions for fillPx, tradelId, fillSz, fillPnl, fillTime, fillFee, fillFeeCcy, and execType differ between the REST order information endpoints and the orders channel.

## WS / FILLS CHANNEL

Retrieve transaction information. Data will not be pushed when first subscribed. Data will only be pushed when there are order book fill events, where tradelId > 0.

The channel is exclusively available to users with trading fee tier VIP5 or above. For other users, please use WS / Order channel.

### URL PATH

/ws/v5/private (required login)

Request Example: single

```
import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/private",
        useServerTime=False
    )
    await ws.start()
    args = [
        {
            "channel": "fills",
            "instId": "BTC-USDT-SWAP"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

Request Example

```
import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/private",
        useServerTime=False
    )
    await ws.start()
    args = [
        {
```

```

        "channel": "fills"
    }

]

await ws.subscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

await ws.unsubscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>fills</code>
> instId	String	No	Instrument ID

Successful Response Example: single

```
{
    "id": "1512",
    "event": "subscribe",
    "arg": {
        "channel": "fills",
        "instId": "BTC-USDT-SWAP"
    },
    "connId": "a4d3ae55"
}
```

Successful Response Example

```
{
    "id": "1512",
    "event": "subscribe",
    "arg": {
        "channel": "fills"
    },
    "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event <code>subscribe</code> <code>unsubscribe</code> <code>error</code>
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name

Parameter	Type	Required	Description
> instId	String	No	Instrument ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

Push Data Example: single

```
{
  "arg": {
    "channel": "fills",
    "instId": "BTC-USDT-SWAP",
    "uid": "614488474791111"
  },
  "data": [
    {
      "instId": "BTC-USDT-SWAP",
      "fillSz": "100",
      "fillPx": "70000",
      "side": "buy",
      "ts": "1705449605015",
      "ordId": "680800019749904384",
      "clOrdId": "1234567890",
      "tradeId": "12345",
      "execType": "T",
      "count": "10"
    }
  ]
}
```

## PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> uid	String	User Identifier
> instId	String	Instrument ID
data	Array of objects	Subscribed data
> instId	String	Instrument ID
> fillSz	String	Filled quantity. If the trade is aggregated, the filled quantity will also be aggregated.
> fillPx	String	Last filled price
> side	String	Trade direction <code>buy</code> <code>sell</code>
> ts	String	Filled time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> ordId	String	Order ID
> clOrdId	String	Client Order ID as assigned by the client
> tradeId	String	The last trade ID in the trades aggregation
> execType	String	Liquidity taker or maker, <code>T</code> : taker <code>M</code> : maker

Parameter	Type	Description
> count	String	The count of trades aggregated
<ul style="list-style-type: none"> <li>- The channel is exclusively available to users with trading fee tier VIP5 or above. Others will receive error code 60029 when subscribing to it.</li> <li>- The channel only pushes partial information of the orders channel. Fill events of block trading, nitro spread, liquidation, ADL, and some other non order book events will not be pushed through this channel. Users should also subscribe to the orders channel for order confirmation.</li> <li>- When a fill event is received by this channel, the account balance, margin, and position information might not have changed yet.</li> <li>- Taker orders will be aggregated based on different fill prices. When aggregation occurs, the count field indicates the number of orders matched, and the tradeld represents the tradeld of the last trade in the aggregation. Maker orders will not be aggregated.</li> <li>- The channel returns clOrdId. The field will be returned upon trade execution. Note that the fills channel will only return this field if the user-provided clOrdId conforms to the signed int64 positive integer format (1-9223372036854775807, 2^63-1); if the user does not provide this field or if clOrdId does not meet the format requirements, the field will return "0". The order endpoints and channel will continue to return the user-provided clOrdId as usual. All request and response parameters are of string type.</li> <li>- In the future, connection limits will be imposed on this channel. The maximum number of connections subscribing to this channel per subaccount will be 20. We recommend users always use this channel within this limit to avoid any impact on their strategies when the limit is enforced.</li> </ul>		

## WS / PLACE ORDER

You can place an order only if you have sufficient funds.

### URL PATH

/ws/v5/private (required login)

**RATE LIMIT: 60 REQUESTS PER 2 SECONDS**

**RATE LIMIT OF LEAD TRADER LEAD INSTRUMENTS FOR COPY TRADING: 4 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE (EXCEPT OPTIONS): USER ID + INSTRUMENT ID**

**RATE LIMIT RULE (OPTIONS ONLY): USER ID + INSTRUMENT FAMILY**

Rate limit of this endpoint will also be affected by the rules Sub-account rate limit and Fill ratio based sub-account rate limit.

Rate limit is shared with the 'Place order' REST API endpoints

### Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	Yes	<p>Unique identifier of the message</p> <p>Provided by client. It will be returned in response message for identifying the corresponding request.</p> <p>A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.</p>
op	String	Yes	Operation order
args	Array of objects	Yes	Request parameters
> instIdCode	Integer	Conditional	<p>Instrument ID code.</p> <p>If both <code>instId</code> and <code>instIdCode</code> are provided, <code>instIdCode</code> takes precedence.</p>
> instId	String	Conditional	<p>Instrument ID</p> <p>Will be deprecated on February 2026.</p>

Parameter	Type	Required	Description
> tdMode	String	Yes	Trade mode Margin mode <code>isolated</code> <code>cross</code> Non-Margin mode <code>cash</code> <code>spot_isolated</code> (only applicable to SPOT lead trading, <code>tdMode</code> should be <code>spot_isolated</code> for <code>SPOT</code> lead trading.)
> ccy	String	No	Margin currency Applicable to all <code>isolated</code> <code>MARGIN</code> orders and <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code> .
> clOrdId	String	No	Client Order ID as assigned by the client A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
> tag	String	No	Order tag A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters.
> side	String	Yes	Order side, <code>buy</code> <code>sell</code>
> posSide	String	Conditional	Position side The default is <code>net</code> in the <code>net</code> mode It is required in the <code>long/short</code> mode, and can only be <code>long</code> or <code>short</code> . Only applicable to <code>FUTURES</code> / <code>SWAP</code> .
> ordType	String	Yes	Order type <code>market</code> : Market order, only applicable to <code>SPOT/MARGIN/FUTURES/SWAP</code> <code>limit</code> : limit order <code>post_only</code> : Post-only order <code>fok</code> : Fill-or-kill order <code>ioc</code> : Immediate-or-cancel order <code>optimal_limit_ioc</code> : Market order with immediate-or-cancel order <code>mmp</code> : Market Maker Protection (only applicable to Option in Portfolio Margin mode) <code>mmp_and_post_only</code> : Market Maker Protection and Post-only order (only applicable to Option in Portfolio Margin mode)
> sz	String	Yes	Quantity to buy or sell.
> px	String	Conditional	Order price. Only applicable to <code>limit</code> , <code>post_only</code> , <code>fok</code> , <code>ioc</code> , <code>mmp</code> , <code>mmp_and_post_only</code> order. When placing an option order, one of px/pxUsd/pxVol must be filled in, and only one can be filled in
> pxUsd	String	Conditional	Place options orders in <code>USD</code> Only applicable to options When placing an option order, one of px/pxUsd/pxVol must be filled in, and only one can be filled in
> pxVol	String	Conditional	Place options orders based on implied volatility, where 1 represents 100% Only applicable to options When placing an option order, one of px/pxUsd/pxVol must be filled in, and only one can be filled in
> reduceOnly	Boolean	No	Whether the order can only reduce the position size. Valid options: <code>true</code> or <code>false</code> . The default value is <code>false</code> . Only applicable to <code>MARGIN</code> orders, and <code>FUTURES</code> / <code>SWAP</code> orders in <code>net</code> mode Only applicable to <code>Futures mode</code> and <code>Multi-currency margin</code>
> tgtCcy	String	No	Order quantity unit setting for <code>sz</code> <code>base_ccy</code> : Base currency, <code>quote_ccy</code> : Quote currency Only applicable to <code>SPOT</code> Market Orders Default is <code>quote_ccy</code> for buy, <code>base_ccy</code> for sell
> banAmend	Boolean	No	Whether to disallow the system from amending the size of the SPOT Market Order. Valid options: <code>true</code> or <code>false</code> . The default value is <code>false</code> . If <code>true</code> , system will not amend and reject the market order if user does not have sufficient funds. Only applicable to SPOT Market Orders

Parameter	Type	Required	Description
> pxAmendType	String	No	<p>The price amendment type for orders</p> <p>0: Do not allow the system to amend to order price if <code>px</code> exceeds the price limit</p> <p>1: Allow the system to amend the price to the best available value within the price limit if <code>px</code> exceeds the price limit</p> <p>The default value is 0</p>
> tradeQuoteCcy	String	No	<p>The quote currency used for trading. Only applicable to <code>SPOT</code>.</p> <p>The default value is the quote currency of the <code>instId</code>, for example: for <code>BTC-USD</code>, the default is <code>USD</code>.</p>
			<p>Self trade prevention mode.</p> <p><code>cancel_maker</code> <code>cancel_taker</code>, <code>cancel_both</code>.</p> <p>Cancel both does not support FOK</p>
> stpMode	String	No	<p>The account-level <code>acctStpMode</code> will be used to place orders. The default value of this field is <code>cancel_maker</code>. Users can log in to the webpage through the master account to modify this configuration.</p> <p>Users can also utilize the <code>stpMode</code> request parameter of the placing order endpoint to determine the <code>stpMode</code> of a certain order.</p>
expTime	String	No	Request effective deadline. Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

#### Successful Response Example

```
{
  "id": "1512",
  "op": "order",
  "data": [
    {
      "clOrdId": "",
      "ordId": "12345689",
      "tag": "",
      "ts": "1695190491421",
      "sCode": "0",
      "sMsg": ""
    }
  ],
  "code": "0",
  "msg": "",
  "inTime": "1695190491421339",
  "outTime": "1695190491423240"
}
```

#### Failure Response Example

```
{
  "id": "1512",
  "op": "order",
  "data": [
    {
      "clOrdId": "",
      "ordId": "",
      "tag": "",
      "ts": "1695190491421",
      "sCode": "5XXX",
      "sMsg": "not exist"
    }
  ],
  "code": "1",
  "msg": "",
  "inTime": "1695190491421339",
  "outTime": "1695190491423240"
}
```

#### Response Example When Format Error

```
{
  "id": "1512",
```

```

"op": "order",
"data": [],
"code": "60013",
"msg": "Invalid args",
"inTime": "1695190491421339",
"outTime": "1695190491423240"
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
id	String	Unique identifier of the message
op	String	Operation
code	String	Error Code
msg	String	Error message
data	Array of objects	Data
> ordId	String	Order ID
> clOrdId	String	Client Order ID as assigned by the client
> tag	String	Order tag
> ts	String	Timestamp when the order request processing is finished by our system, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> sCode	String	Order status code, <code>0</code> means success
> sMsg	String	Rejection or success message of event execution.
inTime	String	Timestamp at Websocket gateway when the request is received, Unix timestamp format in microseconds, e.g. <code>1597026383085123</code>
outTime	String	Timestamp at Websocket gateway when the response is sent, Unix timestamp format in microseconds, e.g. <code>1597026383085123</code>

### tdMode

Trade Mode, when placing an order, you need to specify the trade mode.

#### Spot mode:

- SPOT and OPTION buyer: cash

#### Futures mode:

- Isolated MARGIN: isolated
- Cross MARGIN: cross
- SPOT: cash
- Cross FUTURES/SWAP/OPTION: cross
- Isolated FUTURES/SWAP/OPTION: isolated

#### Multi-currency margin:

- Isolated MARGIN: isolated
- Cross SPOT: cross
- Cross FUTURES/SWAP/OPTION: cross
- Isolated FUTURES/SWAP/OPTION: isolated

#### Portfolio margin:

- Isolated MARGIN: isolated
- Cross SPOT: cross

- Cross FUTURES/SWAP/OPTION: cross

- Isolated FUTURES/SWAP/OPTION: isolated

#### clOrdId

clOrdId is a user-defined unique ID used to identify the order. It will be included in the response parameters if you have specified during order submission, and can be used as a request parameter to the endpoints to query, cancel and amend orders.

clOrdId must be unique among the clOrdIds of all pending orders.

#### posSide

Position side, this parameter is not mandatory in **net** mode. If you pass it through, the only valid value is **net**.

In **long/short** mode, it is mandatory. Valid values are **long** or **short**.

In **long/short** mode, **side** and **posSide** need to be specified in the combinations below:

Open long: buy and open long (side: fill in buy; posSide: fill in long)

Open short: sell and open short (side: fill in sell; posSide: fill in short)

Close long: sell and close long (side: fill in sell; posSide: fill in long)

Close short: buy and close short (side: fill in buy; posSide: fill in short)

Portfolio margin mode: Expiry Futures and Perpetual Futures only support net mode

#### ordType

Order type. When creating a new order, you must specify the order type. The order type you specify will affect: 1) what order parameters are required, and 2) how the matching system executes your order. The following are valid order types:

limit: Limit order, which requires specified sz and px.

market: Market order. For SPOT and MARGIN, market order will be filled with market price (by swiping opposite order book). For Expiry Futures and Perpetual Futures, market order will be placed to order book with most aggressive price allowed by Price Limit Mechanism. For OPTION, market order is not supported yet. As the filled price for market orders cannot be determined in advance, OKX reserves/freezes your quote currency by an additional 5% for risk check.

post\_only: Post-only order, which the order can only provide liquidity to the market and be a maker. If the order would have executed on placement, it will be canceled instead.

fok: Fill or kill order. If the order cannot be fully filled, the order will be canceled. The order would not be partially filled.

ioc: Immediate or cancel order. Immediately execute the transaction at the order price, cancel the remaining unfilled quantity of the order, and the order quantity will not be displayed in the order book.

optimal\_limit\_ioc: Market order with ioc (immediate or cancel). Immediately execute the transaction of this market order, cancel the remaining unfilled quantity of the order, and the order quantity will not be displayed in the order book. Only applicable to Expiry Futures and Perpetual Futures.

#### sz

Quantity to buy or sell.

For SPOT/MARGIN Buy and Sell Limit Orders, it refers to the quantity in base currency.

For MARGIN Buy Market Orders, it refers to the quantity in quote currency.

For MARGIN Sell Market Orders, it refers to the quantity in base currency.

For SPOT Market Orders, it is set by tgtCcy.

For FUTURES/SWAP/OPTION orders, it refers to the number of contracts.

#### reduceOnly

When placing an order with this parameter set to true, it means that the order will reduce the size of the position only

For the same MARGIN instrument, the coin quantity of all reverse direction pending orders adds `sz` of new `reduceOnly` order cannot exceed the position assets. After the debt is paid off, if there is a remaining size of orders, the position will not be opened in reverse, but will be traded in SPOT.

For the same FUTURES/SWAP instrument, the sum of the current order size and all reverse direction reduce-only pending orders which's price-time priority is higher than the current order, cannot exceed the contract quantity of position.

Only applicable to `Futures mode` and `Multi-currency margin`

Only applicable to `MARGIN` orders, and `FUTURES`/`SWAP` orders in `net` mode

### tgtCcy

This parameter is used to specify the order quantity in the order request is denominated in the quantity of base or quote currency. This is applicable to SPOT Market Orders only.

Base currency: base\_ccy

Quote currency: quote\_ccy

If you use the Base Currency quantity for buy market orders or the Quote Currency for sell market orders, please note:

1. If the quantity you enter is greater than what you can buy or sell, the system will execute the order according to your maximum buyable or sellable quantity. If you want to trade according to the specified quantity, you should use Limit orders.

2. When the market price is too volatile, the locked balance may not be sufficient to buy the Base Currency quantity or sell to receive the Quote Currency that you specified. We will change the quantity of the order to execute the order based on best effort principle based on your account balance. In addition, we will try to over lock a fraction of your balance to avoid changing the order quantity.

#### 2.1 Example of base currency buy market order:

Taking the market order to buy 10 LTCs as an example, and the user can buy 11 LTC. At this time, if  $10 < 11$ , the order is accepted. When the LTC-USDT market price is 200, and the locked balance of the user is 3,000 USDT, as  $200*10 < 3,000$ , the market order of 10 LTC is fully executed; If the market is too volatile and the LTC-USDT market price becomes 400,  $400*10 > 3,000$ , the user's locked balance is not sufficient to buy using the specified amount of base currency, the user's maximum locked balance of 3,000 USDT will be used to settle the trade. Final transaction quantity becomes  $3,000/400 = 7.5$  LTC.

#### 2.2 Example of quote currency sell market order:

Taking the market order to sell 1,000 USDT as an example, and the user can sell 1,200 USDT,  $1,000 < 1,200$ , the order is accepted. When the LTC-USDT market price is 200, and the locked balance of the user is 6 LTC, as  $1,000/200 < 6$ , the market order of 1,000 USDT is fully executed; If the market is too volatile and the LTC-USDT market price becomes 100,  $100*6 < 1,000$ , the user's locked balance is not sufficient to sell using the specified amount of quote currency, the user's maximum locked balance of 6 LTC will be used to settle the trade. Final transaction quantity becomes  $6 * 100 = 600$  USDT.

### px

The value for px must be a multiple of tickSz for OPTION orders.

If not, the system will apply the rounding rules below. Using tickSz 0.0005 as an example:

The px will be rounded up to the nearest 0.0005 when the remainder of px to 0.0005 is more than 0.00025 or 'px' is less than 0.0005.

The px will be rounded down to the nearest 0.0005 when the remainder of px to 0.0005 is less than 0.00025 and 'px' is more than 0.0005.

### Mandatory self trade prevention (STP)

The trading platform imposes mandatory self trade prevention at master account level, which means the accounts under the same master account, including master account itself and all its affiliated sub-accounts, will be prevented from self trade. The account-level acctStpMode will be used to place orders by default. The default value of this field is 'cancel\_maker'. Users can log in to the webpage through the master account to modify this configuration. Users can also utilize the stpMode request parameter of the placing order endpoint to determine the stpMode of a certain order.

Mandatory self trade prevention will not lead to latency.

There are three STP modes. The STP mode is always taken based on the configuration in the taker order.

1. Cancel Maker: This is the default STP mode, which cancels the maker order to prevent self-trading. Then, the taker order continues to match with the next order based on the order book priority.
2. Cancel Taker: The taker order is canceled to prevent self-trading. If the user's own maker order is lower in the order book priority, the taker order is partially filled and then canceled. FOK orders are always honored and canceled if they would result in self-trading.
3. Cancel Both: Both taker and maker orders are canceled to prevent self-trading. If the user's own maker order is lower in the order book priority, the taker order is partially filled. Then, the remaining quantity of the taker order and the first maker order are canceled. FOK orders are not supported in this mode.

### WS / PLACE MULTIPLE ORDERS

Place orders in a batch. Maximum 20 orders can be placed per request

**RATE LIMIT: 300 ORDERS PER 2 SECONDS****RATE LIMIT OF LEAD TRADER LEAD INSTRUMENTS FOR COPY TRADING: 4 ORDERS PER 2 SECONDS****RATE LIMIT RULE (EXCEPT OPTIONS): USER ID + INSTRUMENT ID****RATE LIMIT RULE (OPTIONS ONLY): USER ID + INSTRUMENT FAMILY**

Rate limit of this endpoint will also be affected by the rules Sub-account rate limit and Fill ratio based sub-account rate limit.

Unlike other endpoints, the rate limit of this endpoint is determined by the number of orders. If there is only one order in the request, it will consume the rate limit of 'Place order'.

Rate limit is shared with the 'Place multiple orders' REST API endpoints

**Request Example****REQUEST PARAMETERS**

Parameter	Type	Required	Description
id	String	Yes	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation batch-orders
args	Array of objects	Yes	Request Parameters
> instIdCode	Integer	Conditional	Instrument ID code. If both <code>instId</code> and <code>instIdCode</code> are provided, <code>instIdCode</code> takes precedence.
> instId	String	Conditional	Instrument ID Will be deprecated on February 2026.
> tdMode	String	Yes	Trade mode Margin mode <code>isolated</code> <code>cross</code> Non-Margin mode <code>cash</code> <code>spot_isolated</code> (only applicable to SPOT lead trading, <code>tdMode</code> should be <code>spot_isolated</code> for <code>SPOT</code> lead trading.) Note: <code>isolated</code> is not available in multi-currency margin mode and portfolio margin mode.
> ccy	String	No	Margin currency Applicable to all <code>isolated</code> <code>MARGIN</code> orders and <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code> .
> clOrOrderId	String	No	Client Order ID as assigned by the client A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
> tag	String	No	Order tag A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters.
> side	String	Yes	Order side, <code>buy</code> <code>sell</code>
> posSide	String	Conditional	Position side The default <code>net</code> in the <code>net</code> mode It is required in the <code>long/short</code> mode, and only be <code>long</code> or <code>short</code> . Only applicable to <code>FUTURES</code> / <code>SWAP</code> .
> ordType	String	Yes	Order type <code>market</code> : Market order, only applicable to <code>SPOT/MARGIN/FUTURES/SWAP</code>

Parameter	Type	Required	Description
			<p><code>limit</code>: limit order  <code>post_only</code>: Post-only order  <code>fok</code>: Fill-or-kill order  <code>ioc</code>: Immediate-or-cancel order  <code>optimal_limit_ioc</code>: Market order with immediate-or-cancel order (applicable only to Expiry Futures and Perpetual Futures)  <code>mmp</code>: Market Maker Protection (only applicable to Option in Portfolio Margin mode)  <code>mmp_and_post_only</code>: Market Maker Protection and Post-only order(only applicable to Option in Portfolio Margin mode).</p>
<code>&gt; sz</code>	String	Yes	Quantity to buy or sell.
<code>&gt; px</code>	String	Conditional	Order price. Only applicable to <code>limit</code> , <code>post_only</code> , <code>fok</code> , <code>ioc</code> , <code>mmp</code> , <code>mmp_and_post_only</code> order. When placing an option order, one of px/pxUsd/pxVol must be filled in, and only one can be filled in
<code>&gt; pxUsd</code>	String	Conditional	Place options orders in <code>USD</code> Only applicable to options When placing an option order, one of px/pxUsd/pxVol must be filled in, and only one can be filled in
<code>&gt; pxVol</code>	String	Conditional	Place options orders based on implied volatility, where 1 represents 100% Only applicable to options When placing an option order, one of px/pxUsd/pxVol must be filled in, and only one can be filled in
<code>&gt; reduceOnly</code>	Boolean	No	Whether the order can only reduce the position size. Valid options: <code>true</code> or <code>false</code> . The default value is <code>false</code> . Only applicable to <code>MARGIN</code> orders, and <code>FUTURES</code> / <code>SWAP</code> orders in <code>net</code> mode Only applicable to <code>Futures mode</code> and <code>Multi-currency margin</code>
<code>&gt; tgtCcy</code>	String	No	Order quantity unit setting for <code>sz</code> <code>base_ccy</code> : Base currency , <code>quote_ccy</code> : Quote currency Only applicable to <code>SPOT</code> Market Orders Default is <code>quote_ccy</code> for buy, <code>base_ccy</code> for sell
<code>&gt; banAmend</code>	Boolean	No	Whether to disallow the system from amending the size of the SPOT Market Order. Valid options: <code>true</code> or <code>false</code> . The default value is <code>false</code> . If <code>true</code> , system will not amend and reject the market order if user does not have sufficient funds. Only applicable to SPOT Market Orders
<code>&gt; pxAmendType</code>	String	No	The price amendment type for orders <code>0</code> : Do not allow the system to amend to order price if <code>px</code> exceeds the price limit <code>1</code> : Allow the system to amend the price to the best available value within the price limit if <code>px</code> exceeds the price limit The default value is <code>0</code>
<code>&gt; tradeQuoteCcy</code>	String	No	The quote currency used for trading. Only applicable to <code>SPOT</code> . The default value is the quote currency of the <code>instId</code> , for example: for <code>BTC-USD</code> , the default is <code>USD</code> .
			<p>Self trade prevention mode.  <code>cancel_maker</code>, <code>cancel_taker</code>, <code>cancel_both</code>  Cancel both does not support FOK.</p>
<code>&gt; stpMode</code>	String	No	The account-level <code>acctStpMode</code> will be used to place orders by default. The default value of this field is <code>cancel_maker</code> . Users can log in to the webpage through the master account to modify this configuration. Users can also utilize the <code>stpMode</code> request parameter of the placing order endpoint to determine the <code>stpMode</code> of a certain order.
<code>expTime</code>	String	No	Request effective deadline. Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

Response Example When All Succeed

```
{
  "id": "1513",
  "op": "batch-orders",
  "data": [
    {
      "clOrdId": "",
      "ordId": "12345689",
      "tag": "",
      "ts": "1695190491421",
      "sCode": "0",
      "sMsg": ""
    },
    {
      "clOrdId": "",
      "ordId": "12344",
      "tag": "",
      "ts": "1695190491421",
      "sCode": "0",
      "sMsg": ""
    }
  ],
  "code": "0",
  "msg": "",
  "inTime": "1695190491421339",
  "outTime": "1695190491423240"
}
```

### Response Example When Partially Successful

```
{
  "id": "1513",
  "op": "batch-orders",
  "data": [
    {
      "clOrdId": "",
      "ordId": "12345689",
      "tag": "",
      "ts": "1695190491421",
      "sCode": "0",
      "sMsg": ""
    },
    {
      "clOrdId": "",
      "ordId": "",
      "tag": "",
      "ts": "1695190491421",
      "sCode": "5XXX",
      "sMsg": "Insufficient margin"
    }
  ],
  "code": "2",
  "msg": "",
  "inTime": "1695190491421339",
  "outTime": "1695190491423240"
}
```

### Response Example When All Failed

```
{
  "id": "1513",
  "op": "batch-orders",
  "data": [
    {
      "clOrdId": "oktswap6",
      "ordId": "",
      "tag": "",
      "ts": "1695190491421",
      "sCode": "5XXX",
      "sMsg": "Insufficient margin"
    },
    {
      "clOrdId": "oktswap7",
      "ordId": "",
      "tag": "",
      "ts": "1695190491421",
      "sCode": "5XXX",
      "sMsg": "Insufficient margin"
    }
  ],
  "code": "400",
  "msg": "Bad Request"
}
```

```

        "tag": "",
        "ts": "1695190491421",
        "sCode": "5XXX",
        "sMsg": "Insufficient margin"
    }
],
"code": "1",
"msg": "",
"inTime": "1695190491421339",
"outTime": "1695190491423240"
}

```

### Response Example When Format Error

```

{
    "id": "1513",
    "op": "batch-orders",
    "data": [],
    "code": "60013",
    "msg": "Invalid args",
    "inTime": "1695190491421339",
    "outTime": "1695190491423240"
}

```

### RESPONSE PARAMETERS

Parameter	Type	Description
id	String	Unique identifier of the message
op	String	Operation
code	String	Error Code
msg	String	Error message
data	Array of objects	Data
> ordId	String	Order ID
> clOrdId	String	Client Order ID as assigned by the client
> tag	String	Order tag
> ts	String	Timestamp when the order request processing is finished by our system, Unix timestamp format in milliseconds, e.g. 1597026383085
> sCode	String	Order status code, 0 means success
> sMsg	String	Rejection or success message of event execution.
inTime	String	Timestamp at Websocket gateway when the request is received, Unix timestamp format in microseconds, e.g. 1597026383085123
outTime	String	Timestamp at Websocket gateway when the response is sent, Unix timestamp format in microseconds, e.g. 1597026383085123

In the 'Portfolio Margin' account mode, either all orders are accepted by the system successfully, or all orders are rejected by the system.

#### clOrdId

clOrdId is a user-defined unique ID used to identify the order. It will be included in the response parameters if you have specified during order

submission, and can be used as a request parameter to the endpoints to query, cancel and amend orders. clOrdId must be unique among all pending orders and the current request.

## WS / CANCEL ORDER

Cancel an incomplete order

### URL PATH

/ws/v5/private (required login)

**RATE LIMIT: 60 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE (EXCEPT OPTIONS): USER ID + INSTRUMENT ID**

**RATE LIMIT RULE (OPTIONS ONLY): USER ID + INSTRUMENT FAMILY**

Rate limit is shared with the 'Cancel order' REST API endpoints

### Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	Yes	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation cancel-order
args	Array of objects	Yes	Request Parameters
> instIdCode	Integer	Conditional	Instrument ID code. If both <code>instId</code> and <code>instIdCode</code> are provided, <code>instIdCode</code> takes precedence.
> instId	String	Conditional	Instrument ID Will be deprecated on February 2026.
> ordId	String	Conditional	Order ID Either <code>ordId</code> or <code>clOrdId</code> is required, if both are passed, <code>ordId</code> will be used
> clOrdId	String	Conditional	Client Order ID as assigned by the client A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.

### Successful Response Example

```
{  
  "id": "1514",  
  "op": "cancel-order",  
  "data": [  
    {  
      "clOrdId": "",  
      "ordId": "2510789768709120",  
      "ts": "1695190491421",  
      "sCode": "0",  
      "sMsg": ""  
    }  
,  
    {"code": "0",  
     "msg": "",  
     "inTime": "1695190491421339",  
     "outTime": "1695190491423240"  
  }]
```

```
{
  "id": "1514",
  "op": "cancel-order",
  "data": [
    {
      "clOrdId": "",
      "ordId": "2510789768709120",
      "ts": "1695190491421",
      "sCode": "5XXX",
      "sMsg": "Order not exist"
    }
  ],
  "code": "1",
  "msg": "",
  "inTime": "1695190491421339",
  "outTime": "1695190491423240"
}
```

## Response Example When Format Error

```
{
  "id": "1514",
  "op": "cancel-order",
  "data": [],
  "code": "60013",
  "msg": "Invalid args",
  "inTime": "1695190491421339",
  "outTime": "1695190491423240"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
id	String	Unique identifier of the message
op	String	Operation
code	String	Error Code
msg	String	Error message
data	Array of objects	Data
> ordId	String	Order ID
> clOrdId	String	Client Order ID as assigned by the client
> ts	String	Timestamp when the order request processing is finished by our system, Unix timestamp format in milliseconds, e.g. 1597026383085
> sCode	String	Order status code, 0 means success
> sMsg	String	Order status message
inTime	String	Timestamp at Websocket gateway when the request is received, Unix timestamp format in microseconds, e.g. 1597026383085123
outTime	String	Timestamp at Websocket gateway when the response is sent, Unix timestamp format in microseconds, e.g. 1597026383085123

Cancel order returns with sCode equal to 0. It is not strictly considered that the order has been canceled. It only means that your cancellation request has been accepted by the system server. The result of the cancellation is subject to the state pushed by the order channel or

**WS / CANCEL MULTIPLE ORDERS**

Cancel incomplete orders in batches. Maximum 20 orders can be canceled per request.

**URL PATH**

/ws/v5/private (required login)

**RATE LIMIT: 300 ORDERS PER 2 SECONDS**

**RATE LIMIT RULE (EXCEPT OPTIONS): USER ID + INSTRUMENT ID**

**RATE LIMIT RULE (OPTIONS ONLY): USER ID + INSTRUMENT FAMILY**

Unlike other endpoints, the rate limit of this endpoint is determined by the number of orders. If there is only one order in the request, it will consume the rate limit of 'Cancel order'.

Rate limit is shared with the 'Cancel multiple orders' REST API endpoints

**Request Example****REQUEST PARAMETERS**

Parameter	Type	Required	Description
id	String	Yes	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>batch-cancel-orders</code>
args	Array of objects	Yes	Request Parameters
> instIdCode	Integer	Conditional	Instrument ID code. If both <code>instId</code> and <code>instIdCode</code> are provided, <code>instIdCode</code> takes precedence.
> instId	String	Conditional	Instrument ID Will be deprecated on February 2026.
> ordId	String	Conditional	Order ID Either <code>ordId</code> or <code>c1ordId</code> is required, if both are passed, <code>ordId</code> will be used
> c1ordId	String	Conditional	Client Order ID as assigned by the client A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.

**Response Example When All Succeed**

```
{
  "id": "1515",
  "op": "batch-cancel-orders",
  "data": [
    {
      "c1ordId": "oktswap6",
      "ordId": "2517748157541376",
      "ts": "1695190491421",
      "sCode": "0",
      "sMsg": ""
    },
    {
      "c1ordId": "oktswap7",
      "ordId": "2517748155771904",
      "ts": "1695190491422",
      "sCode": "0",
      "sMsg": ""
    }
  ]
}
```

```

"ts": "1695190491421",
"sCode": "0",
"sMsg": ""
},
],
"code": "0",
"msg": "",
"inTime": "1695190491421339",
"outTime": "1695190491423240"
}
}

```

### Response Example When partially successfully

```

{
  "id": "1515",
  "op": "batch-cancel-orders",
  "data": [
    {
      "clOrdId": "oktswap6",
      "ordId": "2517748157541376",
      "ts": "1695190491421",
      "sCode": "0",
      "sMsg": ""
    },
    {
      "clOrdId": "oktswap7",
      "ordId": "2517748155771904",
      "ts": "1695190491421",
      "sCode": "5XXX",
      "sMsg": "order not exist"
    }
  ],
  "code": "2",
  "msg": "",
  "inTime": "1695190491421339",
  "outTime": "1695190491423240"
}
}

```

### Response Example When All Failed

```

{
  "id": "1515",
  "op": "batch-cancel-orders",
  "data": [
    {
      "clOrdId": "oktswap6",
      "ordId": "2517748157541376",
      "ts": "1695190491421",
      "sCode": "5XXX",
      "sMsg": "order not exist"
    },
    {
      "clOrdId": "oktswap7",
      "ordId": "2517748155771904",
      "ts": "1695190491421",
      "sCode": "5XXX",
      "sMsg": "order not exist"
    }
  ],
  "code": "1",
  "msg": "",
  "inTime": "1695190491421339",
  "outTime": "1695190491423240"
}
}

```

### Response Example When Format Error

```

{
  "id": "1515",
  "op": "batch-cancel-orders",
  "data": [],
  "code": "60013",
  "msg": "batch-cancel-orders format error"
}
}

```

```

"msg": "Invalid args",
"inTime": "1695190491421339",
"outTime": "1695190491423240"
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
id	String	Unique identifier of the message
op	String	Operation
code	String	Error Code
msg	String	Error message
data	Array of objects	Data
> ordId	String	Order ID
> clOrdId	String	Client Order ID as assigned by the client
> ts	String	Timestamp when the order request processing is finished by our system, Unix timestamp format in milliseconds, e.g. 1597026383085
> sCode	String	Order status code, 0 means success
> sMsg	String	Order status message
inTime	String	Timestamp at Websocket gateway when the request is received, Unix timestamp format in microseconds, e.g. 1597026383085123
outTime	String	Timestamp at Websocket gateway when the response is sent, Unix timestamp format in microseconds, e.g. 1597026383085123

## WS / AMEND ORDER

Amend an incomplete order.

### URL PATH

/ws/v5/private (required login)

**RATE LIMIT: 60 REQUESTS PER 2 SECONDS**

**RATE LIMIT OF LEAD TRADER LEAD INSTRUMENTS FOR COPY TRADING: 4 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE (EXCEPT OPTIONS): USER ID + INSTRUMENT ID**

**RATE LIMIT RULE (OPTIONS ONLY): USER ID + INSTRUMENT FAMILY**

Rate limit of this endpoint will also be affected by the rules Sub-account rate limit and Fill ratio based sub-account rate limit.

Rate limit is shared with the `Amend order` REST API endpoints

### Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	Yes	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.

Parameter	Type	Required	Description
op	String	Yes	Operation amend-order
args	Array of objects	Yes	Request Parameters
> instIdCode	Integer	Conditional	Instrument ID code. If both <code>instId</code> and <code>instIdCode</code> are provided, <code>instIdCode</code> takes precedence.
> instId	String	Conditional	Instrument ID Will be deprecated on February 2026.
> cxlOnFail	Boolean	No	Whether the order needs to be automatically canceled when the order amendment fails Valid options: <code>false</code> or <code>true</code> , the default is <code>false</code> .
> ordId	String	Conditional	Order ID Either <code>ordId</code> or <code>clOrdId</code> is required, if both are passed, <code>ordId</code> will be used.
> clOrdId	String	Conditional	Client Order ID as assigned by the client
> reqId	String	No	Client Request ID as assigned by the client for order amendment A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
> newSz	String	Conditional	New quantity after amendment and it has to be larger than 0. Either <code>newSz</code> or <code>newPx</code> is required. When amending a partially-filled order, the <code>newSz</code> should include the amount that has been filled.
> newPx	String	Conditional	New price after amendment. When modifying options orders, users can only fill in one of the following: <code>newPx</code> , <code>newPxUsd</code> , or <code>newPxVol</code> . It must be consistent with parameters when placing orders. For example, if users placed the order using <code>px</code> , they should use <code>newPx</code> when modifying the order.
> newPxUsd	String	Conditional	Modify options orders using USD prices Only applicable to options. When modifying options orders, users can only fill in one of the following: <code>newPx</code> , <code>newPxUsd</code> , or <code>newPxVol</code> .
> newPxVol	String	Conditional	Modify options orders based on implied volatility, where 1 represents 100% Only applicable to options. When modifying options orders, users can only fill in one of the following: <code>newPx</code> , <code>newPxUsd</code> , or <code>newPxVol</code> .
> pxAmendType	String	No	The price amendment type for orders 0: Do not allow the system to amend to order price if <code>newPx</code> exceeds the price limit 1: Allow the system to amend the price to the best available value within the price limit if <code>newPx</code> exceeds the price limit The default value is 0
expTime	String	No	Request effective deadline. Unix timestamp format in milliseconds, e.g. 1597026383085

#### Successful Response Example

```
{
  "id": "1512",
  "op": "amend-order",
  "data": [
    {
      "clOrdId": "",
      "ordId": "2510789768709120",
      "ts": "1695190491421",
      "reqId": "b12344",
      "sCode": "0",
      "sMsg": ""
    }
  ],
  "code": "0",
  "msg": "Order amended successfully"
}
```

```

"msg": "",
"inTime": "1695190491421339",
"outTime": "1695190491423240"
}

```

## Failure Response Example

```

{
  "id": "1512",
  "op": "amend-order",
  "data": [
    {
      "clOrdId": "",
      "ordId": "2510789768709120",
      "ts": "1695190491421",
      "reqId": "b12344",
      "sCode": "5XXXX",
      "sMsg": "order not exist"
    }
  ],
  "code": "1",
  "msg": "",
  "inTime": "1695190491421339",
  "outTime": "1695190491423240"
}

```

## Response Example When Format Error

```

{
  "id": "1512",
  "op": "amend-order",
  "data": [],
  "code": "60013",
  "msg": "Invalid args",
  "inTime": "1695190491421339",
  "outTime": "1695190491423240"
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
id	String	Unique identifier of the message
op	String	Operation
code	String	Error Code
msg	String	Error message
data	Array of objects	Data
> ordId	String	Order ID
> clOrdId	String	Client Order ID as assigned by the client
> ts	String	Timestamp when the order request processing is finished by our system, Unix timestamp format in milliseconds, e.g. 1597026383085
> reqId	String	Client Request ID as assigned by the client for order amendment
> sCode	String	Order status code, 0 means success
> sMsg	String	Order status message

Parameter	Type	Description
inTime	String	Timestamp at Websocket gateway when the request is received, Unix timestamp format in microseconds, e.g. 1597026383085123
outTime	String	Timestamp at Websocket gateway when the response is sent, Unix timestamp format in microseconds, e.g. 1597026383085123

### newSz

If the new quantity of the order is less than or equal to the filled quantity when you are amending a partially-filled order, the order status will be changed to filled.

The amend order returns sCode equal to 0. It is not strictly considered that the order has been amended. It only means that your amend order request has been accepted by the system server. The result of the amend is subject to the status pushed by the order channel or the order status query

## WS / AMEND MULTIPLE ORDERS

Amend incomplete orders in batches. Maximum 20 orders can be amended per request.

### URL PATH

/ws/v5/private (required login)

### RATE LIMIT: 300 ORDERS PER 2 SECONDS

### RATE LIMIT OF LEAD TRADER LEAD INSTRUMENTS FOR COPY TRADING: 4 ORDERS PER 2 SECONDS

### RATE LIMIT RULE (EXCEPT OPTIONS): USER ID + INSTRUMENT ID

### RATE LIMIT RULE (OPTIONS ONLY): USER ID + INSTRUMENT FAMILY

Rate limit of this endpoint will also be affected by the rules Sub-account rate limit and Fill ratio based sub-account rate limit.

Unlike other endpoints, the rate limit of this endpoint is determined by the number of orders. If there is only one order in the request, it will consume the rate limit of 'Amend order'.

Rate limit is shared with the 'Amend multiple orders' REST API endpoints

### Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	Yes	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation batch-amend-orders
args	Array of objects	Yes	Request Parameters
> instIdCode	Integer	Conditional	Instrument ID code. If both <code>instId</code> and <code>instIdCode</code> are provided, <code>instIdCode</code> takes precedence.

Parameter	Type	Required	Description
> instId	String	Conditional	Instrument ID Will be deprecated on February 2026.
> cxlOnFail	Boolean	No	Whether the order needs to be automatically canceled when the order amendment fails Valid options: <code>false</code> or <code>true</code> , the default is <code>false</code> .
> ordId	String	Conditional	Order ID Either <code>ordId</code> or <code>clOrdId</code> is required, if both are passed, <code>ordId</code> will be used.
> clOrdId	String	Conditional	Client Order ID as assigned by the client
> reqId	String	No	Client Request ID as assigned by the client for order amendment A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
> newSz	String	Conditional	New quantity after amendment and it has to be larger than 0. Either <code>newSz</code> or <code>newPx</code> is required. When amending a partially-filled order, the <code>newSz</code> should include the amount that has been filled.
> newPx	String	Conditional	New price after amendment. When modifying options orders, users can only fill in one of the following: <code>newPx</code> , <code>newPxUsd</code> , or <code>newPxVol</code> . It must be consistent with parameters when placing orders. For example, if users placed the order using <code>px</code> , they should use <code>newPx</code> when modifying the order.
> newPxUsd	String	Conditional	Modify options orders using USD prices Only applicable to options. When modifying options orders, users can only fill in one of the following: <code>newPx</code> , <code>newPxUsd</code> , or <code>newPxVol</code> .
> newPxVol	String	Conditional	Modify options orders based on implied volatility, where 1 represents 100% Only applicable to options. When modifying options orders, users can only fill in one of the following: <code>newPx</code> , <code>newPxUsd</code> , or <code>newPxVol</code> .
> pxAmendType	String	No	The price amendment type for orders <code>0</code> : Do not allow the system to amend to order price if <code>newPx</code> exceeds the price limit <code>1</code> : Allow the system to amend the price to the best available value within the price limit if <code>newPx</code> exceeds the price limit The default value is <code>0</code>
expTime	String	No	Request effective deadline. Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

Response Example When All Succeed

```
{
  "id": "1513",
  "op": "batch-amend-orders",
  "data": [
    {
      "clOrdId": "oktswap6",
      "ordId": "12345689",
      "ts": "1695190491421",
      "reqId": "b12344",
      "sCode": "0",
      "sMsg": ""
    },
    {
      "clOrdId": "oktswap7",
      "ordId": "12344",
      "ts": "1695190491421",
      "reqId": "b12344",
      "sCode": "0",
      "sMsg": ""
    }
  ],
  "code": "0",
  "msg": "",
  "inTime": "1695190491421339"
}
```

```
"outTime": "1695190491423240"  
}
```

## Response Example When All Failed

```
{  
  "id": "1513",  
  "op": "batch-amend-orders",  
  "data": [  
    {  
      "clOrdId": "",  
      "ordId": "12345689",  
      "ts": "1695190491421",  
      "reqId": "b12344",  
      "sCode": "5XXXX",  
      "sMsg": "order not exist"  
    },  
    {  
      "clOrdId": "oktswap7",  
      "ordId": "",  
      "ts": "1695190491421",  
      "reqId": "b12344",  
      "sCode": "5XXXX",  
      "sMsg": "order not exist"  
    }  
,  
  "code": "1",  
  "msg": "",  
  "inTime": "1695190491421339",  
  "outTime": "1695190491423240"  
}
```

## Response Example When Partially Successful

```
{  
  "id": "1513",  
  "op": "batch-amend-orders",  
  "data": [  
    {  
      "clOrdId": "",  
      "ordId": "12345689",  
      "ts": "1695190491421",  
      "reqId": "b12344",  
      "sCode": "0",  
      "sMsg": ""  
    },  
    {  
      "clOrdId": "oktswap7",  
      "ordId": "",  
      "ts": "1695190491421",  
      "reqId": "b12344",  
      "sCode": "5XXXX",  
      "sMsg": "order not exist"  
    }  
,  
  "code": "2",  
  "msg": "",  
  "inTime": "1695190491421339",  
  "outTime": "1695190491423240"  
}
```

## Response Example When Format Error

```
{  
  "id": "1513",  
  "op": "batch-amend-orders",  
  "data": [],  
  "code": "60013",  
  "msg": "Invalid args",  
  "inTime": "1695190491421339",  
  "outTime": "1695190491423240"  
}
```

```
"outTime": "1695190491423240"  
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
id	String	Unique identifier of the message
op	String	Operation
code	String	Error Code
msg	String	Error message
data	Array of objects	Data
> ordId	String	Order ID
> clOrdId	String	Client Order ID as assigned by the client
> ts	String	Timestamp when the order request processing is finished by our system, Unix timestamp format in milliseconds, e.g. 1597026383085
> reqId	String	Client Request ID as assigned by the client for order amendment If the user provides reqId in the request, the corresponding reqId will be returned
> sCode	String	Order status code, 0 means success
> sMsg	String	Order status message
inTime	String	Timestamp at Websocket gateway when the request is received, Unix timestamp format in microseconds, e.g. 026383085123
outTime	String	Timestamp at Websocket gateway when the response is sent, Unix timestamp format in microseconds, e.g. 1597026383085123

### newSz

If the new quantity of the order is less than or equal to the filled quantity when you are amending a partially-filled order, the order status will be changed to filled.

## WS / MASS CANCEL ORDER

Cancel all the MMP pending orders of an instrument family.

Only applicable to Option in Portfolio Margin mode, and MMP privilege is required.

### URL PATH

/ws/v5/private (required login)

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

Rate limit is shared with the `Mass Cancel Order` REST API endpoints

### Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	Yes	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation mass-cancel
args	Array of objects	Yes	Request parameters
> instType	String	Yes	Instrument type OPTION
> instFamily	String	Yes	Instrument family
> lockInterval	String	No	Lock interval(ms) The range should be [0, 10 000] The default is 0. You can set it as "0" if you want to unlock it immediately. Error 54008 will be returned when placing order during lock interval, it is different from 51034 which is thrown when MMP is triggered

#### SUCCESSFUL RESPONSE EXAMPLE

```
{
  "id": "1512",
  "op": "mass-cancel",
  "data": [
    {
      "result": true
    }
  ],
  "code": "0",
  "msg": ""
}
```

#### Response Example When Format Error

```
{
  "id": "1512",
  "op": "mass-cancel",
  "data": [],
  "code": "60013",
  "msg": "Invalid args"
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
id	String	Unique identifier of the message
op	String	Operation
code	String	Error Code
msg	String	Error message
data	Array of objects	Data
> result	Boolean	Result of the request true, false

# Algo Trading

## POST / PLACE ALGO ORDER

The algo order includes `trigger` order, `oco` order, `chase` order, `conditional` order, `twap` order and trailing order.

### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

### RATE LIMIT OF LEAD TRADER LEAD INSTRUMENTS FOR COPY TRADING: 1 REQUEST PER 2 SECONDS

#### RATE LIMIT RULE (EXCEPT OPTIONS): USER ID + INSTRUMENT ID

#### RATE LIMIT RULE (OPTIONS ONLY): USER ID + INSTRUMENT FAMILY

#### PERMISSION: TRADE

#### HTTP REQUEST

```
POST /api/v5/trade/order-algo
```

#### Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# One-way stop order
result = tradeAPI.place_algo_order(
    instId="BTC-USDT",
    tdMode="cross",
    side="buy",
    ordType="conditional",
    sz="2",
    tpTriggerPx="15",
    tpOrdPx="18"
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT</code>
tdMode	String	Yes	Trade mode Margin mode <code>cross</code> <code>isolated</code> Non-Margin mode <code>cash</code> <code>spot_isolated</code> (only applicable to SPOT lead trading) Note: <code>isolated</code> is not available in multi-currency margin mode and portfolio margin mode.
ccy	String	No	Margin currency Applicable to all <code>isolated</code> <code>MARGIN</code> orders and <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code> .
side	String	Yes	Order side, <code>buy</code> <code>sell</code>
posSide	String	Conditional	Position side Required in <code>long/short</code> mode and only be <code>long</code> or <code>short</code>
ordType	String	Yes	Order type <code>conditional</code> : One-way stop order <code>oco</code> : One-cancels-the-other order <code>chase</code> : chase order, only applicable to FUTURES and SWAP <code>trigger</code> : Trigger order

Parameter	Type	Required	Description
			<p><code>move_order_stop</code>: Trailing order  <code>twap</code>: TWAP order</p>
sz	String	Conditional	<p>Quantity to buy or sell  Either <code>sz</code> or <code>closeFraction</code> is required.</p>
tag	String	No	<p>Order tag  A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters.</p>
tgtCcy	String	No	<p>Order quantity unit setting for <code>sz</code>  <code>base_ccy</code>: Base currency, <code>quote_ccy</code>: Quote currency  Only applicable to <code>SPOT</code> traded with Market buy <code>conditional</code> order  Default is <code>quote_ccy</code> for buy, <code>base_ccy</code> for sell</p>
algoClOrdId	String	No	<p>Client-supplied Algo ID  A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.</p>
closeFraction	String	Conditional	<p>Fraction of position to be closed when the algo order is triggered.  Currently the system supports fully closing the position only so the only accepted value is <code>1</code>. For the same position, only one TPSL pending order for fully closing the position is supported.  This is only applicable to <code>FUTURES</code> or <code>SWAP</code> instruments.  If <code>posSide</code> is <code>net</code>, <code>reduceOnly</code> must be <code>true</code>.  This is only applicable if <code>ordType</code> is <code>conditional</code> or <code>oco</code>.  This is only applicable if the stop loss and take profit order is executed as market order.  This is not supported in Portfolio Margin mode.  Either <code>sz</code> or <code>closeFraction</code> is required.</p>
tradeQuoteCcy	String	No	<p>The quote currency used for trading. Only applicable to <code>SPOT</code>.  The default value is the quote currency of the <code>instId</code>, for example: for <code>BTC-USD</code>, the default is <code>USD</code>.</p>

## Take Profit / Stop Loss Order

Predefine the price you want the order to trigger a market order to execute immediately or it will place a limit order.

This type of order will not freeze your free margin in advance.

learn more about Take Profit / Stop Loss Order

Parameter	Type	Required	Description
tpTriggerPx	String	No	<p>Take-profit trigger price  If you fill in this parameter, you should fill in the take-profit order price as well.</p>
tpTriggerPxType	String	No	<p>Take-profit trigger price type  <code>last</code>: last price  <code>index</code>: index price  <code>mark</code>: mark price  The default is <code>last</code></p>
tpOrdPx	String	No	<p>Take-profit order price  For condition TP order, if you fill in this parameter, you should fill in the take-profit trigger price as well.  For limit TP order, you need to fill in this parameter, but the take-profit trigger price doesn't need to be filled.  If the price is <code>-1</code>, take-profit will be executed at the market price.</p>
tpOrdKind	String	No	<p>TP order kind  <code>condition</code>  <code>limit</code>  The default is <code>condition</code></p>
slTriggerPx	String	No	<p>Stop-loss trigger price  If you fill in this parameter, you should fill in the stop-loss order price.</p>

Parameter	Type	Required	Description
slTriggerPxType	String	No	<p>Stop-loss trigger price type</p> <p><code>last</code>: last price  <code>index</code>: index price  <code>mark</code>: mark price</p> <p>The default is <code>last</code></p>
slOrdPx	String	No	<p>Stop-loss order price</p> <p>If you fill in this parameter, you should fill in the stop-loss trigger price.</p> <p>If the price is <code>-1</code>, stop-loss will be executed at the market price.</p>
cxlOnClosePos	Boolean	No	<p>Whether the TP/SL order placed by the user is associated with the corresponding position of the instrument. If it is associated, the TP/SL order will be canceled when the position is fully closed; if it is not, the TP/SL order will not be affected when the position is fully closed.</p> <p>Valid values:</p> <p><code>true</code>: Place a TP/SL order associated with the position  <code>false</code>: Place a TP/SL order that is not associated with the position</p> <p>The default value is <code>false</code>. If <code>true</code> is passed in, users must pass <code>reduceOnly = true</code> as well, indicating that when placing a TP/SL order associated with a position, it must be a <code>reduceOnly</code> order.</p> <p>Only applicable to <code>Futures mode</code> and <code>Multi-currency margin</code>.</p>
reduceOnly	Boolean	No	<p>Whether the order can only reduce the position size.</p> <p>Valid options: <code>true</code> or <code>false</code>. The default value is <code>false</code>.</p>

### Take Profit / Stop Loss Order

When placing net TP/SL order (`ordType=conditional`) and both take-profit and stop-loss parameters are sent, only stop-loss logic will be performed and take-profit logic will be ignored.

### Chase order

It will place a Post Only order immediately and amend it continuously

Chase order and corresponding Post Only order can't be amended.

Parameter	Type	Required	Description
chaseType	String	No	<p>Chase type.</p> <p><code>distance</code>: distance from best bid/ask price, the default value.  <code>ratio</code>: ratio.</p>
chaseVal	String	No	<p>Chase value.</p> <p>It represents distance from best bid/ask price when <code>chaseType</code> is <code>distance</code>.</p> <p>For USDT-margined contract, the unit is USDT.</p> <p>For USDC-margined contract, the unit is USDC.</p> <p>For Crypto-margined contract, the unit is USD.</p> <p>It represents ratio when <code>chaseType</code> is <code>ratio</code>. 0.1 represents 10%.</p> <p>The default value is 0.</p>
maxChaseType	String	Conditional	<p>Maximum chase type.</p> <p><code>distance</code>: maximum distance from best bid/ask price  <code>ratio</code>: the ratio.</p> <p><code>maxChaseType</code> and <code>maxChaseVal</code> need to be used together or none of them.</p>
maxChaseVal	String	Conditional	<p>Maximum chase value.</p> <p>It represents maximum distance when <code>maxChaseType</code> is <code>distance</code>.</p> <p>It represents ratio when <code>maxChaseType</code> is <code>ratio</code>. 0.1 represents 10%.</p>
reduceOnly	Boolean	No	<p>Whether the order can only reduce the position size.</p> <p>Valid options: <code>true</code> or <code>false</code>. The default value is <code>false</code>.</p>

## Trigger Order

Use a trigger order to place a market or limit order when a specific price level is crossed.

When a Trigger Order is triggered, if your account balance is lower than the order amount, the system will automatically place the order based on your current balance.

Trigger orders do not freeze assets when placed.

Only applicable to SPOT/FUTURES/SWAP

learn more about Trigger Order

Parameter	Type	Required	Description
triggerPx	String	Yes	Trigger price
orderPx	String	Yes	Order Price If the price is <code>-1</code> , the order will be executed at the market price.
triggerPxType	String	No	Trigger price type <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price The default is <code>last</code>
attachAlgoOrds	Array of objects	No	Attached SL/TP orders info Applicable to <code>Futures mode/Multi-currency margin/Portfolio margin</code>
> attachAlgoClOrdId	String	No	Client-supplied Algo ID when placing order attaching TP/SL. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. It will be posted to algoClOrdId when placing TP/SL order once the general order is filled completely.
> tpTriggerPx	String	No	Take-profit trigger price If you fill in this parameter, you should fill in the take-profit order price as well.
> tpTriggerRatio	String	No	Take profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.
> tpTriggerPxType	String	No	Take-profit trigger price type <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price The default is <code>last</code>
> tpOrdPx	String	No	Take-profit order price If you fill in this parameter, you should fill in the take-profit trigger price as well. If the price is <code>-1</code> , take-profit will be executed at the market price.
> slTriggerPx	String	No	Stop-loss trigger price If you fill in this parameter, you should fill in the stop-loss order price.
> slTriggerRatio	String	No	Stop profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.
> slTriggerPxType	String	No	Stop-loss trigger price type <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price The default is <code>last</code>
> slOrdPx	String	No	Stop-loss order price If you fill in this parameter, you should fill in the stop-loss trigger price. If the price is <code>-1</code> , stop-loss will be executed at the market price.

## Trailing Stop Order

A trailing stop order is a stop order that tracks the market price. Its trigger price changes with the market price. Once the trigger price is reached, a market order is placed.

Actual trigger price for sell orders and short positions = Highest price after order placement – Trail variance (Var.), or Highest price after placement  $\times (1 - \text{Trail variance})$  (Ratio).

Actual trigger price for buy orders and long positions = Lowest price after order placement + Trail variance, or Lowest price after order placement  $\times (1 + \text{Trail variance})$ .

You can use the activation price to set the activation condition for a trailing stop order.

learn more about Trailing Stop Order

Parameter	Type	Required	Description
callbackRatio	String	Conditional	Callback price ratio, e.g. <code>0.01</code> represents <code>1%</code> Either <code>callbackRatio</code> or <code>callbackSpread</code> is allowed to be passed.
callbackSpread	String	Conditional	Callback price variance
activePx	String	No	Active price The system will only start tracking the market and calculating your trigger price after the activation price is reached. If you don't set a price, your order will be activated as soon as it's placed.
reduceOnly	Boolean	No	Whether the order can only reduce the position size. Valid options: <code>true</code> or <code>false</code> . The default value is <code>false</code> . This parameter is only valid in the <code>FUTURES</code> / <code>SWAP</code> net mode, and is ignored in the long/short mode.

## TWAP Order

Time-weighted average price (TWAP) strategy splits your order and places smaller orders at regular time intervals.

It is a strategy that will attempt to execute an order which trades in slices of order quantity at regular intervals of time as specified by users.

learn more about TWAP Order

Parameter	Type	Required	Description
pxVar	String	Conditional	Price variance by percentage, range between <code>[0.0001 ~ 0.01]</code> , e.g. <code>0.01</code> represents <code>1%</code> Take buy orders as an example. When the market price is lower than the limit price, small buy orders will be placed above the best bid price within a certain range. This parameter determines the range by percentage. Either <code>pxVar</code> or <code>pxSpread</code> is allowed to be passed.
pxSpread	String	Conditional	Price variance by constant, should be no less than 0 (no upper limit) Take buy orders as an example. When the market price is lower than the limit price, small buy orders will be placed above the best bid price within a certain range. This parameter determines the range by constant.
szLimit	String	Yes	Average amount Take buy orders as an example. When the market price is lower than the limit price, a certain amount of buy orders will be placed above the best bid price within a certain range. This parameter determines the amount.
pxLimit	String	Yes	Price Limit, should be no less than 0 (no upper limit) Take buy orders as an example. When the market price is lower than the limit price, small buy orders will be placed above the best bid price within a certain range. This parameter represents the limit price.
timeInterval	String	Yes	Time interval in unit of <code>second</code> Take buy orders as an example. When the market price is lower than the limit price, small buy orders will be placed above the best bid price within a certain range based on the time cycle. This parameter represents the time cycle.

Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoClOrdId": "order1234",
      "algoId": "1836487817828872192",
      "clOrdId": "",
      "sCode": "0",
      "sMsg": "",
      "tag": ""
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
algoid	String	Algo ID
clOrdId	String	Client Order ID as assigned by the client (Deprecated)
algoClOrdId	String	Client-supplied Algo ID
sCode	String	The code of the event execution result, 0 means success.
sMsg	String	Rejection message if the request is unsuccessful.
tag	String	Order tag

## POST / CANCEL ALGO ORDER

Cancel unfilled algo orders. A maximum of 10 orders can be canceled per request. Request parameters should be passed in the form of an array.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE (EXCEPT OPTIONS): USER ID + INSTRUMENT ID**

**RATE LIMIT RULE (OPTIONS ONLY): USER ID + INSTRUMENT FAMILY**

**PERMISSION: TRADE**

## HTTP REQUEST

POST /api/v5/trade/cancel-algos

Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Cancel unfilled algo orders (not including Iceberg order, TWAP order, Trailing Stop order)
algo_orders = [
  {"instId": "BTC-USDT", "algoId": "590919993110396111"},
  {"instId": "BTC-USDT", "algoId": "590920138287841222"}
]

result = tradeAPI.cancel_algo_order(algo_orders)
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT</code>
algoid	String	Conditional	Algo ID Either <code>algoid</code> or <code>algoClOrdId</code> is required. If both are passed, <code>algoid</code> will be used.
algoClOrdId	String	Conditional	Client-supplied Algo ID Either <code>algoid</code> or <code>algoClOrdId</code> is required. If both are passed, <code>algoid</code> will be used.

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoClOrdId": "",
      "algoId": "1836489397437468672",
      "clOrdId": "",
      "sCode": "0",
      "sMsg": "",
      "tag": ""
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
algoid	String	Algo ID
sCode	String	The code of the event execution result, <code>0</code> means success.
sMsg	String	Rejection message if the request is unsuccessful.
clOrdId	String	Client Order ID as assigned by the client(Deprecated)
algoClOrdId	String	Client-supplied Algo ID(Deprecated)
tag	String	Order tag(Deprecated)

#### POST / AMEND ALGO ORDER

Amend unfilled algo orders (Support Stop order and Trigger order only, not including Move\_order\_stop order, Iceberg order, TWAP order, Trailing Stop order).

#### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: USER ID + INSTRUMENT ID

#### PERMISSION: TRADE

#### HTTP REQUEST

```
POST /api/v5/trade/amend-algos
```

#### Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID
algoid	String	Conditional	Algo ID Either <code>algoid</code> or <code>algoClOrdId</code> is required. If both are passed, <code>algoid</code> will be used.

Parameter	Type	Required	Description
algoClOrdId	String	Conditional	Client-supplied Algo ID Either <code>algoId</code> or <code>algoClOrdId</code> is required. If both are passed, <code>algoId</code> will be used.
cxlOnFail	Boolean	No	Whether the order needs to be automatically canceled when the order amendment fails Valid options: <code>false</code> or <code>true</code> , the default is <code>false</code> .
reqId	String	Conditional	Client Request ID as assigned by the client for order amendment A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. The response will include the corresponding <code>reqId</code> to help you identify the request if you provide it in the request.
newSz	String	Conditional	New quantity after amendment and it has to be larger than 0.

### Take Profit / Stop Loss Order

Parameter	Type	Required	Description
newTpTriggerPx	String	Conditional	Take-profit trigger price. Either the take-profit trigger price or order price is 0, it means that the take-profit is deleted
newTpOrdPx	String	Conditional	Take-profit order price If the price is -1, take-profit will be executed at the market price.
newSlTriggerPx	String	Conditional	Stop-loss trigger price. Either the stop-loss trigger price or order price is 0, it means that the stop-loss is deleted
newSlOrdPx	String	Conditional	Stop-loss order price If the price is -1, stop-loss will be executed at the market price.
newTpTriggerPxType	String	Conditional	Take-profit trigger price type <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price
newSlTriggerPxType	String	Conditional	Stop-loss trigger price type <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price

### Trigger Order

Parameter	Type	Required	Description
newTriggerPx	String	Yes	New trigger price after amendment
newOrdPx	String	Yes	New order price after amendment If the price is <code>-1</code> , the order will be executed at the market price.
newTriggerPxType	String	No	New trigger price type after amendment <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price The default is <code>last</code>
attachAlgoOrds	Array of objects	No	Attached SL/TP orders info Applicable to <code>Futures mode/Multi-currency margin/Portfolio margin</code>
> newTpTriggerPx	String	No	Take-profit trigger price If you fill in this parameter, you should fill in the take-profit order price as well.
> newTpTriggerRatio	String	No	Take profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.

Parameter	Type	Required	Description
> newTpTriggerPxType	String	No	Take-profit trigger price type last: last price index: index price mark: mark price The default is last
> newTpOrdPx	String	No	Take-profit order price If you fill in this parameter, you should fill in the take-profit trigger price as well. If the price is -1, take-profit will be executed at the market price.
> newSITriggerPx	String	No	Stop-loss trigger price If you fill in this parameter, you should fill in the stop-loss order price.
> newSITriggerRatio	String	No	Stop profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.
> newSITriggerPxType	String	No	Stop-loss trigger price type last: last price index: index price mark: mark price The default is last
> newSIOrdPx	String	No	Stop-loss order price If you fill in this parameter, you should fill in the stop-loss trigger price. If the price is -1, stop-loss will be executed at the market price.

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "algoClOrdId": "algo_01",
      "algoId": "2510789768709120",
      "reqId": "p0103ux",
      "sCode": "0",
      "sMsg": ""
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
algoClOrdId	String	Client-supplied Algo ID
reqId	String	Client Request ID as assigned by the client for order amendment.
sCode	String	The code of the event execution result, 0 means success.
sMsg	String	Rejection message if the request is unsuccessful.

#### GET / ALGO ORDER DETAILS

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

## Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
algoId	String	Conditional	Algo ID Either <code>algoId</code> or <code>algoClOrdId</code> is required. If both are passed, <code>algoId</code> will be used.
algoClOrdId	String	Conditional	Client-supplied Algo ID A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "activePx": "",
      "actualPx": "",
      "actualSide": "",
      "actualSz": "0",
      "algoClOrdId": "",
      "algoId": "1753184812254216192",
      "amendPxOnTriggerType": "0",
      "attachAlgoOrds": [],
      "cTime": "1724751378980",
      "callbackRatio": "",
      "callbackSpread": "",
      "ccy": "",
      "chaseType": "",
      "chaseVal": "",
      "clOrdId": "",
      "closeFraction": "",
      "failCode": "0",
      "instId": "BTC-USDT",
      "instType": "SPOT",
      "isTradeBorrowMode": "",
      "last": "62916.5",
      "lever": "",
      "linkedOrd": {
        "ordId": ""
      },
      "maxChaseType": "",
      "maxChaseVal": "",
      "moveTriggerPx": "",
      "ordId": "",
      "ordIdList": [],
      "ordPx": "",
      "ordType": "conditional",
      "posSide": "net",
      "pxLimit": "",
      "pxSpread": "",
      "pxVar": "",
      "quickMgnType": "",
      "reduceOnly": "false",
      "side": "buy",
      "slOrdPx": "",
      "slTriggerPx": "",
      "slTriggerPxType": "",
      "state": "live",
      "sz": "10",
      "szLimit": "",
      "tag": "",
      "tdMode": "cash",
      "tgtCcy": "quote_ccy",
      "timeInterval": "",
      "tpOrdPx": "-1",
      "tpTriggerPx": "10000",
      "tpTriggerPxType": "last",
      "triggerPx": "",
      "triggerPxType": ""
    }
  ]
}
```

```

        "triggerTime": "",
        "tradeQuoteCcy": "USDT",
        "uTime": "1724751378980"
    },
    "msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type
instId	String	Instrument ID
ccy	String	Margin currency Applicable to all <b>isolated</b> <b>MARGIN</b> orders and <b>cross</b> <b>MARGIN</b> orders in <b>Futures mode</b> , <b>FUTURES</b> and <b>SWAP</b> contracts.
ordId	String	Latest order ID. It will be deprecated soon
ordIdList	Array of strings	Order ID list. There will be multiple order IDs when there is TP/SL splitting order.
algId	String	Algo ID
clOrdId	String	Client Order ID as assigned by the client
sz	String	Quantity to buy or sell
closeFraction	String	Fraction of position to be closed when the algo order is triggered
ordType	String	Order type
side	String	Order side
posSide	String	Position side
tdMode	String	Trade mode
tgtCcy	String	Order quantity unit setting for <b>sz</b> <b>base_ccy</b> : Base currency, <b>quote_ccy</b> : Quote currency Only applicable to <b>SPOT</b> Market Orders Default is <b>quote_ccy</b> for buy, <b>base_ccy</b> for sell
state	String	State <b>live</b> <b>pause</b> <b>partially_effective</b> <b>effective</b> <b>canceled</b> <b>order_failed</b> <b>partially_failed</b>
lever	String	Leverage, from <b>0.01</b> to <b>125</b> . Only applicable to <b>MARGIN/FUTURES/SWAP</b>
tpTriggerPx	String	Take-profit trigger price.
tpTriggerPxType	String	Take-profit trigger price type. <b>last</b> : last price <b>index</b> : index price <b>mark</b> : mark price
tpOrdPx	String	Take-profit order price.

Parameter	Type	Description
slTriggerPx	String	Stop-loss trigger price.
slTriggerPxType	String	Stop-loss trigger price type. [last]: last price [index]: index price [mark]: mark price
slOrdPx	String	Stop-loss order price.
triggerPx	String	trigger price.
triggerPxType	String	trigger price type. [last]: last price [index]: index price [mark]: mark price
ordPx	String	Order price for the trigger order
actualSz	String	Actual order quantity
actualPx	String	Actual order price
tag	String	Order tag
actualSide	String	Actual trigger side, [tp]: take profit [sl]: stop loss Only applicable to oco order and conditional order
triggerTime	String	Trigger time, Unix timestamp format in milliseconds, e.g. 1597026383085
pxVar	String	Price ratio Only applicable to [iceberg] order or [twap] order
pxSpread	String	Price variance Only applicable to [iceberg] order or [twap] order
szLimit	String	Average amount Only applicable to [iceberg] order or [twap] order
pxLimit	String	Price Limit Only applicable to [iceberg] order or [twap] order
timeInterval	String	Time interval Only applicable to [twap] order
callbackRatio	String	Callback price ratio Only applicable to [move_order_stop] order
callbackSpread	String	Callback price variance Only applicable to [move_order_stop] order
activePx	String	Active price Only applicable to [move_order_stop] order
moveTriggerPx	String	Trigger price Only applicable to [move_order_stop] order
reduceOnly	String	Whether the order can only reduce the position size. Valid options: true or false.
quickMgnType	String	Quick Margin type, Only applicable to Quick Margin Mode of isolated margin [manual], [auto_borrow], [auto_repay]
last	String	Last filled price while placing

Parameter	Type	Description
failCode	String	<p>It represents that the reason that algo order fails to trigger. It is "" when the state is <code>effective</code>/<code>cancelled</code>. There will be value when the state is <code>order_failed</code>, e.g. 51008; Only applicable to Stop Order, Trailing Stop Order, Trigger order.</p>
algoClOrdId	String	Client-supplied Algo ID
amendPxOnTriggerType	String	<p>Whether to enable Cost-price SL. Only applicable to SL order of split TPs.</p> <p><code>0</code>: disable, the default value  <code>1</code>: Enable</p>
attachAlgoOrds	Array of objects	<p>Attached SL/TP orders info</p> <p>Applicable to <code>Futures mode/Multi-currency margin/Portfolio margin</code></p>
> attachAlgoClOrdId	String	<p>Client-supplied Algo ID when placing order attaching TP/SL.</p> <p>A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.</p> <p>It will be posted to algoClOrdId when placing TP/SL order once the general order is filled completely.</p>
> tpTriggerPx	String	<p>Take-profit trigger price</p> <p>If you fill in this parameter, you should fill in the take-profit order price as well.</p>
> tpTriggerRatio	String	<p>Take profit trigger ratio, 0.3 represents 30%</p> <p>Only applicable to FUTURES and SWAP.</p>
> tpTriggerPxType	String	<p>Take-profit trigger price type</p> <p><code>last</code>: last price  <code>index</code>: index price  <code>mark</code>: mark price</p>
> tpOrdPx	String	<p>Take-profit order price</p> <p>If you fill in this parameter, you should fill in the take-profit trigger price as well.</p> <p>If the price is <code>-1</code>, take-profit will be executed at the market price.</p>
> slTriggerPx	String	<p>Stop-loss trigger price</p> <p>If you fill in this parameter, you should fill in the stop-loss order price.</p>
> slTriggerRatio	String	<p>Stop profit trigger ratio, 0.3 represents 30%</p> <p>Only applicable to FUTURES and SWAP.</p>
> slTriggerPxType	String	<p>Stop-loss trigger price type</p> <p><code>last</code>: last price  <code>index</code>: index price  <code>mark</code>: mark price</p>
> slOrdPx	String	<p>Stop-loss order price</p> <p>If you fill in this parameter, you should fill in the stop-loss trigger price.</p> <p>If the price is <code>-1</code>, stop-loss will be executed at the market price.</p>
linkedOrd	Object	Linked TP order detail, only applicable to SL order that comes from the one-cancels-the-other (OCO) order that contains the TP limit order.
> ordId	String	Order ID
cTime	String	Creation time Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
uTime	String	Order updated time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
isTradeBorrowMode	String	<p>Whether borrowing currency automatically</p> <p><code>true</code>  <code>false</code></p> <p>Only applicable to <code>trigger order</code>, <code>trailing order</code> and <code>twap order</code></p>
chaseType	String	Chase type. Only applicable to <code>chase</code> order.

Parameter	Type	Description
chaseVal	String	Chase value. Only applicable to <code>chase</code> order.
maxChaseType	String	Maximum chase type. Only applicable to <code>chase</code> order.
maxChaseVal	String	Maximum chase value. Only applicable to <code>chase</code> order.
tradeQuoteCcy	String	The quote currency used for trading.

#### GET / ALGO ORDER LIST

Retrieve a list of untriggered Algo orders under the current account.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

GET /api/v5/trade/orders-algo-pending

#### Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Retrieve a list of untriggered one-way stop orders
result = tradeAPI.order_algos_list(
    ordType="conditional"
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
ordType	String	Yes	<p>Order type</p> <ul style="list-style-type: none"> <li><code>conditional</code>: One-way stop order</li> <li><code>oco</code>: One-cancels-the-other order</li> <li><code>chase</code>: chase order, only applicable to FUTURES and SWAP</li> <li><code>trigger</code>: Trigger order</li> <li><code>move_order_stop</code>: Trailing order</li> <li><code>iceberg</code>: Iceberg order</li> <li><code>twap</code>: TWAP order</li> </ul> <p>For every request, unlike other <code>ordType</code> which only can use one type, <code>conditional</code> and <code>oco</code> both can be used and separated with comma.</p>
algId	String	No	Algo ID
instType	String	No	<p>Instrument type</p> <ul style="list-style-type: none"> <li><code>SPOT</code></li> <li><code>SWAP</code></li> <li><code>FUTURES</code></li> <li><code>MARGIN</code></li> </ul>
instId	String	No	Instrument ID, e.g. <code>BTC-USDT</code>

Parameter	Type	Required	Description
after	String	No	Pagination of data to return records earlier than the requested <code>algoId</code> .
before	String	No	Pagination of data to return records newer than the requested <code>algoId</code> .
limit	String	No	Number of results per request. The maximum is <code>100</code> . The default is <code>100</code>

### Response Example

```
{
  "code": "0",
  "data": [
    {
      "activePx": "",
      "actualPx": "",
      "actualSide": "",
      "actualSz": "0",
      "algoClOrdId": "",
      "algoId": "1753184812254216192",
      "amendPxOnTriggerType": "0",
      "attachAlgoOrds": [],
      "cTime": "1724751378980",
      "callbackRatio": "",
      "callbackSpread": "",
      "ccy": "",
      "chaseType": "",
      "chaseVal": "",
      "clOrdId": "",
      "closeFraction": "",
      "failCode": "0",
      "instId": "BTC-USDT",
      "instType": "SPOT",
      "isTradeBorrowMode": "",
      "last": "62916.5",
      "lever": "",
      "linkedOrd": {
        "ordId": ""
      },
      "maxChaseType": "",
      "maxChaseVal": "",
      "moveTriggerPx": "",
      "ordId": "",
      "ordIdList": [],
      "ordPx": "",
      "ordType": "conditional",
      "posSide": "net",
      "pxLimit": "",
      "pxSpread": "",
      "pxVar": "",
      "quickMgnType": "",
      "reduceOnly": "false",
      "side": "buy",
      "slOrdPx": "",
      "slTriggerPx": "",
      "slTriggerPxType": "",
      "state": "live",
      "sz": "10",
      "szLimit": "",
      "tag": "",
      "tdMode": "cash",
      "tgtCcy": "quote_ccy",
      "timeInterval": "",
      "tpOrdPx": "-1",
      "tpTriggerPx": "10000",
      "tpTriggerPxType": "last",
      "triggerPx": "",
      "triggerPxType": "",
      "triggerTime": "",
      "tradeQuoteCcy": "USDT",
      "uTime": "1724751378980"
    }
  ],
  "code": "0",
  "msg": "Success"
}
```

```

"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type
instId	String	Instrument ID
ccy	String	Margin currency Applicable to all <code>isolated</code> <code>MARGIN</code> orders and <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code> , <code>FUTURES</code> and <code>SWAP</code> contracts.
ordId	String	Latest order ID. It will be deprecated soon
ordIdList	Array of strings	Order ID list. There will be multiple order IDs when there is TP/SL splitting order.
algId	String	Algo ID
clOrdId	String	Client Order ID as assigned by the client
sz	String	Quantity to buy or sell
closeFraction	String	Fraction of position to be closed when the algo order is triggered
ordType	String	Order type
side	String	Order side
posSide	String	Position side
tdMode	String	Trade mode
tgtCcy	String	Order quantity unit setting for <code>sz</code> <code>base_ccy</code> : Base currency, <code>quote_ccy</code> : Quote currency Only applicable to <code>SPOT</code> traded with Market order
state	String	State <code>live</code> <code>pause</code>
lever	String	Leverage, from <code>0.01</code> to <code>125</code> . Only applicable to <code>MARGIN/FUTURES/SWAP</code>
tpTriggerPx	String	Take-profit trigger price
tpTriggerPxType	String	Take-profit trigger price type. <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price
tpOrdPx	String	Take-profit order price
slTriggerPx	String	Stop-loss trigger price
slTriggerPxType	String	Stop-loss trigger price type. <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price
slOrdPx	String	Stop-loss order price

Parameter	Type	Description
triggerPx	String	Trigger price
triggerPxType	String	Trigger price type. last: last price index: index price mark: mark price
ordPx	String	Order price for the trigger order
actualSz	String	Actual order quantity
tag	String	Order tag
actualPx	String	Actual order price
actualSide	String	Actual trigger side tp: take profit s1: stop loss Only applicable to oco order and conditional order
triggerTime	String	Trigger time, Unix timestamp format in milliseconds, e.g. 1597026383085
pxVar	String	Price ratio Only applicable to <code>iceberg</code> order or <code>twap</code> order
pxSpread	String	Price variance Only applicable to <code>iceberg</code> order or <code>twap</code> order
szLimit	String	Average amount Only applicable to <code>iceberg</code> order or <code>twap</code> order
pxLimit	String	Price Limit Only applicable to <code>iceberg</code> order or <code>twap</code> order
timeInterval	String	Time interval Only applicable to <code>twap</code> order
callbackRatio	String	Callback price ratio Only applicable to <code>move_order_stop</code> order
callbackSpread	String	Callback price variance Only applicable to <code>move_order_stop</code> order
activePx	String	Active price Only applicable to <code>move_order_stop</code> order
moveTriggerPx	String	Trigger price Only applicable to <code>move_order_stop</code> order
reduceOnly	String	Whether the order can only reduce the position size. Valid options: true or false.
quickMgnType	String	Quick Margin type, Only applicable to Quick Margin Mode of isolated margin manual, auto_borrow, auto_repay
last	String	Last filled price while placing
failCode	String	It represents that the reason that algo order fails to trigger. There will be value when the state is <code>order_failed</code> , e.g. 51008; For this endpoint, it always is "".
algoClOrdId	String	Client-supplied Algo ID
amendPxOnTriggerType	String	Whether to enable Cost-price SL. Only applicable to SL order of split TPs. 0: disable, the default value

Parameter	Type	Description
		1: Enable
attachAlgoOrds	Array of objects	Attached SL/TP orders info Applicable to <code>Futures mode/Multi-currency margin/Portfolio margin</code>
> attachAlgoClOrdId	String	Client-supplied Algo ID when placing order attaching TP/SL. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. It will be posted to algoClOrdId when placing TP/SL order once the general order is filled completely.
> tpTriggerPx	String	Take-profit trigger price If you fill in this parameter, you should fill in the take-profit order price as well.
> tpTriggerRatio	String	Take profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.
> tpTriggerPxType	String	Take-profit trigger price type <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price
> tpOrdPx	String	Take-profit order price If you fill in this parameter, you should fill in the take-profit trigger price as well. If the price is <code>-1</code> , take-profit will be executed at the market price.
> slTriggerPx	String	Stop-loss trigger price If you fill in this parameter, you should fill in the stop-loss order price.
> slTriggerRatio	String	Stop profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.
> slTriggerPxType	String	Stop-loss trigger price type <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price
> slOrdPx	String	Stop-loss order price If you fill in this parameter, you should fill in the stop-loss trigger price. If the price is <code>-1</code> , stop-loss will be executed at the market price.
linkedOrd	Object	Linked TP order detail, only applicable to SL order that comes from the one-cancels-the-other (OCO) order that contains the TP limit order.
> ordId	String	Order ID
cTime	String	Creation time Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
uTime	String	Order updated time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
isTradeBorrowMode	String	Whether borrowing currency automatically <code>true</code> <code>false</code> Only applicable to <code>trigger order</code> , <code>trailing order</code> and <code>twap order</code>
chaseType	String	Chase type. Only applicable to <code>chase</code> order.
chaseVal	String	Chase value. Only applicable to <code>chase</code> order.
maxChaseType	String	Maximum chase type. Only applicable to <code>chase</code> order.
maxChaseVal	String	Maximum chase value. Only applicable to <code>chase</code> order.
tradeQuoteCcy	String	The quote currency used for trading.

Retrieve a list of all algo orders under the current account in the last 3 months.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/trade/orders-algo-history`

Request Example

```
import okx.Trade as Trade

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

tradeAPI = Trade.TradeAPI(apikey, secretkey, passphrase, False, flag)

# Retrieve a list of all one-way stop algo orders
result = tradeAPI.order_algos_history(
    state="effective",
    ordType="conditional"
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
ordType	String	Yes	<p>Order type</p> <p><code>conditional</code>: One-way stop order</p> <p><code>oco</code>: One-cancels-the-other order</p> <p><code>chase</code>: chase order, only applicable to FUTURES and SWAP</p> <p><code>trigger</code>: Trigger order</p> <p><code>move_order_stop</code>: Trailing order</p> <p><code>iceberg</code>: Iceberg order</p> <p><code>twap</code>: TWAP order</p> <p>For every request, unlike other ordType which only can use one type, <code>conditional</code> and <code>oco</code> both can be used and separated with comma.</p>
state	String	Conditional	<p>State</p> <p><code>effective</code></p> <p><code>canceled</code></p> <p><code>order_failed</code></p> <p>Either <code>state</code> or <code>algoId</code> is required</p>
algoId	String	Conditional	<p>Algo ID</p> <p>Either <code>state</code> or <code>algoId</code> is required.</p>
instType	String	No	<p>Instrument type</p> <p><code>SPOT</code></p> <p><code>SWAP</code></p> <p><code>FUTURES</code></p> <p><code>MARGIN</code></p>
instId	String	No	Instrument ID, e.g. <code>BTC-USDT</code>
after	String	No	Pagination of data to return records earlier than the requested <code>algoId</code>
before	String	No	Pagination of data to return records new than the requested <code>algoId</code>

Parameter	Type	Required	Description
limit	String	No	Number of results per request. The maximum is <code>100</code> . The default is <code>100</code>

Response Example

```
{
  "code": "0",
  "data": [
    {
      "activePx": "",
      "actualPx": "",
      "actualSide": "tp",
      "actualSz": "100",
      "algoClOrdId": "",
      "algoId": "1880721064716505088",
      "amendPxOnTriggerType": "0",
      "attachAlgoOrds": [],
      "cTime": "1728552255493",
      "callbackRatio": "",
      "callbackSpread": "",
      "ccy": "",
      "chaseType": "",
      "chaseVal": "",
      "clOrdId": "",
      "closeFraction": "1",
      "failCode": "1",
      "instId": "BTC-USDT-SWAP",
      "instType": "SWAP",
      "isTradeBorrowMode": "",
      "last": "60777.5",
      "lever": "10",
      "linkedOrd": {
        "ordId": ""
      },
      "maxChaseType": "",
      "maxChaseVal": "",
      "moveTriggerPx": "",
      "ordId": "1884789786215137280",
      "ordIdList": [
        "1884789786215137280"
      ],
      "ordPx": "",
      "ordType": "oco",
      "posSide": "long",
      "pxLimit": "",
      "pxSpread": "",
      "pxVar": "",
      "quickMgnType": "",
      "reduceOnly": "true",
      "side": "sell",
      "slOrdPx": "-1",
      "slTriggerPx": "57000",
      "slTriggerPxType": "mark",
      "state": "effective",
      "sz": "100",
      "szLimit": "",
      "tag": "",
      "tdMode": "isolated",
      "tgtCcy": "",
      "timeInterval": "",
      "tpOrdPx": "-1",
      "tpTriggerPx": "63000",
      "tpTriggerPxType": "last",
      "triggerPx": "",
      "triggerPxType": "",
      "triggerTime": "1728673513447",
      "tradeQuoteCcy": "USDT",
      "uTime": "1728673513447"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type
instId	String	Instrument ID
ccy	String	Margin currency Applicable to all <code>isolated</code> <code>MARGIN</code> orders and <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code> , <code>FUTURES</code> and <code>SWAP</code> contracts.
ordId	String	Latest order ID. It will be deprecated soon
ordIdList	Array of strings	Order ID list. There will be multiple order IDs when there is TP/SL splitting order.
algId	String	Algo ID
clOrdId	String	Client Order ID as assigned by the client
sz	String	Quantity to buy or sell
closeFraction	String	Fraction of position to be closed when the algo order is triggered
ordType	String	Order type
side	String	Order side
posSide	String	Position side
tdMode	String	Trade mode
tgtCcy	String	Order quantity unit setting for <code>sz</code> <code>base_ccy</code> : Base currency, <code>quote_ccy</code> : Quote currency Only applicable to <code>SPOT</code> Market Orders Default is <code>quote_ccy</code> for buy, <code>base_ccy</code> for sell
state	String	State <code>effective</code> <code>canceled</code> <code>order_failed</code> <code>partially_failed</code>
lever	String	Leverage, from <code>0.01</code> to <code>125</code> . Only applicable to <code>MARGIN/FUTURES/SWAP</code>
tpTriggerPx	String	Take-profit trigger price.
tpTriggerPxType	String	Take-profit trigger price type. <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price
tpOrdPx	String	Take-profit order price.
slTriggerPx	String	Stop-loss trigger price.
slTriggerPxType	String	Stop-loss trigger price type. <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price
slOrdPx	String	Stop-loss order price.

Parameter	Type	Description
triggerPx	String	trigger price.
triggerPxType	String	trigger price type. <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price
ordPx	String	Order price for the trigger order
actualSz	String	Actual order quantity
actualPx	String	Actual order price
tag	String	Order tag
actualSide	String	Actual trigger side, <code>tp</code> : take profit <code>s1</code> : stop loss Only applicable to oco order and conditional order
triggerTime	String	Trigger time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
pxVar	String	Price ratio Only applicable to <code>iceberg</code> order or <code>twap</code> order
pxSpread	String	Price variance Only applicable to <code>iceberg</code> order or <code>twap</code> order
szLimit	String	Average amount Only applicable to <code>iceberg</code> order or <code>twap</code> order
pxLimit	String	Price Limit Only applicable to <code>iceberg</code> order or <code>twap</code> order
timeInterval	String	Time interval Only applicable to <code>twap</code> order
callbackRatio	String	Callback price ratio Only applicable to <code>move_order_stop</code> order
callbackSpread	String	Callback price variance Only applicable to <code>move_order_stop</code> order
activePx	String	Active price Only applicable to <code>move_order_stop</code> order
moveTriggerPx	String	Trigger price Only applicable to <code>move_order_stop</code> order
reduceOnly	String	Whether the order can only reduce the position size. Valid options: true or false.
quickMgnType	String	Quick Margin type, Only applicable to Quick Margin Mode of isolated margin <code>manual</code> , <code>auto_borrow</code> , <code>auto_repay</code>
last	String	Last filled price while placing
failCode	String	It represents that the reason that algo order fails to trigger. It is "" when the state is <code>effective</code> / <code>cancelled</code> . There will be value when the state is <code>order_failed</code> , e.g. 51008; Only applicable to Stop Order, Trailing Stop Order, Trigger order.
algoCIOrderId	String	Client Algo Order ID as assigned by the client.

Parameter	Type	Description
amendPxOnTriggerType	String	Whether to enable Cost-price SL. Only applicable to SL order of split TPs. 0: disable, the default value 1: Enable
attachAlgoOrds	Array of objects	Attached SL/TP orders info Applicable to <code>Futures mode/Multi-currency margin/Portfolio margin</code>
> attachAlgoClOrdId	String	Client-supplied Algo ID when placing order attaching TP/SL. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. It will be posted to algoClOrdId when placing TP/SL order once the general order is filled completely.
> tpTriggerPx	String	Take-profit trigger price If you fill in this parameter, you should fill in the take-profit order price as well.
> tpTriggerRatio	String	Take profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.
> tpTriggerPxType	String	Take-profit trigger price type last: last price index: index price mark: mark price
> tpOrdPx	String	Take-profit order price If you fill in this parameter, you should fill in the take-profit trigger price as well. If the price is -1, take-profit will be executed at the market price.
> slTriggerPx	String	Stop-loss trigger price If you fill in this parameter, you should fill in the stop-loss order price.
> slTriggerRatio	String	Stop profit trigger ratio, 0.3 represents 30% Only applicable to FUTURES and SWAP.
> slTriggerPxType	String	Stop-loss trigger price type last: last price index: index price mark: mark price
> slOrdPx	String	Stop-loss order price If you fill in this parameter, you should fill in the stop-loss trigger price. If the price is -1, stop-loss will be executed at the market price.
linkedOrd	Object	Linked TP order detail, only applicable to SL order that comes from the one-cancels-the-other (OCO) order that contains the TP limit order.
> ordId	String	Order ID
cTime	String	Creation time Unix timestamp format in milliseconds, e.g. 1597026383085
uTime	String	Order updated time, Unix timestamp format in milliseconds, e.g. 1597026383085
isTradeBorrowMode	String	Whether borrowing currency automatically true false Only applicable to <code>trigger order</code> , <code>trailing order</code> and <code>twap order</code>
chaseType	String	Chase type. Only applicable to <code>chase</code> order.
chaseVal	String	Chase value. Only applicable to <code>chase</code> order.
maxChaseType	String	Maximum chase type. Only applicable to <code>chase</code> order.
maxChaseVal	String	Maximum chase value. Only applicable to <code>chase</code> order.

Parameter	Type	Description
tradeQuoteCcy	String	The quote currency used for trading.

#### WS / ALGO ORDERS CHANNEL

Retrieve algo orders (includes `trigger` order, `oco` order, `conditional` order). Data will not be pushed when first subscribed. Data will only be pushed when there are order updates.

#### URL PATH

/ws/v5/business (required login)

Request Example : single

```
import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/business",
        useServerTime=False
    )
    await ws.start()
    args = [
        {
            "channel": "orders-algo",
            "instType": "FUTURES",
            "instFamily": "BTC-USD",
            "instId": "BTC-USD-200329"
        }
    ]

    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

Request Example

```
import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/business",
        useServerTime=False
    )
    await ws.start()
    args = [
```

```

{
  "channel": "orders-algo",
  "instType": "FUTURES",
  "instFamily": "BTC-USD"
}
]

await ws.subscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

await ws.unsubscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

asyncio.run(main())

```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>orders-algo</code>
> instType	String	Yes	Instrument type <code>SPOT</code> <code>MARGIN</code> <code>SWAP</code> <code>FUTURES</code> <code>ANY</code>
> instFamily	String	No	Instrument family Applicable to <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>
> instId	String	No	Instrument ID

Successful Response Example : single

```

{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "orders-algo",
    "instType": "FUTURES",
    "instFamily": "BTC-USD",
    "instId": "BTC-USD-200329"
  },
  "connId": "a4d3ae55"
}

```

Successful Response Example

```

{
  "id": "1512",
  "event": "subscribe",
  "arg": {

```

```

"channel": "orders-algo",
"instType": "FUTURES",
"instFamily": "BTC-USD"
},
"connId": "a4d3ae55"
}

```

## Failure Response Example

```

{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \"subscribe\", \\"args\\\": [{ \\"channel\\\" : \\"orders-algo\\\", \\"instType\\\" : \\"FUTURES\\\" }]}",
  "connId": "a4d3ae55"
}

```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event   
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> instType	String	Yes	Instrument type     
> instFamily	String	No	Instrument family Applicable to   
> instId	String	No	Instrument ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

## Push Data Example: single

```

{
  "arg": {
    "channel": "orders-algo",
    "uid": "77982378738415879",
    "instType": "FUTURES",
    "instId": "BTC-USD-200329"
  },
  "data": [
    {
      "actualPx": "0",
      "actualSide": "",
      "actualSz": "0",
      "algoClOrdId": "",
      "algoid": "581878926302093312",
      "attachAlgoOrds": [],
      "amendResult": "",
      "cTime": "1685002746818",
      "clOrdId": "581878926302093312",
      "clOrdType": "L1",
      "clOrdTypeText": "Limit"
    }
  ]
}

```

```

"uTime": "1708679675245",
"ccy": "",
"clOrdId": "",
"closeFraction": "",
"failCode": "",
"instId": "BTC-USDC",
"instType": "SPOT",
"last": "26174.8",
"lever": "0",
"notionalUsd": "11.0",
"ordId": "",
"ordIdList": [],
"ordPx": "",
"ordType": "conditional",
"posSide": "",
"quickMgnType": "",
"reduceOnly": "false",
"reqId": "",
"side": "buy",
"slOrdPx": "",
"slTriggerPx": "",
"slTriggerPxType": "",
"state": "live",
"sz": "11",
"tag": "",
"tdMode": "cross",
"tgtCcy": "quote_ccy",
"tpOrdPx": "-1",
"tpTriggerPx": "1",
"tpTriggerPxType": "last",
"triggerPx": "",
"triggerTime": "",
"tradeQuoteCcy": "USDT",
"amendPxOnTriggerType": "0",
"linkedOrd": {
    "ordId": "98192973880283"
},
"isTradeBorrowMode": ""
},
}]
}

```

#### RESPONSE PARAMETERS WHEN DATA IS PUSHED.

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> uid	String	User Identifier
> instType	String	Instrument type
> instFamily	String	Instrument family
> instId	String	Instrument ID
data	Array of objects	Subscribed data
> instType	String	Instrument type
> instId	String	Instrument ID
> ccy	String	Margin currency Applicable to all <b>isolated</b> <b>MARGIN</b> orders and <b>cross</b> <b>MARGIN</b> orders in <b>Futures mode</b> , <b>FUTURES</b> and <b>SWAP</b> contracts.
> ordId	String	Latest order ID, the order ID associated with the algo order. It will be deprecated soon

Parameter	Type	Description
> ordIdList	Array of strings	Order ID list. There will be multiple order IDs when there is TP/SL splitting order.
> algId	String	Algo ID
> clOrdId	String	Client Order ID as assigned by the client
> sz	String	Quantity to buy or sell. [SPOT]/[MARGIN]: in the unit of currency. [FUTURES]/[SWAP]/[OPTION]: in the unit of contract.
> ordType	String	Order type [conditional]: One-way stop order [oco]: One-cancels-the-other order [trigger]: Trigger order [chase]: Chase order
> side	String	Order side [buy] [sell]
> posSide	String	Position side [net] [long] or [short] Only applicable to [FUTURES]/[SWAP]
> tdMode	String	Trade mode [cross]: cross [isolated]: isolated [cash]: cash
> tgtCcy	String	Order quantity unit setting for [sz] [base_ccy]: Base currency [quote_ccy]: Quote currency Only applicable to [SPOT] Market Orders Default is [quote_ccy] for buy, [base_ccy] for sell
> lever	String	Leverage, from [0.01] to [125]. Only applicable to [MARGIN/FUTURES/SWAP]
> state	String	Order status [live]: to be effective [effective]: effective [canceled]: canceled [order_failed]: order failed [partially_failed]: partially failed [partially_effective]: partially effective
> tpTriggerPx	String	Take-profit trigger price.
> tpTriggerPxType	String	Take-profit trigger price type. [last]: last price [index]: index price [mark]: mark price
> tpOrdPx	String	Take-profit order price.
> slTriggerPx	String	Stop-loss trigger price.
> slTriggerPxType	String	Stop-loss trigger price type. [last]: last price

Parameter	Type	Description
		<p><code>index</code>: index price  <code>mark</code>: mark price</p>
> slOrdPx	String	Stop-loss order price.
> triggerPx	String	Trigger price
> triggerPxType	String	Trigger price type. <code>last</code> : last price <code>index</code> : index price <code>mark</code> : mark price
> ordPx	String	Order price for the trigger order
> last	String	Last filled price while placing
> actualSz	String	Actual order quantity
> actualPx	String	Actual order price
> notionalUsd	String	Estimated national value in <code>USD</code> of order
> tag	String	Order tag
> actualSide	String	Actual trigger side Only applicable to oco order and conditional order
> triggerTime	String	Trigger time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> reduceOnly	String	Whether the order can only reduce the position size. Valid options: <code>true</code> or <code>false</code> .
> failCode	String	It represents that the reason that algo order fails to trigger. It is "" when the state is <code>effective</code> / <code>cancelled</code> . There will be value when the state is <code>order_failed</code> , e.g. 51008; Only applicable to Stop Order, Trailing Stop Order, Trigger order.
> algoClOrdId	String	Client Algo Order ID as assigned by the client.
> reqId	String	Client Request ID as assigned by the client for order amendment. "" will be returned if there is no order amendment.
> amendResult	String	The result of amending the order <code>-1</code> : failure <code>0</code> : success
> amendPxOnTriggerType	String	Whether to enable Cost-price SL. Only applicable to SL order of split TPs. <code>0</code> : disable, the default value <code>1</code> : Enable
> attachAlgoOrds	Array of objects	Attached SL/TP orders info Applicable to <code>Futures mode/Multi-currency margin/Portfolio margin</code>
>> attachAlgoClOrdId	String	Client-supplied Algo ID when placing order attaching TP/SL. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. It will be posted to algoClOrdId when placing TP/SL order once the general order is filled completely.
>> tpTriggerPx	String	Take-profit trigger price If you fill in this parameter, you should fill in the take-profit order price as well.
>> tpTriggerRatio	String	Take-profit trigger ratio, 0.3 represents 30%. Only applicable to <code>FUTURES</code> / <code>SWAP</code> contracts.
>> tpTriggerPxType	String	Take-profit trigger price type <code>last</code> : last price

Parameter	Type	Description
		<b>index</b> : index price <b>mark</b> : mark price
>> tpOrdPx	String	Take-profit order price If you fill in this parameter, you should fill in the take-profit trigger price as well. If the price is <b>-1</b> , take-profit will be executed at the market price.
>> slTriggerPx	String	Stop-loss trigger price If you fill in this parameter, you should fill in the stop-loss order price.
>> slTriggerRatio	String	Stop-loss trigger ratio, 0.3 represents 30%. Only applicable to <b>FUTURES</b> / <b>SWAP</b> contracts.
>> slTriggerPxType	String	Stop-loss trigger price type <b>last</b> : last price <b>index</b> : index price <b>mark</b> : mark price
>> slOrdPx	String	Stop-loss order price If you fill in this parameter, you should fill in the stop-loss trigger price. If the price is <b>-1</b> , stop-loss will be executed at the market price.
> linkedOrd	Object	Linked TP order detail, only applicable to SL order that comes from the one-cancels-the-other (OCO) order that contains the TP limit order.
>> ordId	String	Order ID
> cTime	String	Creation time Unix timestamp format in milliseconds, e.g. <b>1597026383085</b>
> uTime	String	Order updated time, Unix timestamp format in milliseconds, e.g. <b>1597026383085</b>
> isTradeBorrowMode	String	Whether borrowing currency automatically <b>true</b> <b>false</b> Only applicable to <b>trigger order</b> , <b>trailing order</b> and <b>twap order</b>
> chaseType	String	Chase type. Only applicable to <b>chase</b> order.
> chaseVal	String	Chase value. Only applicable to <b>chase</b> order.
> maxChaseType	String	Maximum chase type. Only applicable to <b>chase</b> order.
> maxChaseVal	String	Maximum chase value. Only applicable to <b>chase</b> order.
> tradeQuoteCcy	String	The quote currency used for trading.

#### WS / ADVANCE ALGO ORDERS CHANNEL

Retrieve advance algo orders (including Iceberg order, TWAP order, Trailing order). Data will be pushed when first subscribed. Data will be pushed when triggered by events such as placing/canceling order.

#### URL PATH

/ws/v5/business (required login)

Request Example : single

```

import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

```

```

ws = WsPrivateAsync(
    apiKey = "YOUR_API_KEY",
    passphrase = "YOUR_PASSPHRASE",
    secretKey = "YOUR_SECRET_KEY",
    url = "wss://ws.okx.com:8443/ws/v5/business",
    useServerTime=False
)
await ws.start()
args = [
{
    "channel": "algo-advance",
    "instType": "SPOT",
    "instId": "BTC-USDT"
}
]
await ws.subscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

await ws.unsubscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

asyncio.run(main())

```

## Request Example

```

import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/business",
        useServerTime=False
    )
    await ws.start()
    args = [
    {
        "channel": "algo-advance",
        "instType": "SPOT"
    }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.

Parameter	Type	Required	Description
op	String	Yes	Operation <input type="button" value="subscribe"/> <input type="button" value="unsubscribe"/>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <input type="button" value="algo-advance"/>
> instType	String	Yes	Instrument type <input type="button" value="SPOT"/> <input type="button" value="MARGIN"/> <input type="button" value="SWAP"/> <input type="button" value="FUTURES"/> <input type="button" value="ANY"/>
> instId	String	No	Instrument ID
> algoid	String	No	Algo Order ID

Successful Response Example : single

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "algo-advance",
    "instType": "SPOT",
    "instId": "BTC-USDT"
  },
  "connId": "a4d3ae55"
}
```

Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "algo-advance",
    "instType": "SPOT"
  },
  "connId": "a4d3ae55"
}
```

Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \\"subscribe\\\", \\"args\\\": [{ \\"channel\\\" : \\"algo-advance\\\", \\"instType\\\" : \\"FUTURES\\\" }]}",
  "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event <input type="button" value="subscribe"/>

Parameter	Type	Required	Description
			<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px; border-radius: 10px; text-align: center;">unsubscribe</div> <div style="border: 1px solid black; padding: 2px; border-radius: 10px; text-align: center;">error</div> </div>
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
			Instrument type
> instType	String	Yes	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px; border-radius: 10px; text-align: center;">SPOT</div> <div style="border: 1px solid black; padding: 2px; border-radius: 10px; text-align: center;">MARGIN</div> <div style="border: 1px solid black; padding: 2px; border-radius: 10px; text-align: center;">SWAP</div> <div style="border: 1px solid black; padding: 2px; border-radius: 10px; text-align: center;">FUTURES</div> <div style="border: 1px solid black; padding: 2px; border-radius: 10px; text-align: center;">ANY</div> </div>
> instId	String	No	Instrument ID
> algold	String	No	Algo Order ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

### Push Data Example: single

```
{
  "arg": {
    "channel": "algo-advance",
    "uid": "77982378738415879",
    "instType": "SPOT",
    "instId": "BTC-USDT"
  },
  "data": [
    {
      "actualPx": "",
      "actualSide": "",
      "actualSz": "0",
      "algoId": "355056228680335360",
      "cTime": "1630924001545",
      "ccy": "",
      "clOrdId": "",
      "count": "1",
      "instId": "BTC-USDT",
      "instType": "SPOT",
      "lever": "0",
      "notionalUsd": "",
      "ordPx": "",
      "ordType": "iceberg",
      "pTime": "1630924295204",
      "posSide": "net",
      "pxLimit": "10",
      "pxSpread": "1",
      "pxVar": "",
      "side": "buy",
      "slOrdPx": "",
      "slTriggerPx": "",
      "state": "pause",
      "sz": "0.1",
      "szLimit": "0.1",
      "tdMode": "cash",
      "timeInterval": "",
      "tpOrdPx": "",
      "tpTriggerPx": "",
      "tag": "adadadadad",
      "triggerPx": "",
      "triggerTime": "",
      "tradeQuoteCcy": "USDT",
      "callbackRatio": ""
    }
  ]
}
```

```

        "callbackSpread": "",
        "activePx": "",
        "moveTriggerPx": "",
        "failCode": "",
        "algoClOrdId": "",
        "reduceOnly": "",
        "isTradeBorrowMode": true
    }
]
}

```

#### RESPONSE PARAMETERS WHEN DATA IS PUSHED.

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> uid	String	User Identifier
> instType	String	Instrument type
> instId	String	Instrument ID
> algoid	String	Algo Order ID
data	Array of objects	Subscribed data
> instType	String	Instrument type
> instId	String	Instrument ID
> ccy	String	Margin currency Applicable to all <code>isolated</code> <code>MARGIN</code> orders and <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code> , <code>FUTURES</code> and <code>SWAP</code> contracts.
> ordId	String	Order ID, the order ID associated with the algo order.
> algoid	String	Algo ID
> clOrdId	String	Client Order ID as assigned by the client
> sz	String	Quantity to buy or sell. <code>SPOT</code> / <code>MARGIN</code> : in the unit of currency. <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code> : in the unit of contract.
> ordType	String	Order type <code>iceberg</code> : Iceberg order <code>twap</code> : TWAP order <code>move_order_stop</code> : Trailing order
> side	String	Order side, <code>buy</code> <code>sell</code>
> posSide	String	Position side <code>net</code> <code>long</code> or <code>short</code> Only applicable to <code>FUTURES</code> / <code>SWAP</code>
> tdMode	String	Trade mode, <code>cross</code> : cross <code>isolated</code> : isolated <code>cash</code> : cash
> tgtCcy	String	Order quantity unit setting for <code>sz</code> <code>base_ccy</code> : Base currency, <code>quote_ccy</code> : Quote currency Only applicable to <code>SPOT</code> Market Orders Default is <code>quote_ccy</code> for buy, <code>base_ccy</code> for sell
> lever	String	Leverage, from <code>0.01</code> to <code>125</code> . Only applicable to <code>MARGIN/FUTURES/SWAP</code>

Parameter	Type	Description
> state	String	Order status <code>live</code> : to be effective <code>effective</code> : effective <code>partially_effective</code> : partially effective <code>canceled</code> : canceled <code>order_failed</code> : order failed <code>pause</code> : pause
> tpTriggerPx	String	Take-profit trigger price.
> tpOrdPx	String	Take-profit order price.
> slTriggerPx	String	Stop-loss trigger price.
> slOrdPx	String	Stop-loss order price.
> triggerPx	String	Trigger price
> ordPx	String	Order price
> actualSz	String	Actual order quantity
> actualPx	String	Actual order price
> notionalUsd	String	Estimated national value in <code>USD</code> of order
> tag	String	Order tag
> actualSide	String	Actual trigger side
> triggerTime	String	Trigger time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> cTime	String	Creation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> pxVar	String	Price ratio Only applicable to <code>iceberg</code> order or <code>twap</code> order
> pxSpread	String	Price variance Only applicable to <code>iceberg</code> order or <code>twap</code> order
> szLimit	String	Average amount Only applicable to <code>iceberg</code> order or <code>twap</code> order
> pxLimit	String	Price limit Only applicable to <code>iceberg</code> order or <code>twap</code> order
> timeInterval	String	Time interval Only applicable to <code>twap</code> order
> count	String	Algo Order count Only applicable to <code>iceberg</code> order or <code>twap</code> order
> callbackRatio	String	Callback price ratio Only applicable to <code>move_order_stop</code> order
> callbackSpread	String	Callback price variance Only applicable to <code>move_order_stop</code> order
> activePx	String	Active price Only applicable to <code>move_order_stop</code> order

Parameter	Type	Description
> moveTriggerPx	String	Trigger price Only applicable to <code>move_order_stop</code> order
> failCode	String	It represents that the reason that algo order fails to trigger. It is "" when the state is <code>effective</code> / <code>canceled</code> . There will be value when the state is <code>order_failed</code> , e.g. 51008; Only applicable to Stop Order, Trailing Stop Order, Trigger order.
> algoClOrdId	String	Client Algo Order ID as assigned by the client.
> reduceOnly	String	Whether the order can only reduce the position size. Valid options: <code>true</code> or <code>false</code> .
> pTime	String	Push time of algo order information, millisecond format of Unix timestamp, e.g. <code>1597026383085</code>
> isTradeBorrowMode	Boolean	Whether borrowing currency automatically true false Only applicable to <code>trigger order</code> , <code>trailing order</code> and <code>twap order</code>
> tradeQuoteCcy	String	The quote currency used for trading.

## Grid Trading

Grid trading works by the simple strategy of buy low and sell high. After you set the parameters, the system automatically places orders at incrementally increasing or decreasing prices. Overall, the grid bot seeks to capitalize on normal price volatility by placing buy and sell orders at certain regular intervals above and below a predefined base price.

The API endpoints of `Grid Trading` require authentication.

**POST / PLACE GRID ALGO ORDER**

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID + INSTRUMENT ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

`POST /api/v5/tradingBot/grid/order-algo`

Request Example

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT-SWAP</code>
algoOrdType	String	Yes	Algo order type <code>grid</code> : Spot grid <code>contract_grid</code> : Contract grid
maxPx	String	Yes	Upper price of price range
minPx	String	Yes	Lower price of price range
gridNum	String	Yes	Grid quantity
runType	String	No	Grid type <code>1</code> : Arithmetic, <code>2</code> : Geometric Default is Arithmetic

Parameter	Type	Required	Description
tpTriggerPx	String	No	TP trigger price Applicable to <code>Spot grid</code> / <code>Contract grid</code>
slTriggerPx	String	No	SL trigger price Applicable to <code>Spot grid</code> / <code>Contract grid</code>
algoCIOrld	String	No	Client-supplied Algo ID A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
tag	String	No	Order tag
profitSharingRatio	String	No	Profit sharing ratio, it only supports these values <code>0</code> , <code>0.1</code> , <code>0.2</code> , <code>0.3</code> 0.1 represents 10%
triggerParams	Array of objects	No	Trigger Parameters Applicable to <code>Spot grid</code> / <code>Contract grid</code>
> triggerAction	String	Yes	Trigger action <code>start</code> <code>stop</code>
> triggerStrategy	String	Yes	Trigger strategy <code>instant</code> <code>price</code> <code>rsi</code> Default is <code>instant</code>
> delaySeconds	String	No	Delay seconds after action triggered
> timeframe	String	No	K-line type <code>3m</code> , <code>5m</code> , <code>15m</code> , <code>30m</code> ( <code>m</code> : minute) <code>1H</code> , <code>4H</code> ( <code>H</code> : hour) <code>1D</code> ( <code>D</code> : day) This field is only valid when <code>triggerStrategy</code> is <code>rsi</code>
> thold	String	No	Threshold The value should be an integer between 1 to 100 This field is only valid when <code>triggerStrategy</code> is <code>rsi</code>
> triggerCond	String	No	Trigger condition <code>cross_up</code> <code>cross_down</code> <code>above</code> <code>below</code> <code>cross</code> This field is only valid when <code>triggerStrategy</code> is <code>rsi</code>
> timePeriod	String	No	Time Period <code>14</code> This field is only valid when <code>triggerStrategy</code> is <code>rsi</code>
> triggerPx	String	No	Trigger Price This field is only valid when <code>triggerStrategy</code> is <code>price</code>
> stopType	String	No	Stop type Spot grid ①: Sell base currency ②: Keep base currency Contract grid ①: Market Close All positions ②: Keep positions This field is only valid when <code>triggerAction</code> is <code>stop</code>

Parameter	Type	Required	Description
quoteSz	String	Conditional	Invest amount for quote currency Either <code>quoteSz</code> or <code>baseSz</code> is required
baseSz	String	Conditional	Invest amount for base currency Either <code>quoteSz</code> or <code>baseSz</code> is required
tradeQuoteCcy	String	No	The quote currency for trading. Only applicable to SPOT. The default value is the quote currency of instId, e.g. USD for BTC-USD.

### Contract Grid Order

Parameter	Type	Required	Description
sz	String	Yes	Used margin based on <code>USDT</code>
direction	String	Yes	Contract grid type <code>long</code> , <code>short</code> , <code>neutral</code>
lever	String	Yes	Leverage
basePos	Boolean	No	Whether or not open a position when the strategy activates Default is <code>false</code> Neutral contract grid should omit the parameter
tpRatio	String	No	Take profit ratio, 0.1 represents 10%
slRatio	String	No	Stop loss ratio, 0.1 represents 10%

### Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoClOrdId": "",
      "algoId": "447053782921515008",
      "sCode": "0",
      "sMsg": "",
      "tag": ""
    }
  ],
  "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
algoClOrdId	String	Client-supplied Algo ID
sCode	String	The code of the event execution result, <code>0</code> means success.
sMsg	String	Rejection message if the request is unsuccessful.
tag	String	Order tag

### POST / AMEND GRID ALGO ORDER BASIC PARAM

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

## HTTP REQUEST

```
POST /api/v5/tradingBot/grid/amend-algo-basic-param
```

Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
algold	String	Yes	Algo ID
minPx	String	Yes	Minimum price range
maxPx	String	Yes	Maximum price range
gridNum	int	Yes	Grid quantity

Response Example

```
{
  "code": 55123,
  "msg": "100",
  "data": {
    "algoId": "448965992920907776",
    "requiredTopupAmount": "1.235"
  }
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
requiredTopupAmount	String	Required top up investment amount to edit grid parameters.

## POST / AMEND GRID ALGO ORDER

RATE LIMIT: 20 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: TRADE

### HTTP REQUEST

```
POST /api/v5/tradingBot/grid/amend-order-algo
```

Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
algold	String	Yes	Algo ID
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT-SWAP</code>
slTriggerPx	String	No	New stop-loss trigger price if <code>slTriggerPx</code> is set "" means stop-loss trigger price is canceled. Either <code>slTriggerPx</code> or <code>tpTriggerPx</code> is required.
tpTriggerPx	String	No	New take-profit trigger price if <code>tpTriggerPx</code> is set "" means take-profit trigger price is canceled.
tpRatio	String	No	Take profit ratio, 0.1 represents 10%, only applicable to contract grid if it is set "" means take-profit ratio is canceled.

Parameter	Type	Required	Description
slRatio	String	No	Stop loss ratio, 0.1 represents 10%, only applicable to contract grid` if it is set "" means stop-loss ratio is canceled.
topUpAmt	String	No	Top up amount, only applicable to spot grid
triggerParams	Array of objects	No	Trigger Parameters
> triggerAction	String	Yes	<p>Trigger action</p> <div style="display: flex; justify-content: space-around;"> <span>start</span> <span>stop</span> </div>
> triggerStrategy	String	Yes	<p>Trigger strategy</p> <div style="display: flex; justify-content: space-around;"> <span>instant</span> <span>price</span> <span>rsi</span> </div>
> triggerPx	String	No	<p>Trigger Price</p> <p>This field is only valid when <code>triggerStrategy</code> is <code>price</code></p>
> stopType	String	No	<p>Stop type</p> <p>Spot grid ①: Sell base currency ②: Keep base currency</p> <p>Contract grid ①: Market Close All positions ②: Keep positions</p> <p>This field is only valid when <code>triggerAction</code> is <code>stop</code></p>

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "algoCl0rId": "",
      "algoId": "448965992920907776",
      "sCode": "0",
      "sMsg": "",
      "tag": ""
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
algoid	String	Algo ID
algoCl0rId	String	Client-supplied Algo ID
sCode	String	The code of the event execution result, ① means success.
sMsg	String	Rejection message if the request is unsuccessful.
tag	String	Order tag

#### POST / STOP GRID ALGO ORDER

A maximum of 10 orders can be stopped per request.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

POST /api/v5/tradingBot/grid/stop-order-algo

### Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
algold	String	Yes	Algo ID
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT</code>
algoOrdType	String	Yes	Algo order type <code>grid</code> : Spot grid <code>contract_grid</code> : Contract grid
stopType	String	Yes	Stop type Spot grid <code>1</code> : Sell base currency <code>2</code> : Keep base currency Contract grid <code>1</code> : Market Close All positions <code>2</code> : Keep positions

### Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoClOrdId": "",
      "algoId": "448965992920907776",
      "sCode": "0",
      "sMsg": "",
      "tag": ""
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
algoClOrdId	String	Client-supplied Algo ID
sCode	String	The code of the event execution result, <code>0</code> means success.
sMsg	String	Rejection message if the request is unsuccessful.
tag	String	Order tag

#### POST / CLOSE POSITION FOR CONTRACT GRID

Close position when the contract grid stop type is 'keep position'.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

#### HTTP REQUEST

POST /api/v5/tradingBot/grid/close-position

### Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
algold	String	Yes	Algo ID
mktClose	Boolean	Yes	Market close all the positions or not [true]: Market close all position, [false]: Close part of position
sz	String	Conditional	Close position amount, with unit of [contract] If [mktClose] is [false], the parameter is required.
px	String	Conditional	Close position price If [mktClose] is [false], the parameter is required.

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoClOrdId": "",
      "algoId": "448965992920907776",
      "ordId": "",
      "tag": ""
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
ordId	String	Close position order ID If [mktClose] is [true], the parameter will return "".
algoClOrdId	String	Client-supplied Algo ID
tag	String	Order tag

#### POST / CANCEL CLOSE POSITION ORDER FOR CONTRACT GRID

RATE LIMIT: 20 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: TRADE

HTTP REQUEST

```
POST /api/v5/tradingBot/grid/cancel-close-order
```

#### Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
algold	String	Yes	Algo ID
ordId	String	Yes	Close position order ID

#### Response Example

```
{
  "code": "0",
  "msg": ""
```

```

"data": [
  {
    "algoClOrdId": "",
    "algoId": "448965992920907776",
    "ordId": "570627699870375936",
    "tag": ""
  }
]
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
ordId	String	Close position order ID
algoClOrdId	String	Client-supplied Algo ID
tag	String	Order tag

#### POST / INSTANT TRIGGER GRID ALGO ORDER

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID + INSTRUMENT ID**

**PERMISSION: TRADE**

#### HTTP REQUEST

```
POST /api/v5/tradingBot/grid/order-instant-trigger
```

Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
algold	String	Yes	Algo ID
topUpAmt	String	No	Top up amount, only applicable to spot grid

Response Example

```

{
  "code": "0",
  "data": [
    {
      "algoClOrdId": "",
      "algoId": "561564133246894080"
    }
  ],
  "msg": ""
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
algoClOrdId	String	Client-supplied Algo ID

#### GET / GRID ALGO ORDER LIST

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

## HTTP REQUEST

GET /api/v5/tradingBot/grid/orders-algo-pending

## Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
algoOrdType	String	Yes	Algo order type grid: Spot grid contract_grid: Contract grid
algoid	String	No	Algo ID
instId	String	No	Instrument ID, e.g. <code>BTC-USDT</code>
instType	String	No	Instrument type <code>SPOT</code> <code>MARGIN</code> <code>FUTURES</code> <code>SWAP</code>
after	String	No	Pagination of data to return records earlier than the requested <code>algoid</code> .
before	String	No	Pagination of data to return records newer than the requested <code>algoid</code> .
limit	String	No	Number of results per request. The maximum is 100. The default is 100

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "actualLever": "",
      "algoClOrdId": "",
      "algoId": "56802*****64032",
      "algoOrdType": "grid",
      "arbitrageNum": "0",
      "availEq": "",
      "basePos": false,
      "baseSz": "0",
      "cTime": "1681700496249",
      "cancelType": "0",
      "direction": "",
      "floatProfit": "0",
      "gridNum": "10",
      "gridProfit": "0",
      "instFamily": "",
      "instId": "BTC-USDT",
      "instType": "SPOT",
      "investment": "25",
      "lever": "",
      "liqPx": "",
      "maxPx": "5000",
      "minPx": "400",
      "ordFrozen": "",
      "pnlRatio": "0",
      "quoteSz": "25",
      "rebateTrans": [
        {
          "rebate": "0",
          "rebateCcy": "BTC"
        },
        {
          "rebate": "0",
          "rebateCcy": "USDT"
        }
      ]
    }
  ]
}
```

```

    ],
    "runType": "1",
    "slTriggerPx": "",
    "state": "running",
    "stopType": "",
    "sz": "",
    "tag": "",
    "totalPnl": "0",
    "tpTriggerPx": "",
    "triggerParams": [
      {
        "triggerAction": "start",
        "delaySeconds": "0",
        "triggerStrategy": "instant",
        "triggerType": "auto",
        "triggerTime": ""
      },
      {
        "triggerAction": "stop",
        "delaySeconds": "0",
        "triggerStrategy": "instant",
        "stopType": "1",
        "triggerPx": "1000",
        "triggerType": "manual",
        "triggerTime": ""
      }
    ],
    "uTime": "1682062564350",
    "uly": "BTC-USDT",
    "profitSharingRatio": "",
    "copyType": "0",
    "fee": "",
    "fundingFee": "",
    "tradeQuoteCcy": "USDT"
  }
],
"msg": ""
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
algoClOrdId	String	Client-supplied Algo ID
instType	String	Instrument type
instId	String	Instrument ID
cTime	String	Algo order created time, Unix timestamp format in milliseconds, e.g. 1597026383085
uTime	String	Algo order updated time, Unix timestamp format in milliseconds, e.g. 1597026383085
algoOrdType	String	Algo order type grid: Spot grid contract_grid: Contract grid
state	String	Algo order state starting running stopping pending_signal no_close_position: stopped algo order but have not closed position yet
rebateTrans	Array of objects	Rebate transfer info
> rebate	String	Rebate amount

Parameter	Type	Description
> rebateCcy	String	Rebate currency
triggerParams	Array of objects	Trigger Parameters
> triggerAction	String	<p>Trigger action</p> <div style="display: flex; justify-content: space-around;"> <span>start</span> <span>stop</span> </div>
> triggerStrategy	String	<p>Trigger strategy</p> <div style="display: flex; justify-content: space-around;"> <span>instant</span> <span>price</span> <span>rsi</span> </div>
> delaySeconds	String	Delay seconds after action triggered
> triggerTime	String	Actual action triggered time, unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> triggerType	String	<p>Actual action triggered type</p> <div style="display: flex; justify-content: space-around;"> <span>manual</span> <span>auto</span> </div>
> timeframe	String	<p>K-line type</p> <div style="display: flex; justify-content: space-around;"> <span>3m</span> <span>5m</span> <span>15m</span> <span>30m</span> <span>(m: minute)</span> </div> <div style="display: flex; justify-content: space-around;"> <span>1H</span> <span>4H</span> <span>(H: hour)</span> </div> <div style="display: flex; justify-content: space-around;"> <span>1D</span> <span>(D: day)</span> </div> <p>This field is only valid when <code>triggerStrategy</code> is <code>rsi</code></p>
> thold	String	<p>Threshold</p> <p>The value should be an integer between 1 to 100</p> <p>This field is only valid when <code>triggerStrategy</code> is <code>rsi</code></p>
> triggerCond	String	<p>Trigger condition</p> <div style="display: flex; justify-content: space-around;"> <span>cross_up</span> <span>cross_down</span> </div> <div style="display: flex; justify-content: space-around;"> <span>above</span> <span>below</span> </div> <div style="display: flex; justify-content: space-around;"> <span>cross</span> </div> <p>This field is only valid when <code>triggerStrategy</code> is <code>rsi</code></p>
> timePeriod	String	<p>Time Period</p> <div style="display: flex; justify-content: space-around;"> <span>14</span> </div> <p>This field is only valid when <code>triggerStrategy</code> is <code>rsi</code></p>
> triggerPx	String	<p>Trigger Price</p> <p>This field is only valid when <code>triggerStrategy</code> is <code>price</code></p>
> stopType	String	<p>Stop type</p> <p>Spot grid ①: Sell base currency ②: Keep base currency</p> <p>Contract grid ①: Market Close All positions ②: Keep positions</p> <p>This field is only valid when <code>triggerAction</code> is <code>stop</code></p>
maxPx	String	Upper price of price range
minPx	String	Lower price of price range
gridNum	String	Grid quantity
runType	String	<p>Grid type</p> <p>①: Arithmetic, ②: Geometric</p>
tpTriggerPx	String	Take-profit trigger price

Parameter	Type	Description
slTriggerPx	String	Stop-loss trigger price
arbitrageNum	String	The number of arbitrages executed
totalPnl	String	Total P&L
pnlRatio	String	P&L ratio
investment	String	Accumulated investment amount Spot grid investment amount calculated on quote currency
gridProfit	String	Grid profit
floatProfit	String	Variable P&L
cancelType	String	Algo order stop reason 0: None 1: Manual stop 2: Take profit 3: Stop loss 4: Risk control 5: Delivery 6: Signal
		Actual Stop type Spot 1: Sell base currency 2: Keep base currency
		Contract grid 1: Market Close All positions 2: Keep positions
		Quote currency investment amount Only applicable to <code>Spot grid</code>
		Base currency investment amount Only applicable to <code>Spot grid</code>
		Contract grid type long short neutral Only applicable to <code>contract grid</code>
		Whether or not to open a position when the strategy is activated Only applicable to <code>contract grid</code>
sz	String	Used margin based on <code>USDT</code> Only applicable to <code>contract grid</code>
lever	String	Leverage Only applicable to <code>contract grid</code>
actualLever	String	Actual Leverage Only applicable to <code>contract grid</code>
liqPx	String	Estimated liquidation price Only applicable to <code>contract grid</code>
uly	String	Underlying Only applicable to <code>contract grid</code>
instFamily	String	Instrument family Only applicable to <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code> Only applicable to <code>contract grid</code>
ordFrozen	String	Margin used by pending orders Only applicable to <code>contract grid</code>

Parameter	Type	Description
availEq	String	Available margin Only applicable to <code>contract grid</code>
tag	String	Order tag
profitSharingRatio	String	Profit sharing ratio Value range [0, 0.3] If it is a normal order (neither copy order nor lead order), this field returns ""
copyType	String	Profit sharing order type 0: Normal order 1: Copy order without profit sharing 2: Copy order with profit sharing 3: Lead order
fee	String	Accumulated fee. Only applicable to contract grid, or it will be ""
fundingFee	String	Accumulated funding fee. Only applicable to contract grid, or it will be ""
tradeQuoteCcy	String	The quote currency for trading.

## GET / GRID ALGO ORDER HISTORY

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

## RATE LIMIT RULE: USER ID

## PERMISSION: READ

## HTTP REQUEST

GET /a

Page 77 of 118 Page 77 of 118

## Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
algoOrdType	String	Yes	Algo order type grid: Spot grid contract_grid: Contract grid
algoid	String	No	Algo ID
instId	String	No	Instrument ID, e.g. BTC-USDT
instType	String	No	Instrument type SPOT MARGIN FUTURES SWAP
after	String	No	Pagination of data to return records earlier than the requested algoId.
before	String	No	Pagination of data to return records newer than the requested algoId.
limit	String	No	Number of results per request. The maximum is 100. The default is 100.

## Response Example

```
{  
  "code": "0",  
  "data": [  
    {
```

```
"actualLever": "",  
"algoClOrdId": "",  
"algoId": "565849588675117056",  
"algoOrdType": "grid",  
"arbitrageNum": "0",  
"availEq": "",  
"basePos": false,  
"baseSz": "0",  
"cTime": "1681181054927",  
"cancelType": "1",  
"direction": "",  
"floatProfit": "0",  
"gridNum": "10",  
"gridProfit": "0",  
"instFamily": "",  
"instId": "BTC-USDT",  
"instType": "SPOT",  
"investment": "25",  
"lever": "0",  
"liqPx": "",  
"maxPx": "5000",  
"minPx": "400",  
"ordFrozen": "",  
"pnlRatio": "0",  
"quoteSz": "25",  
"rebateTrans": [  
    {  
        "rebate": "0",  
        "rebateCcy": "BTC"  
    },  
    {  
        "rebate": "0",  
        "rebateCcy": "USDT"  
    }  
],  
"runType": "1",  
"slTriggerPx": "0",  
"state": "stopped",  
"stopResult": "0",  
"stopType": "1",  
"sz": "",  
"tag": "",  
"totalPnl": "0",  
"tpTriggerPx": "0",  
"triggerParams": [  
    {  
        "triggerAction": "start",  
        "delaySeconds": "0",  
        "triggerStrategy": "instant",  
        "triggerType": "auto",  
        "triggerTime": ""  
    },  
    {  
        "triggerAction": "stop",  
        "delaySeconds": "0",  
        "triggerStrategy": "instant",  
        "stopType": "1",  
        "triggerPx": "1000",  
        "triggerType": "manual",  
        "triggerTime": "1681181186484"  
    }  
],  
"uTime": "1681181186496",  
"uly": "BTC-USDT",  
"profitSharingRatio": "",  
"copyType": "0",  
"fee": "",  
"fundingFee": "",  
"tradeQuoteCcy": "USDT"  
},  
],  
"msg": ""  
}
```

Parameter	Type	Description
algoid	String	Algo ID
algoCIOrId	String	Client-supplied Algo ID
instType	String	Instrument type
instId	String	Instrument ID
cTime	String	Algo order created time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
uTime	String	Algo order updated time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
algoOrdType	String	Algo order type <code>grid</code> : Spot grid <code>contract_grid</code> : Contract grid
state	String	Algo order state <code>stopped</code>
rebateTrans	Array of objects	Rebate transfer info
> rebate	String	Rebate amount
> rebateCcy	String	Rebate currency
triggerParams	Array of objects	Trigger Parameters
> triggerAction	String	Trigger action <code>start</code> <code>stop</code>
> triggerStrategy	String	Trigger strategy <code>instant</code> <code>price</code> <code>rsi</code>
> delaySeconds	String	Delay seconds after action triggered
> triggerTime	String	Actual action triggered time, unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> triggerType	String	Actual action triggered type <code>manual</code> <code>auto</code>
> timeframe	String	K-line type <code>3m</code> , <code>5m</code> , <code>15m</code> , <code>30m</code> ( <code>m</code> : minute) <code>1H</code> , <code>4H</code> ( <code>H</code> : hour) <code>1D</code> ( <code>D</code> : day) This field is only valid when <code>triggerStrategy</code> is <code>rsi</code>
> thold	String	Threshold The value should be an integer between 1 to 100 This field is only valid when <code>triggerStrategy</code> is <code>rsi</code>
> triggerCond	String	Trigger condition <code>cross_up</code> <code>cross_down</code> <code>above</code> <code>below</code> <code>cross</code> This field is only valid when <code>triggerStrategy</code> is <code>rsi</code>

Parameter	Type	Description
> timePeriod	String	Time Period 14 This field is only valid when <code>triggerStrategy</code> is <code>rsi</code>
> triggerPx	String	Trigger Price This field is only valid when <code>triggerStrategy</code> is <code>price</code>
> stopType	String	Stop type Spot grid 1: Sell base currency 2: Keep base currency Contract grid 1: Market Close All positions 2: Keep positions This field is only valid when <code>triggerAction</code> is <code>stop</code>
maxPx	String	Upper price of price range
minPx	String	Lower price of price range
gridNum	String	Grid quantity
runType	String	Grid type 1: Arithmetic, 2: Geometric
tpTriggerPx	String	Take-profit trigger price
slTriggerPx	String	Stop-loss trigger price
arbitrageNum	String	The number of arbitrages executed
totalPnl	String	Total P&L
pnlRatio	String	P&L ratio
investment	String	Accumulated investment amount Spot grid investment amount calculated on quote currency
gridProfit	String	Grid profit
floatProfit	String	Variable P&L
cancelType	String	Algo order stop reason 0: None 1: Manual stop 2: Take profit 3: Stop loss 4: Risk control 5: Delivery 6: Signal
stopType	String	Actual Stop type Spot grid 1: Sell base currency 2: Keep base currency Contract grid 1: Market Close All positions 2: Keep positions
quoteSz	String	Quote currency investment amount Only applicable to <code>Spot grid</code>
baseSz	String	Base currency investment amount Only applicable to <code>Spot grid</code>
direction	String	Contract grid type long short neutral Only applicable to <code>contract grid</code>

Parameter	Type	Description
basePos	Boolean	Whether or not to open a position when the strategy is activated Only applicable to <code>contract grid</code>
sz	String	Used margin based on <code>USDT</code> Only applicable to <code>contract grid</code>
lever	String	Leverage Only applicable to <code>contract grid</code>
actualLever	String	Actual Leverage Only applicable to <code>contract grid</code>
liqPx	String	Estimated liquidation price Only applicable to <code>contract grid</code>
uly	String	Underlying Only applicable to <code>contract grid</code>
instFamily	String	Instrument family Only applicable to <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code> Only applicable to <code>contract grid</code>
ordFrozen	String	Margin used by pending orders Only applicable to <code>contract grid</code>
availEq	String	Available margin Only applicable to <code>contract grid</code>
tag	String	Order tag
profitSharingRatio	String	Profit sharing ratio Value range [0, 0.3] If it is a normal order (neither copy order nor lead order), this field returns ""
copyType	String	Profit sharing order type ①: Normal order ②: Copy order without profit sharing ③: Copy order with profit sharing ④: Lead order
fee	String	Accumulated fee. Only applicable to contract grid, or it will be ""
fundingFee	String	Accumulated funding fee. Only applicable to contract grid, or it will be ""
stopResult	String	Stop result ①: default, ②: Successful selling of currency at market price, ③: Failed to sell currency at market price Only applicable to <code>Spot grid</code>
tradeQuoteCcy	String	The quote currency for trading.

#### GET / GRID ALGO ORDER DETAILS

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/tradingBot/grid/orders-algo-details`

Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
algoOrdType	String	Yes	Algo order type <div style="display: flex; justify-content: space-between; align-items: center;"> <span>grid</span>: Spot grid  <span>contract_grid</span>: Contract grid         </div>
algoid	String	Yes	Algo ID

### Response Example

```
{
  "code": "0",
  "data": [
    {
      "actualLever": "",
      "activeOrdNum": "0",
      "algoClOrdId": "",
      "algoId": "448965992920907776",
      "algoOrdType": "grid",
      "annualizedRate": "0",
      "arbitrageNum": "0",
      "availEq": "",
      "basePos": false,
      "baseSz": "0",
      "cTime": "1681181054927",
      "cancelType": "1",
      "curBaseSz": "0",
      "curQuoteSz": "0",
      "direction": "",
      "eq": "",
      "floatProfit": "0",
      "gridNum": "10",
      "gridProfit": "0",
      "instFamily": "",
      "instId": "BTC-USDT",
      "instType": "SPOT",
      "investment": "25",
      "lever": "0",
      "liqPx": "",
      "maxPx": "5000",
      "minPx": "400",
      "ordFrozen": "",
      "perMaxProfitRate": "1.14570215",
      "perMinProfitRate": "0.0991200440528634356837",
      "pnlRatio": "0",
      "profit": "0.00000000",
      "quoteSz": "25",
      "rebateTrans": [
        {
          "rebate": "0",
          "rebateCcy": "BTC"
        },
        {
          "rebate": "0",
          "rebateCcy": "USDT"
        }
      ],
      "runType": "1",
      "runPx": "30089.7",
      "singleAmt": "0.00101214",
      "slTriggerPx": "0",
      "state": "stopped",
      "stopResult": "0",
      "stopType": "1",
      "sz": "",
      "tag": "",
      "totalAnnualizedRate": "0",
      "totalPnl": "0",
      "tpTriggerPx": "0",
      "tradeNum": "0",
      "triggerParams": [
        {
          "triggerAction": "start",
          "delaySeconds": "0",
          "delayType": "seconds"
        }
      ]
    }
  ]
}
```

```

        "triggerStrategy": "instant",
        "triggerType": "auto",
        "triggerTime": ""

    },
    {
        "triggerAction": "stop",
        "delaySeconds": "0",
        "triggerStrategy": "instant",
        "stopType": "1",
        "triggerType": "manual",
        "triggerTime": "1681181186484"
    }
],
"uTime": "1681181186496",
"uly": "",
"profitSharingRatio": "",
"copyType": "0",
"tpRatio": "",
"slRatio": "",
"fee": "",
"fundingFee": "",
"tradeQuoteCcy": "USDT"
}
],
"msg": ""
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
algoCIOrid	String	Client-supplied Algo ID
instType	String	Instrument type
instId	String	Instrument ID
cTime	String	Algo order created time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
uTime	String	Algo order updated time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
algoOrdType	String	Algo order type <code>grid</code> : Spot grid <code>contract_grid</code> : Contract grid
state	String	Algo order state <code>starting</code> <code>running</code> <code>stopping</code> <code>no_close_position</code> : stopped algo order but have not closed position yet <code>stopped</code>
rebateTrans	Array of objects	Rebate transfer info
> rebate	String	Rebate amount
> rebateCcy	String	Rebate currency
triggerParams	Array of objects	Trigger Parameters
> triggerAction	String	Trigger action <code>start</code> <code>stop</code>
> triggerStrategy	String	Trigger strategy <code>instant</code>

Parameter	Type	Description
		<div style="display: flex; justify-content: space-around;"> <span>price</span> <span>rsi</span> </div>
> delaySeconds	String	Delay seconds after action triggered
> triggerTime	String	Actual action triggered time, unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> triggerType	String	Actual action triggered type <span>manual</span> <span>auto</span>
> timeframe	String	K-line type <span>3m</span> <span>5m</span> <span>15m</span> <span>30m</span> (m: minute) <span>1H</span> <span>4H</span> (H: hour) <span>1D</span> (D: day) This field is only valid when <code>triggerStrategy</code> is <code>rsi</code>
> thold	String	Threshold The value should be an integer between 1 to 100 This field is only valid when <code>triggerStrategy</code> is <code>rsi</code>
> triggerCond	String	Trigger condition <span>cross_up</span> <span>cross_down</span> <span>above</span> <span>below</span> <span>cross</span> This field is only valid when <code>triggerStrategy</code> is <code>rsi</code>
> timePeriod	String	Time Period <span>14</span> This field is only valid when <code>triggerStrategy</code> is <code>rsi</code>
> triggerPx	String	Trigger Price This field is only valid when <code>triggerStrategy</code> is <code>price</code>
> stopType	String	Stop type Spot grid <span>1</span> : Sell base currency <span>2</span> : Keep base currency Contract grid <span>1</span> : Market Close All positions <span>2</span> : Keep positions This field is only valid when <code>triggerAction</code> is <code>stop</code>
maxPx	String	Upper price of price range
minPx	String	Lower price of price range
gridNum	String	Grid quantity
runType	String	Grid type <span>1</span> : Arithmetic, <span>2</span> : Geometric
tpTriggerPx	String	Take-profit trigger price
slTriggerPx	String	Stop-loss trigger price
tradeNum	String	The number of trades executed
arbitrageNum	String	The number of arbitrages executed
singleAmt	String	Amount per grid
perMinProfitRate	String	Estimated minimum Profit margin per grid
perMaxProfitRate	String	Estimated maximum Profit margin per grid

Parameter	Type	Description
runPx	String	Price at launch
totalPnl	String	Total P&L
pnlRatio	String	P&L ratio
investment	String	Accumulated investment amount Spot grid investment amount calculated on quote currency
gridProfit	String	Grid profit
floatProfit	String	Variable P&L
totalAnnualizedRate	String	Total annualized rate
annualizedRate	String	Grid annualized rate
cancelType	String	Algo order stop reason ①: None ②: Manual stop ③: Take profit ④: Stop loss ⑤: Risk control ⑥: Delivery ⑦: Signal
		Stop type Spot grid ①: Sell base currency ②: Keep base currency Contract grid ①: Market Close All positions ②: Keep positions
		Active count of pending sub orders
		Quote currency investment amount Only applicable to <a href="#">Spot grid</a>
		Base currency investment amount Only applicable to <a href="#">Spot grid</a>
		Assets of quote currency currently held Only applicable to <a href="#">Spot grid</a>
		Assets of base currency currently held Only applicable to <a href="#">Spot grid</a>
profit	String	Current available profit based on quote currency Only applicable to <a href="#">Spot grid</a>
stopResult	String	Stop result ①: default, ②: Successful selling of currency at market price, ③: Failed to sell currency at market price Only applicable to <a href="#">Spot grid</a>
direction	String	Contract grid type ①: long, ②: short, ③: neutral Only applicable to <a href="#">contract grid</a>
basePos	Boolean	Whether or not to open a position when the strategy is activated Only applicable to <a href="#">contract grid</a>
sz	String	Used margin based on <a href="#">USDT</a> Only applicable to <a href="#">contract grid</a>

Parameter	Type	Description
lever	String	Leverage Only applicable to <code>contract grid</code>
actualLever	String	Actual Leverage Only applicable to <code>contract grid</code>
liqPx	String	Estimated liquidation price Only applicable to <code>contract grid</code>
uly	String	Underlying Only applicable to <code>contract grid</code>
instFamily	String	Instrument family Only applicable to <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code> Only applicable to <code>contract grid</code>
ordFrozen	String	Margin used by pending orders Only applicable to <code>contract grid</code>
availEq	String	Available margin Only applicable to <code>contract grid</code>
eq	String	Total equity of strategy account Only applicable to <code>contract grid</code>
tag	String	Order tag
profitSharingRatio	String	Profit sharing ratio Value range [0, 0.3] If it is a normal order (neither copy order nor lead order), this field returns ""
copyType	String	Profit sharing order type <input type="radio"/> 0: Normal order <input type="radio"/> 1: Copy order without profit sharing <input type="radio"/> 2: Copy order with profit sharing <input type="radio"/> 3: Lead order
tpRatio	String	Take profit ratio, 0.1 represents 10%
sLRatio	String	Stop loss ratio, 0.1 represents 10%
fee	String	Accumulated fee. Only applicable to contract grid, or it will be ""
fundingFee	String	Accumulated funding fee. Only applicable to contract grid, or it will be ""
tradeQuoteCcy	String	The quote currency for trading.

**GET / GRID ALGO SUB ORDERS**

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/tradingBot/grid/sub-orders`

Request Example

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
algoOrdType	String	Yes	Algo order type grid: Spot grid contract_grid: Contract grid
algold	String	Yes	Algo ID
type	String	Yes	Sub order state live filled
groupId	String	No	Group ID
after	String	No	Pagination of data to return records earlier than the requested <code>ordId</code> .
before	String	No	Pagination of data to return records newer than the requested <code>ordId</code> .
limit	String	No	Number of results per request. The maximum is 100. The default is 100

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "accFillSz": "0",
      "algoClOrdId": "",
      "algoId": "44896599292090776",
      "algoOrdType": "grid",
      "avgPx": "0",
      "cTime": "1653347949771",
      "ccy": "",
      "ctVal": "",
      "fee": "0",
      "feeCcy": "USDC",
      "groupId": "3",
      "instId": "BTC-USDC",
      "instType": "SPOT",
      "lever": "0",
      "ordId": "449109084439187456",
      "ordType": "limit",
      "pnl": "0",
      "posSide": "net",
      "px": "30404.3",
      "rebate": "0",
      "rebateCcy": "USDT",
      "side": "sell",
      "state": "live",
      "sz": "0.00059213",
      "tag": "",
      "tdMode": "cash",
      "uTime": "1653347949831"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
algoClOrdId	String	Client-supplied Algo ID
instType	String	Instrument type
instId	String	Instrument ID

Parameter	Type	Description
algoOrdType	String	Algo order type grid: Spot grid contract_grid: Contract grid
groupId	String	Group ID
ordId	String	Sub order ID
cTime	String	Sub order created time, Unix timestamp format in milliseconds, e.g. 1597026383085
uTime	String	Sub order updated time, Unix timestamp format in milliseconds, e.g. 1597026383085
tdMode	String	Sub order trade mode Margin mode: cross/isolated Non-Margin mode: cash
ccy	String	Margin currency Only applicable to cross MARGIN orders in Futures mode.
ordType	String	Sub order type market: Market order limit: Limit order ioc: Immediate-or-cancel order
sz	String	Sub order quantity to buy or sell
state	String	Sub order state canceled live partially_filled filled cancelling
side	String	Sub order side buy sell
px	String	Sub order price
fee	String	Sub order fee amount
feeCcy	String	Sub order fee currency
rebate	String	Sub order rebate amount
rebateCcy	String	Sub order rebate currency
avgPx	String	Sub order average filled price
accFillSz	String	Sub order accumulated fill quantity
posSide	String	Sub order position side net
pnl	String	Sub order profit and loss
ctVal	String	Contract value Only applicable to FUTURES/SWAP
lever	String	Leverage
tag	String	Order tag

## RATE LIMIT RULE: USER ID

## PERMISSION: READ

## HTTP REQUEST

GET /api/v5/tradingBot/grid/positions

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
algoOrdType	String	Yes	Algo order type contract_grid: Contract grid
algold	String	Yes	Algo ID

Response Example

```
{
  "code": "0",
  "data": [
    {
      "adl": "1",
      "algoClOrdId": "",
      "algoId": "449327675342323712",
      "avgPx": "29215.0142857142857149",
      "cTime": "1653400065917",
      "ccy": "USDT",
      "imr": "2045.386",
      "instId": "BTC-USDT-SWAP",
      "instType": "SWAP",
      "last": "29206.7",
      "lever": "5",
      "liqPx": "661.1684795867162",
      "markPx": "29213.9",
      "mgnMode": "cross",
      "mgnRatio": "217.19370606167573",
      "mmr": "40.90772000000005",
      "notionalUsd": "10216.70307",
      "pos": "35",
      "posSide": "net",
      "uTime": "1653400066938",
      "upl": "1.674999999999818",
      "uplRatio": "0.0008190504784478"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
algoClOrdId	String	Client-supplied Algo ID
instType	String	Instrument type
instId	String	Instrument ID, e.g. BTC-USDT-SWAP
cTime	String	Algo order created time, Unix timestamp format in milliseconds, e.g. 1597026383085
uTime	String	Algo order updated time, Unix timestamp format in milliseconds, e.g. 1597026383085
avgPx	String	Average open price

Parameter	Type	Description
ccy	String	Margin currency
lever	String	Leverage
liqPx	String	Estimated liquidation price
posSide	String	Position side <code>net</code>
pos	String	Quantity of positions
mgnMode	String	Margin mode <code>cross</code> <code>isolated</code>
mgnRatio	String	Maintenance margin ratio
imr	String	Initial margin requirement
mmr	String	Maintenance margin requirement
upl	String	Unrealized profit and loss
uplRatio	String	Unrealized profit and loss ratio
last	String	Latest traded price
notionalUsd	String	Notional value of positions in <code>USD</code>
adl	String	Automatic-Deleveraging, signal area Divided into 5 levels, from 1 to 5, the smaller the number, the weaker the adl intensity.
markPx	String	Mark price

**POST / SPOT GRID WITHDRAW INCOME**

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

`POST /api/v5/tradingBot/grid/withdraw-income`

Request Example

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
algold	String	Yes	Algo ID

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "algoClOrdId": "",
      "algoId": "448965992920907776",
      "profit": "100"
    }
  ]
}
```

]

}

## RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
algoCIOrId	String	Client-supplied Algo ID
profit	String	Withdraw profit

## POST / COMPUTE MARGIN BALANCE

RATE LIMIT: 20 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: TRADE

## HTTP REQUEST

POST /api/v5/tradingBot/grid/compute-margin-balance

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
algold	String	Yes	Algo ID
type	String	Yes	Adjust margin balance type <code>add</code> <code>reduce</code>
amt	String	No	Adjust margin balance amount Default is zero.

Response Example

```
{
  "code": "0",
  "data": [
    {
      "lever": "0.3877200981166066",
      "maxAmt": "1.8309562403342999"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
maxAmt	String	Maximum adjustable margin balance amount
lever	String	Leverage after adjustment of margin balance

## POST / ADJUST MARGIN BALANCE

RATE LIMIT: 20 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: TRADE

## HTTP REQUEST

POST /api/v5/tradingBot/grid/margin-balance

## Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
algold	String	Yes	Algo ID
type	String	Yes	Adjust margin balance type <code>add</code> <code>reduce</code>
amt	String	Conditional	Adjust margin balance amount Either <code>amt</code> or <code>percent</code> is required.
percent	String	Conditional	Adjust margin balance percentage

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoClOrdId": "",
      "algoId": "123456"
    }
  ],
  "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
algoClOrdId	String	Client-supplied Algo ID

### POST / ADD INVESTMENT

It is used to add investment and only applicable to contract grid.

### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: TRADE

### HTTP REQUEST

```
POST /api/v5/tradingBot/grid/adjust-investment
```

## Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
algold	String	Yes	Algo ID
amt	String	Yes	The amount is going to be added
allowReinvestProfit	String	No	Whether reinvesting profits, only applicable to spot grid. <code>true</code> or <code>false</code> . The default is true.

## Response Example

```
{
  "code": "0",
  "data": [
    ...
  ]
}
```

```
{
  "algoId": "448965992920907776"
}
],
"msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
algoid	String	Algo ID

## GET / GRID AI PARAMETER (PUBLIC)

Authentication is not required for this public endpoint.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

**PERMISSION: READ**

## HTTP REQUEST

```
GET /api/v5/tradingBot/grid/ai-param
```

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
algoOrdType	String	Yes	Algo order type <span>grid</span> : Spot grid <span>contract_grid</span> : Contract grid
instId	String	Yes	Instrument ID, e.g. <span>BTC-USDT</span>
direction	String	Conditional	Contract grid type <span>long</span> , <span>short</span> , <span>neutral</span> Required in the case of <span>contract_grid</span>
duration	String	No	Back testing duration <span>7D</span> : 7 Days, <span>30D</span> : 30 Days, <span>180D</span> : 180 Days The default is <span>7D</span> for <span>Spot grid</span> Only <span>7D</span> is available for <span>Contract grid</span>

Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoOrdType": "grid",
      "annualizedRate": "1.5849",
      "ccy": "USDT",
      "direction": "",
      "duration": "7D",
      "gridNum": "5",
      "instId": "BTC-USDT",
      "lever": "0",
      "maxPx": "21373.3",
      "minInvestment": "0.89557758",
      "minPx": "15544.2",
      "perGridProfitRatio": "4.566226200302574",
      "perMaxProfitRate": "0.0733865364573281",
      "perMinProfitRate": "0.0561101403446263",
      "runType": "1",
      "sourceCcy": ""
    }
  ]
}
```

```
],
"msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID, e.g. BTC-USDT-SWAP
algoOrdType	String	Algo order type grid: Spot grid contract_grid: Contract grid
duration	String	Back testing duration 7D: 7 Days, 30D: 30 Days, 180D: 180 Days
gridNum	String	Grid quantity
maxPx	String	Upper price of price range
minPx	String	Lower price of price range
perMaxProfitRate	String	Estimated maximum Profit margin per grid
perMinProfitRate	String	Estimated minimum Profit margin per grid
perGridProfitRatio	String	Per grid profit ratio
annualizedRate	String	Grid annualized rate
minInvestment	String	The minimum invest amount
ccy	String	The invest currency
runType	String	Grid type 1: Arithmetic, 2: Geometric
direction	String	Contract grid type long, short, neutral Only applicable to contract grid
lever	String	Leverage Only applicable to contract grid
sourceCcy	String	Source currency

## POST / COMPUTE MIN INVESTMENT (PUBLIC)

Authentication is not required for this public endpoint.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

**PERMISSION: READ**

**HTTP REQUEST**

`POST /api/v5/tradingBot/grid/min-investment`

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. BTC-USDT-SWAP

Parameter	Type	Required	Description
algoOrdType	String	Yes	Algo order type grid: Spot grid contract_grid: Contract grid
maxPx	String	Yes	Upper price of price range
minPx	String	Yes	Lower price of price range
gridNum	String	Yes	Grid quantity
runType	String	Yes	Grid type ①: Arithmetic, ②: Geometric
direction	String	Conditional	Contract grid type long, short, neutral Only applicable to contract grid
lever	String	Conditional	Leverage Only applicable to contract grid
basePos	Boolean	No	Whether or not open a position when the strategy activates Default is false Neutral contract grid should omit the parameter Only applicable to contract grid
investmentType	String	No	Investment type, only applicable to grid quote base dual
triggerStrategy	String	No	Trigger strategy, instant price rsi
topUpAmt	String	No	Top up amount, only applicable to spot grid
investmentData	Array of objects	No	Invest Data
> amt	String	Yes	Invest amount
> ccy	String	Yes	Invest currency

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "minInvestmentData": [
        {
          "amt": "0.1",
          "ccy": "ETH"
        },
        {
          "amt": "100",
          "ccy": "USDT"
        }
      ],
      "singleAmt": "10"
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
minInvestmentData	Array of objects	Minimum invest Data
> amt	String	Minimum invest amount
> ccy	String	Minimum Invest currency
singleAmt	String	Single grid trading amount In terms of <b>spot grid</b> , the unit is <b>quote currency</b> In terms of <b>contract grid</b> , the unit is <b>contract</b>

## GET / RSI BACK TESTING (PUBLIC)

Authentication is not required for this public endpoint.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

**PERMISSION: READ**

**HTTP REQUEST**

```
GET /api/v5/tradingBot/public/rsi-back-testing
```

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <b>BTC-USDT</b> Only applicable to <b>SPOT</b>
timeframe	String	Yes	K-line type <b>3m</b> , <b>5m</b> , <b>15m</b> , <b>30m</b> ( <b>m</b> : minute) <b>1H</b> , <b>4H</b> ( <b>H</b> : hour) <b>1D</b> ( <b>D</b> : day)
thold	String	Yes	Threshold The value should be an integer between 1 to 100
timePeriod	String	Yes	Time Period <b>14</b>
triggerCond	String	No	Trigger condition <b>cross_up</b> <b>cross_down</b> <b>above</b> <b>below</b> <b>cross</b> Default is <b>cross_down</b>
duration	String	No	Back testing duration <b>1M</b> ( <b>M</b> : month) Default is <b>1M</b>

Response Example

```
{
  "code": "0",
  "data": [
    {
      "triggerNum": "164"
    }
  ],
}
```

```

"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
triggerNum	String	Trigger number

## GET / MAX GRID QUANTITY (PUBLIC)

Authentication is not required for this public endpoint.

Maximum grid quantity can be retrieved from this endpoint. Minimum grid quantity always is 2.

### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: IP

### PERMISSION: READ

### HTTP REQUEST

```
GET /api/v5/tradingBot/grid/grid-quantity
```

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT</code>
runType	String	Yes	Grid type <code>1</code> : Arithmetic <code>2</code> : Geometric
algoOrdType	String	Yes	Algo order type <code>grid</code> : Spot grid <code>contract_grid</code> : Contract grid
maxPx	String	Yes	Upper price of price range
minPx	String	Yes	Lower price of price range
lever	String	Conditional	Leverage, it is required for contract grid

Response Example

```

{
  "code": "0",
  "data": [
    {
      "maxGridQty": "285"
    }
  ],
  "msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
maxGridQty	String	Maximum grid quantity

## WS / SPOT GRID ALGO ORDERS CHANNEL

Retrieve spot grid algo orders. Data will be pushed when triggered by events such as placing/canceling order. It will also be pushed in regular interval according to subscription granularity.

/ws/v5/business (required login)

## Request Example

```

import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/business",
        useServerTime=False
    )
    await ws.start()
    args = [{
        "channel": "grid-orders-spot",
        "instType": "SPOT"
    }]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>grid-orders-spot</code>
> instType	String	Yes	Instrument type <code>SPOT</code> <code>ANY</code>
> instId	String	No	Instrument ID
> algold	String	No	Algo Order ID

## Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {

```

```

"channel": "grid-orders-spot",
"instType": "ANY"
},
"connId": "a4d3ae55"
}

```

## Failure Response Example

```

{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \"subscribe\", \\"args\\\": [{ \\"channel\\\" : \"grid-orders-spot\", \\"instType\\\" : \"FUTURES\"}]}",
  "connId": "a4d3ae55"
}

```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event   
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> instType	String	Yes	Instrument type
> instId	String	No	Instrument ID
> algold	String	No	Algo Order ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

## Push Data Example:

```

{
  "arg": {
    "channel": "grid-orders-spot",
    "instType": "ANY",
    "uid": "44705892343619584"
  },
  "data": [
    {
      "algoClOrdId": "",
      "algoId": "568028283477164032",
      "activeOrdNum": "10",
      "algoOrdType": "grid",
      "annualizedRate": "0",
      "arbitrageNum": "0",
      "baseSz": "0",
      "cTime": "1681700496249",
      "cancelType": "0",
      "curBaseSz": "0",
      "curQuoteSz": "25",
      "floatProfit": "0",
      "gridNum": "10",
      "gridProfit": "0",
      "instId": "BTC-USDT",
      "instType": "SPOT",
      "investment": "25",
      "ordSz": "0",
      "ordType": "grid"
    }
  ]
}

```

```

"maxPx": "5000",
"minPx": "400",
"ptTime": "1682416738467",
"perMaxProfitRate": "1.14570215",
"perMinProfitRate": "0.0991200440528634356837",
"pnlRatio": "0",
"profit": "0",
"quoteSz": "25",
"rebateTrans": [
  {
    "rebate": "0",
    "rebateCcy": "BTC"
  },
  {
    "rebate": "0",
    "rebateCcy": "USDT"
  }
],
"runPx": "30031.7",
"runType": "1",
"triggerParams": [
  {
    "triggerAction": "start",
    "triggerStrategy": "instant",
    "delaySeconds": "0",
    "triggerType": "auto",
    "triggerTime": ""
  },
  {
    "triggerAction": "stop",
    "triggerStrategy": "instant",
    "delaySeconds": "0",
    "stopType": "1",
    "triggerType": "manual",
    "triggerTime": ""
  }
],
"singleAmt": "0.00101214",
"slTriggerPx": "",
"state": "running",
"stopResult": "0",
"stopType": "2",
"tag": "",
"totalAnnualizedRate": "0",
"totalPnl": "0",
"tpTriggerPx": "",
"tradeNum": "0",
"uTime": "1682406665527",
"profitSharingRatio": "",
"copyType": "0",
"tradeQuoteCcy": "USDT"
}
]
}

```

#### RESPONSE PARAMETERS WHEN DATA IS PUSHED.

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> instType	String	Instrument type
> uid	String	User ID
data	Array of objects	Subscribed data
> algold	String	Algo ID
> algoClOrdId	String	Client-supplied Algo ID
> instType	String	Instrument type
> instId	String	Instrument ID
> cTime	String	Algo order created time, Unix timestamp format in milliseconds, e.g. <span style="border: 1px solid #ccc; padding: 2px;">1597026383085</span>

Parameter	Type	Description
> uTime	String	Algo order updated time, Unix timestamp format in milliseconds, e.g. 1597026383085
> algoOrdType	String	Algo order type grid: Spot grid
> state	String	Algo order state starting running stopping stopped
> rebateTrans	Array of objects	Rebate transfer info
>> rebate	String	Rebate amount
>> rebateCcy	String	Rebate currency
> triggerParams	Array of objects	Trigger Parameters
>> triggerAction	String	Trigger action start stop
>> triggerStrategy	String	Trigger strategy instant price rsi
>> delaySeconds	String	Delay seconds after action triggered
>> triggerTime	String	Actual action triggered time, unix timestamp format in milliseconds, e.g. 1597026383085
>> triggerType	String	Actual action triggered type manual auto
>> timeframe	String	K-line type 3m, 5m, 15m, 30m (m: minute) 1H, 4H (H: hour) 1D (D: day) This field is only valid when triggerStrategy is rsi
>> thold	String	Threshold The value should be an integer between 1 to 100 This field is only valid when triggerStrategy is rsi
>> triggerCond	String	Trigger condition cross_up cross_down above below cross This field is only valid when triggerStrategy is rsi
>> timePeriod	String	Time Period 14 This field is only valid when triggerStrategy is rsi
>> triggerPx	String	Trigger Price This field is only valid when triggerStrategy is price
>> stopType	String	Stop type Spot grid 1: Sell base currency 2: Keep base currency

Parameter	Type	Description
		Contract grid <b>1</b> : Market Close All positions <b>2</b> : Keep positions This field is only valid when <code>triggerAction</code> is <code>stop</code>
> maxPx	String	Upper price of price range
> minPx	String	Lower price of price range
> gridNum	String	Grid quantity
> runType	String	Grid type <b>1</b> : Arithmetic, <b>2</b> : Geometric
> tpTriggerPx	String	Take-profit trigger price
> slTriggerPx	String	Stop-loss trigger price
> tradeNum	String	The number of trades executed
> arbitrageNum	String	The number of arbitrages executed
> singleAmt	String	Amount per grid
> perMinProfitRate	String	Estimated minimum Profit margin per grid
> perMaxProfitRate	String	Estimated maximum Profit margin per grid
> runPx	String	Price at launch
> totalPnl	String	Total P&L
> pnlRatio	String	P&L ratio
> investment	String	Investment amount Spot grid investment amount calculated on quote currency
> gridProfit	String	Grid profit
> floatProfit	String	Variable P&L
> totalAnnualizedRate	String	Total annualized rate
> annualizedRate	String	Grid annualized rate
> cancelType	String	Algo order stop reason <b>0</b> : None <b>1</b> : Manual stop <b>2</b> : Take profit <b>3</b> : Stop loss <b>4</b> : Risk control <b>5</b> : Delivery <b>6</b> : Signal
> stopType	String	Stop type <b>1</b> : Sell base currency <b>2</b> : Keep base currency
> quoteSz	String	Quote currency investment amount Only applicable to <code>Spot grid</code>
> baseSz	String	Base currency investment amount Only applicable to <code>Spot grid</code>
> curQuoteSz	String	Assets of quote currency currently held Only applicable to <code>Spot grid</code>

Parameter	Type	Description
> curBaseSz	String	Assets of base currency currently held Only applicable to <code>Spot grid</code>
> profit	String	Current available profit based on quote currency Only applicable to <code>Spot grid</code>
> stopResult	String	Stop result 0: default, 1: Successful selling of currency at market price, -1: Failed to sell currency at market price Only applicable to <code>Spot grid</code>
> activeOrdNum	String	Total count of pending sub orders
> tag	String	Order tag
> profitSharingRatio	String	Profit sharing ratio Value range [0, 0.3] If it is a normal order (neither copy order nor lead order), this field returns ""
> copyType	String	Profit sharing order type 0: Normal order 1: Copy order without profit sharing 2: Copy order with profit sharing 3: Lead order
> pTime	String	Push time of algo grid information, Unix timestamp format in milliseconds, e.g. 1597026383085
> tradeQuoteCcy	String	The quote currency for trading.

#### WS / CONTRACT GRID ALGO ORDERS CHANNEL

Retrieve contract grid algo orders. Data will be pushed when triggered by events such as placing/canceling order. It will also be pushed in regular interval according to subscription granularity.

#### URL PATH

/ws/v5/business (required login)

#### Request Example

```
import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/business",
        useServerTime=False
    )
    await ws.start()
    args = [{"channel": "grid-orders-contract",
              "instType": "SWAP"}]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation subscribe unsubscribe
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name grid-orders-contract
> instType	String	Yes	Instrument type SWAP FUTURES ANY
> instId	String	No	Instrument ID
> algId	String	No	Algo Order ID

## Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "grid-orders-contract",
    "instType": "ANY"
  },
  "connId": "a4d3ae55"
}
```

## Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \\"subscribe\\", \\"args\\": [{ \\"channel\\": \\"grid-orders-contract\\", \\"instType\\": \\"FUTURES\\"}]}",
  "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event subscribe unsubscribe error
arg	Object	No	Subscribed channel

Parameter	Type	Required	Description
> channel	String	Yes	Channel name
> instType	String	Yes	Instrument type
> instId	String	No	Instrument ID
> algold	String	No	Algo Order ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

Push Data Example:

```
{
  "arg": {
    "channel": "grid-orders-contract",
    "instType": "ANY",
    "uid": "4470****9584"
  },
  "data": [
    {
      "actualLever": "2.3481494635276649",
      "activeOrdNum": "10",
      "algoClOrdId": "",
      "algoId": "571039869070475264",
      "algoOrdType": "contract_grid",
      "annualizedRate": "0",
      "arbitrageNum": "0",
      "availEq": "52.3015392887089673",
      "basePos": true,
      "cTime": "1682418514204",
      "cancelType": "0",
      "direction": "long",
      "eq": "108.7945652387089673",
      "floatProfit": "8.7945652387089673",
      "gridNum": "10",
      "gridProfit": "0",
      "instId": "BTC-USDT-SWAP",
      "instType": "SWAP",
      "investment": "100",
      "lever": "5",
      "liqPx": "16370.482143120824",
      "maxPx": "36437.3",
      "minPx": "26931.9",
      "ordFrozen": "5.38638",
      "pTime": "1682492574068",
      "perMaxProfitRate": "0.1687494513302446",
      "perMinProfitRate": "0.1263869357706788",
      "pnlRatio": "0.0879456523870897",
      "rebateTrans": [
        {
          "rebate": "0",
          "rebateCcy": "USDT"
        }
      ],
      "runPx": "27306.9",
      "runType": "1",
      "singleAmt": "1",
      "slTriggerPx": "",
      "state": "running",
      "stopType": "0",
      "sz": "100",
      "tag": "",
      "totalAnnualizedRate": "38.52019574554529",
      "totalPnl": "8.7945652387089673",
      "tpTriggerPx": "",
      "tradeNum": "9",
      "triggerParams": [
        {
          "triggerAction": "start",
          "delaySeconds": "0",
          "triggerStrategy": "price",
          "triggerType": "grid"
        }
      ]
    }
  ]
}
```

```

  "triggerPx": "1",
  "triggerType": "manual",
  "triggerTime": "1682418561497"
}, {
  "triggerAction": "stop",
  "delaySeconds": "0",
  "triggerStrategy": "instant",
  "stopType": "1",
  "triggerType": "manual",
  "triggerTime": "0"
}],
"uTime": "1682492552257",
"profitSharingRatio": "",
"copyType": "0",
"tpRatio": "",
"slRatio": "",
"fee": "",
"fundingFee": ""
}
}

```

#### RESPONSE PARAMETERS WHEN DATA IS PUSHED.

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> instType	String	Instrument type
> uid	String	User ID
data	Array of objects	Subscribed data
> algold	String	Algo ID
> algoClOrdId	String	Client-supplied Algo ID
> instType	String	Instrument type
> instId	String	Instrument ID
> cTime	String	Algo order created time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> uTime	String	Algo order updated time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> algoOrdType	String	Algo order type <code>contract_grid</code> : Contract grid
> state	String	Algo order state <code>starting</code> <code>running</code> <code>stopping</code> <code>no_close_position</code> : stopped algo order but hadn't close position yet <code>stopped</code>
> rebateTrans	Array of objects	Rebate transfer info
>> rebate	String	Rebate amount
>> rebateCcy	String	Rebate currency
> triggerParams	Array of objects	Trigger Parameters
>> triggerAction	String	Trigger action <code>start</code> <code>stop</code>

Parameter	Type	Description
>> triggerStrategy	String	<p>Trigger strategy</p> <div style="display: flex; justify-content: space-around;"> <span>instant</span> <span>price</span> <span>rsi</span> </div>
>> delaySeconds	String	Delay seconds after action triggered
>> triggerTime	String	Actual action triggered time, unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
>> triggerType	String	<p>Actual action triggered type</p> <div style="display: flex; justify-content: space-around;"> <span>manual</span> <span>auto</span> </div>
>> timeframe	String	<p>K-line type</p> <div style="display: flex; justify-content: space-around;"> <span>3m</span> <span>5m</span> <span>15m</span> <span>30m</span> <span>(m: minute)</span> </div> <div style="display: flex; justify-content: space-around;"> <span>1H</span> <span>4H</span> <span>(H: hour)</span> </div> <div style="display: flex; justify-content: space-around;"> <span>1D</span> <span>(D: day)</span> </div> <p>This field is only valid when <code>triggerStrategy</code> is <code>rsi</code></p>
>> thold	String	<p>Threshold</p> <p>The value should be an integer between 1 to 100</p> <p>This field is only valid when <code>triggerStrategy</code> is <code>rsi</code></p>
>> triggerCond	String	<p>Trigger condition</p> <div style="display: flex; justify-content: space-around;"> <span>cross_up</span> <span>cross_down</span> </div> <div style="display: flex; justify-content: space-around;"> <span>above</span> <span>below</span> </div> <div style="display: flex; justify-content: space-around;"> <span>cross</span> </div> <p>This field is only valid when <code>triggerStrategy</code> is <code>rsi</code></p>
>> timePeriod	String	<p>Time Period</p> <div style="display: flex; justify-content: space-around;"> <span>14</span> </div> <p>This field is only valid when <code>triggerStrategy</code> is <code>rsi</code></p>
>> triggerPx	String	<p>Trigger Price</p> <p>This field is only valid when <code>triggerStrategy</code> is <code>price</code></p>
>> stopType	String	<p>Stop type</p> <p>Spot grid ①: Sell base currency ②: Keep base currency</p> <p>Contract grid ①: Market Close All positions ②: Keep positions</p> <p>This field is only valid when <code>triggerAction</code> is <code>stop</code></p>
> maxPx	String	Upper price of price range
> minPx	String	Lower price of price range
> gridNum	String	Grid quantity
> runType	String	<p>Grid type</p> <div style="display: flex; justify-content: space-around;"> <span>①: Arithmetic</span> <span>②: Geometric</span> </div>
> tpTriggerPx	String	Take-profit trigger price
> slTriggerPx	String	Stop-loss trigger price
> tradeNum	String	The number of trades executed
> arbitrageNum	String	The number of arbitrages executed
> singleAmt	String	Amount per grid

Parameter	Type	Description
> perMinProfitRate	String	Estimated minimum Profit margin per grid
> perMaxProfitRate	String	Estimated maximum Profit margin per grid
> runPx	String	Price at launch
> totalPnl	String	Total P&L
> pnlRatio	String	P&L ratio
> investment	String	Accumulated investment amount Spot grid investment amount calculated on quote currency
> gridProfit	String	Grid profit
> floatProfit	String	Variable P&L
> totalAnnualizedRate	String	Total annualized rate
> annualizedRate	String	Grid annualized rate
> cancelType	String	Algo order stop reason ①: None ②: Manual stop ③: Take profit ④: Stop loss ⑤: Risk control ⑥: Delivery ⑦: Signal
> stopType	String	Stop type Spot grid ①: Sell base currency ②: Keep base currency Contract grid ①: Market Close All positions ②: Keep positions
> direction	String	Contract grid type long, short, neutral Only applicable to <code>contract_grid</code>
> basePos	Boolean	Whether or not to open a position when the strategy is activated Only applicable to <code>contract_grid</code>
> sz	String	Used margin based on <code>USDT</code> Only applicable to <code>contract_grid</code>
> lever	String	Leverage Only applicable to <code>contract_grid</code>
> actualLever	String	Actual Leverage Only applicable to <code>contract_grid</code>
> liqPx	String	Estimated liquidation price Only applicable to <code>contract_grid</code>
> ordFrozen	String	Margin used by pending orders Only applicable to <code>contract_grid</code>
> availEq	String	Available margin Only applicable to <code>contract_grid</code>
> eq	String	Total equity of strategy account Only applicable to <code>contract_grid</code>

Parameter	Type	Description
> activeOrdNum	String	Total count of pending sub orders
> tag	String	Order tag
> profitSharingRatio	String	Profit sharing ratio Value range [0, 0.3] If it is a normal order (neither copy order nor lead order), this field returns ""
> copyType	String	Profit sharing order type ①: Normal order ②: Copy order without profit sharing ③: Copy order with profit sharing ④: Lead order
> tpRatio	String	Take profit ratio, 0.1 represents 10%
> slRatio	String	Stop loss ratio, 0.1 represents 10%
> fee	String	Accumulated fee. Only applicable to contract grid, or it will be ""
> fundingFee	String	Accumulated funding fee. Only applicable to contract grid, or it will be ""
> pTime	String	Push time of algo grid information, Unix timestamp format in milliseconds, e.g. [1597026383085]

#### WS / GRID POSITIONS CHANNEL

Retrieve contract grid positions. Data will be pushed when triggered by events such as placing/canceling order.

Please ignore the empty data.

#### URL PATH

/ws/v5/business (required login)

Request Example

```
import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/business",
        useServerTime=False
    )
    await ws.start()
    args = [
        {"channel": "grid-positions",
        "algoId": "449327675342323712"
    }]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>grid-positions</code>
> algold	String	Yes	Algo Order ID

**Successful Response Example**

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "grid-positions",
    "algoId": "449327675342323712"
  },
  "connId": "a4d3ae55"
}
```

**Failure Response Example**

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \"subscribe\", \\"argss\\\": [{ \\"channel\\\" : \"grid-positions\", \\"instType\\\" : \"FUTURES\"}]}",
  "connId": "a4d3ae55"
}
```

**RESPONSE PARAMETERS**

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event <code>subscribe</code> <code>unsubscribe</code> <code>error</code>
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> algold	String	Yes	Algo Order ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

**Push Data Example:**

```
{
  "arg": {
    "channel": "grid-positions",
    "uid": "4470****9584",
    "algoId": "449327675342323712"
  },
  "data": [
    {
      "adl": "1",
      "algoClOrdId": "",
      "algoId": "449327675342323712",
      "avgPx": "29181.463888888888895",
      "cTime": "1653400065917",
      "ccy": "USDT",
      "imr": "2089.269000000002",
      "instId": "BTC-USDT-SWAP",
      "instType": "SWAP",
      "last": "29852.7",
      "lever": "5",
      "liqPx": "604.7617536513744",
      "markPx": "29849.7",
      "mgnMode": "cross",
      "mgnRatio": "217.71740878394456",
      "mmr": "41.78538",
      "notionalUsd": "10435.794191550001",
      "pTime": "1653536068723",
      "pos": "35",
      "posSide": "net",
      "uTime": "1653445498682",
      "upl": "232.83263888888962",
      "uplRatio": "0.1139826489932205"
    }
  ]
}
```

#### RESPONSE PARAMETERS WHEN DATA IS PUSHED.

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> uid	String	User Identifier
> algoId	String	Algo Order ID
data	Array of objects	Subscribed data
> algoId	String	Algo ID
> algoClOrdId	String	Client-supplied Algo ID
> instType	String	Instrument type
> instId	String	Instrument ID
> cTime	String	Algo order created time, Unix timestamp format in milliseconds, e.g. 1597026383085
> uTime	String	Algo order updated time, Unix timestamp format in milliseconds, e.g. 1597026383085
> avgPx	String	Average open price
> ccy	String	Margin currency
> lever	String	Leverage
> liqPx	String	Estimated liquidation price
> posSide	String	Position side net

Parameter	Type	Description
> pos	String	Quantity of positions
> mgnMode	String	Margin mode <code>cross</code> <code>isolated</code>
> mgnRatio	String	Maintenance margin ratio
> imr	String	Initial margin requirement
> mmr	String	Maintenance margin requirement
> upl	String	Unrealized profit and loss
> uplRatio	String	Unrealized profit and loss ratio
> last	String	Latest traded price
> notionalUsd	String	Notional value of positions in <code>USD</code>
> adl	String	Automatic-Deleveraging, signal area Divided into 5 levels, from 1 to 5, the smaller the number, the weaker the adl intensity.
> markPx	String	Mark price
> pTime	String	Push time of positions information, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

#### WS / GRID SUB ORDERS CHANNEL

Retrieve grid sub orders. Data will be pushed when triggered by events such as placing order.

Please ignore the empty data.

#### URL PATH

/ws/v5/business (required login)

#### Request Example

```
import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/business",
        useServerTime=False
    )
    await ws.start()
    args = [{{
        "channel": "grid-sub-orders",
        "algoId": "449327675342323712"
    }}]

    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>grid-sub-orders</code>
> algold	String	Yes	Algo Order ID

## Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "grid-sub-orders",
    "algoId": "449327675342323712"
  },
  "connId": "a4d3ae55"
}
```

## Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \\"subscribe\\\", \\"argss\\\": [{ \\"channel\\\" : \\"grid-sub-orders\\\", \\"instType\\\" : \\"FUTURES\\\" }]}",
  "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event <code>subscribe</code> <code>unsubscribe</code> <code>error</code>
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> algold	String	Yes	Algo Order ID
code	String	No	Error code
msg	String	No	Error message

Parameter	Type	Required	Description
connId	String	Yes	WebSocket connection ID

Push Data Example:

```
{
  "arg": {
    "channel": "grid-sub-orders",
    "uid": "44705892343619584",
    "algoId": "449327675342323712"
  },
  "data": [
    {
      "accFillSz": "0",
      "algoClOrdId": "",
      "algoId": "449327675342323712",
      "algoOrdType": "contract_grid",
      "avgPx": "0",
      "cTime": "1653445498664",
      "ctVal": "0.01",
      "fee": "0",
      "feeCcy": "USDT",
      "groupId": "-1",
      "instId": "BTC-USDT-SWAP",
      "instType": "SWAP",
      "leven": "5",
      "ordId": "449518234142904321",
      "ordType": "limit",
      "pTime": "1653486524502",
      "pnl": "",
      "posSide": "net",
      "px": "28007.2",
      "rebate": "0",
      "rebateCcy": "USDT",
      "side": "buy",
      "state": "live",
      "sz": "1",
      "tag": "",
      "tdMode": "cross",
      "uTime": "1653445498674"
    }
  ]
}
```

#### RESPONSE PARAMETERS WHEN DATA IS PUSHED.

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> uid	String	User Identifier
> algoid	String	Algo Order ID
data	Array of objects	Subscribed data
> algoid	String	Algo ID
> algoClOrdId	String	Client-supplied Algo ID
> instType	String	Instrument type
> instId	String	Instrument ID
> algoOrdType	String	Algo order type grid: Spot grid contract_grid: Contract grid

Parameter	Type	Description
> groupId	String	Group ID
> ordId	String	Sub order ID
> cTime	String	Sub order created time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> uTime	String	Sub order updated time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> tdMode	String	Sub order trade mode Margin mode <code>cross</code> <code>isolated</code> Non-Margin mode <code>cash</code>
> tag	String	Order tag
> ordType	String	Sub order type <code>market</code> : Market order <code>limit</code> : Limit order <code>ioc</code> : Immediate-or-cancel order
> sz	String	Sub order quantity to buy or sell
> state	String	Sub order state <code>canceled</code> <code>live</code> <code>partially_filled</code> <code>filled</code> <code>canceling</code>
> side	String	Sub order side <code>buy</code> <code>sell</code>
> px	String	Sub order price
> fee	String	Sub order fee amount
> feeCcy	String	Sub order fee currency
> rebate	String	Sub order rebate amount
> rebateCcy	String	Sub order rebate currency
> avgPx	String	Sub order average filled price
> accFillSz	String	Sub order accumulated fill quantity
> posSide	String	Sub order position side <code>net</code>
> pnl	String	Sub order profit and loss
> ctVal	String	Contract value Only applicable to <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>
> lever	String	Leverage
> pTime	String	Push time of orders information, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

Create and customize your own signals while gaining access to a diverse selection of signals from top providers. Empower your trading strategies and stay ahead of the game with our comprehensive signal trading platform. Learn more

## POST / CREATE SIGNAL

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

```
POST /api/v5/tradingBot/signal/create-signal
```

Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
signalChanName	String	Yes	Signal channel name
signalChanDesc	String	No	Signal channel description

Response Example

```
{
  "code": "0",
  "data": [
    {
      "signalChanId": "572112109",
      "signalChanToken": "dojuckew3311kx"
    }
  ],
  "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
signalChanId	String	Signal channel Id
signalChanToken	String	User identify when placing orders via signal

## GET / SIGNALS

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

```
GET /api/v5/tradingBot/signal/signals
```

Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
signalSourceType	String	Yes	Signal source type ①: Created by yourself ②: Subscribe ③: Free signal

Parameter	Type	Required	Description
signalChanId	String	No	Signal channel id
after	String	No	Pagination of data to return records <code>signalChanId</code> earlier than the requested timestamp, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
before	String	No	Pagination of data to return records <code>signalChanId</code> newer than the requested timestamp, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. The maximum is 100. The default is 100.

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "signalChanId": "623833708424069120",
      "signalChanName": "test",
      "signalChanDesc": "test",
      "signalChanToken": "test",
      "signalSourceType": "1"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
signalChanId	String	Signal channel id
signalChanName	String	Signal channel name
signalChanDesc	String	Signal channel description
signalChanToken	String	User identify when placing orders via signal
signalSourceType	String	Signal source type ①: Created by yourself ②: Subscribe ③: Free signal

#### POST / CREATE SIGNAL BOT

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

```
POST /api/v5/tradingBot/signal/order-algo
```

#### Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
signalChanId	String	Yes	Signal channel Id

Parameter	Type	Required	Description
lever	String	Yes	Leverage Only applicable to <code>contract signal</code>
investAmt	String	Yes	Investment amount
subOrdType	String	Yes	Sub order type ①: limit order ②: market order ⑨: tradingView signal
includeAll	Boolean	No	Whether to include all USDT-margined contract. The default value is <code>false</code> . <code>true</code> : include <code>false</code> : exclude
instIds	String	No	Instrument IDs. Single currency or multiple currencies separated with comma. When <code>includeAll</code> is <code>true</code> , it is ignored
ratio	String	No	Price offset ratio, calculate the limit price as a percentage offset from the best bid/ask price. Only applicable to <code>subOrdType</code> is <code>limit</code> order
entrySettingParam	String	No	Entry setting
> allowMultipleEntry	String	No	Whether or not allow multiple entries in the same direction for the same trading pairs. The default value is <code>true</code> . <code>true</code> : Allow <code>false</code> : Prohibit
> entryType	String	No	Entry type ①: TradingView signal ②: Fixed margin ③: Contracts ④: Percentage of free margin ⑤: Percentage of the initial invested margin
> amt	String	No	Amount per order Only applicable to <code>entryType</code> in ②/③
> ratio	Array of objects	No	Amount ratio per order Only applicable to <code>entryType</code> in ④/⑤
exitSettingParam	String	No	Exit setting
> tpSIType	String	是	Type of set the take-profit and stop-loss trigger price ①: Based on the estimated profit and loss percentage from the entry point ②: Based on price increase or decrease from the crypto's entry price
> tpPct	String	No	Take-profit percentage
> slPct	String	No	Stop-loss percentage

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoClOrdId": "",
      "algoId": "447053782921515008",
      "sCode": "0",
      "sMsg": ""
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
algoClOrdId	String	Client-supplied Algo ID
sCode	String	The code of the event execution result, 0 means success.
sMsg	String	The code of the event execution result, 0 means success.

#### POST / CANCEL SIGNAL BOTS

A maximum of 10 orders can be stopped per request.

#### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: USER ID

#### PERMISSION: TRADE

#### HTTP REQUEST

POST /api/v5/tradingBot/signal/stop-order-algo

Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
algold	String	Yes	Algo ID

Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoId": "448965992920907776",
      "sCode": "0",
      "sMsg": "",
      "algoClOrdId": ""
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
sCode	String	The code of the event execution result, 0 means success.
sMsg	String	Rejection or success message of event execution.

#### POST / ADJUST MARGIN BALANCE

#### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: USER ID

#### PERMISSION: TRADE

#### HTTP REQUEST

POST /api/v5/tradingBot/signal/margin-balance

## Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
algold	String	Yes	Algo ID
type	String	Yes	Adjust margin balance type add   reduce
amt	String	Yes	Adjust margin balance amount Either <code>amt</code> or <code>percent</code> is required.
allowReinvest	Boolean	No	Whether to reinvest with newly added margin. The default value is <code>false</code> . <code>false</code> : it will be used as passive margin to prevent liquidation and will not be used as active investment <code>true</code> : the margin added here will furthermore be accounted for in calculations of your total investment amount, and furthermore your order size. Only applicable to your signal comes in with an "investmentType" of "percentage_investment"

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoId": "123456"
    }
  ],
  "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID

### POST / AMEND TPSL

RATE LIMIT: 20 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: TRADE

### HTTP REQUEST

POST /api/v5/tradingBot/signal/amendTPSL

## Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
algold	String	Yes	Algo ID
exitSettingParam	String	Yes	Exit setting
> tpSIType	String	Yes	Type of set the take-profit and stop-loss trigger price <code>pnl</code> : Based on the estimated profit and loss percentage from the entry point <code>price</code> : Based on price increase or decrease from the crypto's entry price
> tpPct	String	No	Take-profit percentage
> slPct	String	No	Stop-loss percentage

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoId": "637039348240277504"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID

## POST / SET INSTRUMENTS

RATE LIMIT: 20 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: TRADE

HTTP REQUEST

POST /api/v5/tradingBot/signal/set-instruments

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
algold	String	Yes	Algo ID
instIds	Array of strings	Yes	Instrument IDs. When <code>includeAll</code> is <code>true</code> , it is ignored
includeAll	Boolean	Yes	Whether to include all USDT-margined contract. The default value is <code>false</code> . <code>true</code> : include <code>false</code> : exclude

Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoId": "637039348240277504"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID

## GET / SIGNAL BOT ORDER DETAILS

RATE LIMIT: 20 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: READ

HTTP REQUEST

GET /api/v5/tradingBot/signal/orders-algo-details

## Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
algoOrdType	String	Yes	Algo order type contract: Contract signal
algold	String	Yes	Algo ID

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoId": "623833708424069120",
      "algoClOrdId": "",
      "algoOrdType": "contract",
      "availBal": "1.6561369013122267",
      "cTime": "1695005546360",
      "cancelType": "0",
      "entrySettingParam": {
        "allowMultipleEntry": true,
        "amt": "0",
        "entryType": "1",
        "ratio": ""
      },
      "exitSettingParam": {
        "slPct": "",
        "tpPct": "",
        "tpSlType": "price"
      },
      "floatPnl": "0.1279999999999927",
      "frozenBal": "25.16816",
      "instIds": [
        "BTC-USDT-SWAP",
        "ETH-USDT-SWAP"
      ],
      "instType": "SWAP",
      "investAmt": "100",
      "lever": "10",
      "ratio": "",
      "realizedPnl": "-73.303703098687766",
      "signalChaniId": "623827579484770304",
      "signalChanName": "testing",
      "signalSourceType": "1",
      "state": "running",
      "subOrdType": "9",
      "totalEq": "26.824296901312227",
      "totalPnl": "-73.175703098687733",
      "totalPnlRatio": "-0.7317570309868777",
      "uTime": "1697029422313"
    }
  ],
  "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
algoClOrdId	String	Client-supplied Algo ID
instType	String	Instrument type
instIds	Array of strings	Instrument IDs

Parameter	Type	Description
cTime	String	Algo order created time, Unix timestamp format in milliseconds, e.g. 1597026383085
uTime	String	Algo order updated time, Unix timestamp format in milliseconds, e.g. 1597026383085
algoOrdType	String	Algo order type contract: Contract signal
state	String	Algo order state starting running stopping stopped
cancelType	String	Algo order stop reason 0: None 1: Manual stop
totalPnl	String	Total P&L
totalPnlRatio	String	Total P&L ratio
totalEq	String	Total equity of strategy account
floatPnl	String	Float P&L
realizedPnl	String	Realized P&L
frozenBal	String	Frozen balance
availBal	String	Avail balance
lever	String	Leverage Only applicable to contract signal
investAmt	String	Investment amount
subOrdType	String	Sub order type 1: limit order 2: market order 9: tradingView signal
ratio	String	Price offset ratio, calculate the limit price as a percentage offset from the best bid/ask price Only applicable to subOrdType is limit order
entrySettingParam	Object	Entry setting
> allowMultipleEntry	Boolean	Whether or not allow multiple entries in the same direction for the same trading pairs
> entryType	String	Entry type 1: TradingView signal 2: Fixed margin 3: Contracts 4: Percentage of free margin 5: Percentage of the initial invested margin
> amt	String	Amount per order Only applicable to entryType in 2/3
> ratio	String	Amount ratio per order Only applicable to entryType in 4/5
exitSettingParam	Object	Exit setting

Parameter	Type	Description
> tpSIType	String	Type of set the take-profit and stop-loss trigger price pn1: Based on the estimated profit and loss percentage from the entry point price: Based on price increase or decrease from the crypto's entry price
> tpPct	String	Take-profit percentage
> slPct	String	Stop-loss percentage
signalChanId	String	Signal channel Id
signalChanName	String	Signal channel name
signalSourceType	String	Signal source type 1: Created by yourself 2: Subscribe 3: Free signal

#### GET / ACTIVE SIGNAL BOT

RATE LIMIT: 20 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: READ

#### HTTP REQUEST

GET /api/v5/tradingBot/signal/orders-algo-pending

Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
algoOrdType	String	Yes	Algo order type contract: Contract signal
algoid	String	No	Algo ID
after	String	Yes	Pagination of data to return records <code>algoid</code> earlier than the requested timestamp, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
before	String	No	Pagination of data to return records <code>algoid</code> newer than the requested timestamp, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. The maximum is 100. The default is 100.

Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoId": "623833708424069120",
      "algoClOrdId": "",
      "algoOrdType": "contract",
      "availBal": "1.6561369013122267",
      "cTime": "1695005546360",
      "cancelType": "0",
      "entrySettingParam": {
        "allowMultipleEntry": true,
        "amt": "0",
        "entryType": "1",
        "ratio": ""
      },
      "exitSettingParam": {
        "allowMultipleEntry": true,
        "amt": "0",
        "entryType": "1",
        "ratio": ""
      }
    }
  ]
}
```

```

        "slPct": "",
        "tpPct": "",
        "tpSlType": "price"
    },
    "floatPnl": "0.1279999999999927",
    "frozenBal": "25.16816",
    "instIds": [
        "BTC-USDT-SWAP",
        "ETH-USDT-SWAP"
    ],
    "instType": "SWAP",
    "investAmt": "100",
    "lever": "10",
    "ratio": "",
    "realizedPnl": "-73.303703098687766",
    "signalChnId": "623827579484770304",
    "signalChanName": "my signal",
    "signalSourceType": "1",
    "state": "running",
    "subOrdType": "9",
    "totalEq": "26.824296901312227",
    "totalPnl": "-73.1757030986877733",
    "totalPnlRatio": "-0.7317570309868777",
    "uTime": "1697029422313"
}
],
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
algoClOrdId	String	Client-supplied Algo ID
instType	String	Instrument type
instIds	Array of strings	Instrument IDs
cTime	String	Algo order created time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
uTime	String	Algo order updated time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
algoOrdType	String	Algo order type <code>contract</code> : Contract signal
state	String	Algo order state <code>starting</code> <code>running</code> <code>stopping</code>
cancelType	String	Algo order stop reason <code>0</code> : None
totalPnl	String	Total P&L
totalPnlRatio	String	Total P&L ratio
totalEq	String	Total equity of strategy account
floatPnl	String	Float P&L
realizedPnl	String	Realized P&L
frozenBal	String	Frozen balance
availBal	String	Avail balance

Parameter	Type	Description
lever	String	Leverage Only applicable to <code>contract signal</code>
investAmt	String	Investment amount
subOrdType	String	Sub order type ①: limit order ②: market order ⑨: tradingView signal
ratio	String	Price offset ratio, calculate the limit price as a percentage offset from the best bid/ask price Only applicable to <code>subOrdType</code> is <code>limit order</code>
entrySettingParam	Object	Entry setting
> allowMultipleEntry	Boolean	Whether or not allow multiple entries in the same direction for the same trading pairs
> entryType	String	Entry type ①: TradingView signal ②: Fixed margin ③: Contracts ④: Percentage of free margin ⑤: Percentage of the initial invested margin
> amt	String	Amount per order Only applicable to <code>entryType</code> in ②/③
> ratio	String	Amount ratio per order Only applicable to <code>entryType</code> in ④/⑤
exitSettingParam	Object	Exit setting
> tpSIType	String	Type of set the take-profit and stop-loss trigger price ①: Based on the estimated profit and loss percentage from the entry point ②: Based on price increase or decrease from the crypto's entry price
> tpPct	String	Take-profit percentage
> slPct	String	Stop-loss percentage
signalChanId	String	Signal channel Id
signalChanName	String	Signal channel name
signalSourceType	String	Signal source type ①: Created by yourself ②: Subscribe ③: Free signal

#### GET / SIGNAL BOT HISTORY

RATE LIMIT: 20 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: READ

#### HTTP REQUEST

`GET /api/v5/tradingBot/signal/orders-algo-history`

Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
algoOrdType	String	Yes	Algo order type contract: Contract signal
algold	String	Yes	Algo ID
after	String	Yes	Pagination of data to return records <code>algold</code> earlier than the requested timestamp, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
before	String	No	Pagination of data to return records <code>algold</code> newer than the requested timestamp, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. The maximum is 100. The default is 100.

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoId": "623833708424069120",
      "algoClOrdId": "",
      "algoOrdType": "contract",
      "availBal": "1.6561369013122267",
      "cTime": "1695005546360",
      "cancelType": "1",
      "entrySettingParam": {
        "allowMultipleEntry": true,
        "amt": "0",
        "entryType": "1",
        "ratio": ""
      },
      "exitSettingParam": {
        "slPct": "",
        "tpPct": "",
        "tpSlType": "price"
      },
      "floatPnl": "0.127999999999927",
      "frozenBal": "25.16816",
      "instIds": [
        "BTC-USDT-SWAP",
        "ETH-USDT-SWAP"
      ],
      "instType": "SWAP",
      "investAmt": "100",
      "lever": "10",
      "ratio": "",
      "realizedPnl": "-73.303703098687766",
      "signalChnlId": "623827579484770304",
      "signalChnlName": "my signal",
      "signalSourceType": "1",
      "state": "stopped",
      "subOrdType": "9",
      "totalEq": "26.824296901312227",
      "totalPnl": "-73.1757030986877733",
      "totalPnlRatio": "-0.7317570309868777",
      "uTime": "1697029422313"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
algoClOrdId	String	Client-supplied Algo ID

Parameter	Type	Description
instType	String	Instrument type
instIds	Array of strings	Instrument IDs
cTime	String	Algo order created time, Unix timestamp format in milliseconds, e.g. 1597026383085
uTime	String	Algo order updated time, Unix timestamp format in milliseconds, e.g. 1597026383085
algoOrdType	String	Algo order type contract: Contract signal
state	String	Algo order state stopped
cancelType	String	Algo order stop reason 1: Manual stop
totalPnl	String	Total P&L
totalPnlRatio	String	Total P&L ratio
totalEq	String	Total equity of strategy account
floatPnl	String	Float P&L
realizedPnl	String	Realized P&L
frozenBal	String	Frozen balance
availBal	String	Avail balance
lever	String	Leverage Only applicable to contract signal
investAmt	String	Investment amount
subOrdType	String	Sub order type 1: limit order 2: market order 9: tradingView signal
ratio	String	Price offset ratio, calculate the limit price as a percentage offset from the best bid/ask price Only applicable to subOrdType is limit order
entrySettingParam	Object	Entry setting
> allowMultipleEntry	Boolean	Whether or not allow multiple entries in the same direction for the same trading pairs
> entryType	String	Entry type 1: TradingView signal 2: Fixed margin 3: Contracts 4: Percentage of free margin 5: Percentage of the initial invested margin
> amt	String	Amount per order Only applicable to entryType in 2/3
> ratio	String	Amount ratio per order Only applicable to entryType in 4/5
exitSettingParam	Object	Exit setting

Parameter	Type	Description
> tpSIType	String	Type of set the take-profit and stop-loss trigger price pn1: Based on the estimated profit and loss percentage from the entry point price: Based on price increase or decrease from the crypto's entry price
> tpPct	String	Take-profit percentage
> slPct	String	Stop-loss percentage
signalChanId	String	Signal channel Id
signalChanName	String	Signal channel name
signalSourceType	String	Signal source type 1: Created by yourself 2: Subscribe 3: Free signal

#### GET / SIGNAL BOT ORDER POSITIONS

RATE LIMIT: 20 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: READ

#### HTTP REQUEST

GET /api/v5/tradingBot/signal/positions

Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
algoOrdType	String	Yes	Algo order type contract: Contract signal
algoid	String	Yes	Algo ID

Response Example

```
{
  "code": "0",
  "data": [
    {
      "ad1": "1",
      "algoClOrdId": "",
      "algoId": "623833708424069120",
      "avgPx": "1597.74",
      "cTime": "1697502301460",
      "ccy": "USDT",
      "imr": "23.76495",
      "instId": "ETH-USDT-SWAP",
      "instType": "SWAP",
      "last": "1584.34",
      "lever": "10",
      "liqPx": "1438.7380360728976",
      "markPx": "1584.33",
      "mgnMode": "cross",
      "mgnRatio": "11.719278420807477",
      "mmr": "1.9011959999999997",
      "notionalUsd": "237.75168928499997",
      "pos": "15",
      "posSide": "net",
      "uTime": "1697502301460",
      "upl": "-2.0115000000000123",
      "uplRatio": "-0.0839310526118142"
    }
  ]
}
```

```
],
"msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
algoid	String	Algo ID
algoCIOrdId	String	Client-supplied Algo ID. Used to be extended in the future.
instType	String	Instrument type
instId	String	Instrument ID, e.g. <code>BTC-USDT-SWAP</code>
cTime	String	Algo order created time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
uTime	String	Algo order updated time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
avgPx	String	Average open price
ccy	String	Margin currency
lever	String	Leverage
liqPx	String	Estimated liquidation price
posSide	String	Position side <code>net</code>
pos	String	Quantity of positions
mgnMode	String	Margin mode <code>cross</code> <code>isolated</code>
mgnRatio	String	Maintenance margin ratio
imr	String	Initial margin requirement
mmr	String	Maintenance margin requirement
upl	String	Unrealized profit and loss
uplRatio	String	Unrealized profit and loss ratio
last	String	Latest traded price
notionalUsd	String	Notional value of positions in <code>USD</code>
adl	String	Automatic-Deleveraging, signal area Divided into 5 levels, from 1 to 5, the smaller the number, the weaker the adl intensity.
markPx	String	Mark price

## GET / POSITION HISTORY

Retrieve the updated position data for the last 3 months. Return in reverse chronological order using `utime`.

**RATE LIMIT: 10 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

## Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
algId	String	Yes	Algo ID
instId	String	No	Instrument ID, e.g.: <code>BTC-USD-SWAP</code>
after	String	No	Pagination of data to return records earlier than the requested <code>uTime</code> , Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
before	String	No	Pagination of data to return records newer than the requested <code>uTime</code> , Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. The maximum is 100. The default is 100.

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "cTime": "1704724451471",
      "closeAvgPx": "200",
      "direction": "net",
      "instId": "ETH-USDT-SWAP",
      "lever": "5.0",
      "mgnMode": "cross",
      "openAvgPx": "220",
      "pnl": "-2.021",
      "pnlRatio": "-0.4593181818181818",
      "uTime": "1704724456322",
      "uly": "ETH-USDT"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID
mgnMode	String	Margin mode <code>cross</code> <code>isolated</code>
cTime	String	Created time of position
uTime	String	Updated time of position
openAvgPx	String	Average price of opening position
closeAvgPx	String	Average price of closing position
pnl	String	Profit and loss
pnlRatio	String	P&L ratio
lever	String	Leverage
direction	String	Direction: <code>long</code> <code>short</code>
uly	String	Underlying

## POST / CLOSE POSITION

Close the position of an instrument via a market order.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

```
POST /api/v5/tradingBot/signal/close-position
```

Request Example

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
algold	String	Yes	Algo ID
instId	String	Yes	Instrument ID

Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoId": "448965992920907776"
    }
  ],
  "msg": ""
}
```

**RESPONSE PARAMETERS**

Parameter	Type	Description
algold	String	Algo ID

## POST / PLACE SUB ORDER

You can place an order only if you have sufficient funds.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

```
POST /api/v5/tradingBot/signal/sub-order
```

Request Example

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT-SWAP</code>
algold	String	Yes	Algo ID
side	String	Yes	Order side, <code>buy</code> <code>sell</code>

Parameter	Type	Required	Description
ordType	String	Yes	Order type `market`: Market order `limit`: Limit order
sz	String	Yes	Quantity to buy or sell
px	String	Conditional	Order price. Only applicable to `limit` order.
reduceOnly	Boolean	No	Whether orders can only reduce in position size. Valid options: `true` or `false`. The default value is `false`. Only applicable to `Futures mode` / `Multi-currency margin`

### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
  ]
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, `0` means success
msg	String	The error message, empty if the code is 0
data	Array of objects	Array of objects contains the response results

#### ordType

Order type. When creating a new order, you must specify the order type. The order type you specify will affect: 1) what order parameters are required, and 2) how the matching system executes your order. The following are valid order types:

`limit`: Limit order, which requires specified sz and px.

`market`: Market order. It will be filled with market price (by swiping opposite order book). Market order will be placed to order book with most aggressive price allowed by Price Limit Mechanism.

sz refers to the number of contracts.

#### reduceOnly

When placing an order with this parameter set to true, it means that the order will reduce the size of the position only. The sum of the current order size and all reverse direction reduce-only pending orders which's price-time priority is higher than the current order, cannot exceed the contract quantity of position. Only applicable to `Futures mode` and `Multi-currency margin`

### POST / CANCEL SUB ORDER

Cancel an incomplete order.

### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: TRADE

### HTTP REQUEST

```
POST /api/v5/tradingBot/signal/cancel-sub-order
```

## Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
algold	String	Yes	Algo ID
instId	String	Yes	Instrument ID, e.g. BTC-USDT-SWAP
signalOrdId	String	Yes	Order ID

## Response Example

```
{  
  "code": "0",  
  "msg": "",  
  "data": [  
    {  
      "signalOrdId": "590908157585625111",  
      "sCode": "0",  
      "sMsg": ""  
    }  
  ]  
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, 0 means success
msg	String	The error message, empty if the code is 0
data	Array of objects	Array of objects contains the response results
> signalOrdId	String	Order ID
> sCode	String	The code of the event execution result, 0 means success.
> sMsg	String	Rejection or success message of event execution.

Cancel order returns with sCode equal to 0. It is not strictly considered that the order has been canceled. It only means that your cancellation request has been accepted by the system server. The result of the cancellation is subject to the state by get sub orders endpoint.

### GET / SIGNAL BOT SUB ORDERS

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

```
GET /api/v5/tradingBot/signal/sub-orders
```

## Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
algold	String	Yes	Algo ID
algoOrdType	String	Yes	Algo order type contract: Contract signal

Parameter	Type	Required	Description
state	String	Conditional	<p>Sub order state</p> <p><code>live</code></p> <p><code>partially_filled</code></p> <p><code>filled</code></p> <p><code>cancelled</code></p> <p>Either <code>state</code> or <code>signalOrdId</code> is required, if both are passed in, only <code>state</code> is valid.</p>
signalOrdId	String	Conditional	Sub order ID
after	String	No	Pagination of data to return records earlier than the requested <code>ordId</code>
before	String	No	Pagination of data to return records newer than the requested <code>ordId</code> .
begin	String	No	Return records of <code>ctime</code> after than the requested timestamp (include), Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
end	String	No	Return records of <code>ctime</code> before than the requested timestamp (include), Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. The maximum is 100. The default is 100.
type	String	No	<p>Sub order type</p> <p><code>live</code></p> <p><code>filled</code></p> <p>Either <code>type</code> or <code>c10rdId</code> is required, if both are passed in, only <code>c10rdId</code> is valid.</p>
c10rdId	String	No	<p>Sub order client-supplied ID.</p> <p><code>It will be deprecated soon</code></p>

### Response Example

```
{
  "code": "0",
  "data": [
    {
      "accFillSz": "18",
      "algoC10rdId": "",
      "algoId": "623833708424069120",
      "algoOrdType": "contract",
      "avgPx": "1572.81",
      "cTime": "1697024702320",
      "ccy": "",
      "c10rdId": "0632302662327996418",
      "ctVal": "0.01",
      "fee": "-0.1415529",
      "feeCcy": "USDT",
      "instId": "ETH-USDT-SWAP",
      "instType": "SWAP",
      "lever": "10",
      "ordId": "632302662351958016",
      "ordType": "market",
      "pnl": "-2.6784",
      "posSide": "net",
      "px": "",
      "side": "buy",
      "state": "filled",
      "sz": "18",
      "tag": "",
      "tdMode": "cross",
      "uTime": "1697024702322"
    }
  ],
  "msg": ""
}
```

Parameter	Type	Description
algId	String	Algo ID
algoCIOrId	String	Client-supplied Algo ID. Used to be extended in the future
instType	String	Instrument type
instId	String	Instrument ID
algoOrdType	String	Algo order type contract: Contract signal
ordId	String	Sub order ID
clOrdId	String	Sub order client-supplied ID. It is equal to signalOrdId
cTime	String	Sub order created time, Unix timestamp format in milliseconds, e.g. 1597026383085
uTime	String	Sub order updated time, Unix timestamp format in milliseconds, e.g. 1597026383085
tdMode	String	Sub order trade mode Margin mode: cross/isolated Non-Margin mode: cash
ccy	String	Margin currency Only applicable to cross MARGIN orders in Futures mode.
ordType	String	Sub order type market: Market order limit: Limit order ioc: Immediate-or-cancel order
sz	String	Sub order quantity to buy or sell
state	String	Sub order state canceled live partially_filled filled canceling
side	String	Sub order side buy, sell
px	String	Sub order price
fee	String	Sub order fee amount
feeCcy	String	Sub order fee currency
avgPx	String	Sub order average filled price
accFillSz	String	Sub order accumulated fill quantity
posSide	String	Sub order position side net
pnl	String	Sub order profit and loss
ctVal	String	Contract value Only applicable to FUTURES/SWAP

Parameter	Type	Description
lever	String	Leverage
tag	String	Order tag

#### GET / SIGNAL BOT EVENT HISTORY

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

#### HTTP REQUEST

GET /api/v5/tradingBot/signal/event-history

Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
algoid	String	Yes	Algo ID
after	String	No	Pagination of data to return records <code>eventCtime</code> earlier than the requested timestamp, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
before	String	No	Pagination of data to return records <code>eventCtime</code> newer than the requested timestamp, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. The maximum is 100. The default is 100.

Response Example

```
{
  "code": "0",
  "data": [
    {
      "alertMsg": "{\"marketPosition\":\"short\",\"prevMarketPosition\":\"long\",\"action\":\"sell\",\"instrument\":\"ETHUSDT.P\", \"timestamp\":\"2023-10-16T10:50:00.000Z\", \"maxLeverage\":2.0, \"algoId\": \"623833708424069120\", \"eventCtime\": \"1697453400959\", \"eventProcessMsg\": \"Processed reverse entry signal and placed ETH-USDT-SWAP order with all available balance\", \"eventStatus\": \"success\", \"eventType\": \"signal_processing\", \"eventUtime\": \"\", \"triggeredOrdData\": [ { \"c10rId\": \"0634100754731765763\" }, { \"c10rId\": \"0634100754752737282\" } ] }, \"msg\": \"\""
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
alertMsg	String	Alert message
algoid	String	Algo ID

Parameter	Type	Description
eventType	String	Event type <code>system_action</code> <code>user_action</code> <code>signal_processing</code>
eventCtime	String	Event timestamp of creation. Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
eventUtime	String	Event timestamp of update. Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
eventProcessMsg	String	Event process message
eventStatus	String	Event status <code>success</code> <code>failure</code>
triggeredOrdData	Array of objects	Triggered sub order data
> clOrdId	String	Sub order client-supplied id

## Recurring Buy

Recurring buy is a strategy for investing a fixed amount in crypto at fixed intervals. An appropriate recurring approach in volatile markets allows you to buy crypto at lower costs. Learn more

The API endpoints of `Recurring buy` require authentication.

**POST / PLACE RECURRING BUY ORDER**

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

`POST /api/v5/tradingBot/recurring/order-algo`

Request Example

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
stgyName	String	Yes	Custom name for trading bot, no more than 40 characters
recurringList	Array of objects	Yes	Recurring buy info
> ccy	String	Yes	Recurring currency, e.g. <code>BTC</code>
> ratio	String	Yes	Proportion of recurring currency assets, e.g. "0.2" representing 20%
period	String	Yes	Period <code>monthly</code> <code>weekly</code> <code>daily</code> <code>hourly</code>

Parameter	Type	Required	Description
recurringDay	String	Conditional	Recurring buy date When the period is <code>monthly</code> , the value range is an integer of [1,28] When the period is <code>weekly</code> , the value range is an integer of [1,7] When the period is <code>daily</code> / <code>hourly</code> , the parameter is not required.
recurringHour	String	Conditional	Recurring buy by hourly <code>1/4/8/12</code> , e.g. <code>4</code> represents "recurring buy every 4 hour" When the period is <code>hourly</code> , the parameter is required.
recurringTime	String	Yes	Recurring buy time, the value range is an integer of [0,23] When the period is <code>hourly</code> , the parameter is the time of the first investment occurs.
timeZone	String	Yes	UTC time zone, the value range is an integer of [-12,14] e.g. "8" representing UTC+8 (East 8 District), Beijing Time
amt	String	Yes	Quantity invested per cycle
investmentCcy	String	Yes	The invested quantity unit, can only be <code>USDT</code> / <code>USDC</code>
tdMode	String	Yes	Trading mode Margin mode: <code>cross</code> Non-Margin mode: <code>cash</code>
algoClOrdId	String	No	Client-supplied Algo ID There will be a value when algo order attaching algoClOrdId is triggered, or it will be "". A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
tag	String	No	Order tag A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters.
tradeQuoteCcy	String	No	The quote currency for trading.

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "algoId": "560472804207104000",
      "algoClOrdId": "",
      "sCode": "0",
      "sMsg": "",
      "tag": ""
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
algoid	String	Algo ID
algoClOrdId	String	Client-supplied Algo ID
sCode	String	The code of the event execution result, 0 means success
sMsg	String	Rejection message if the request is unsuccessful
tag	String	Order tag

## POST / AMEND RECURRING BUY ORDER

RATE LIMIT: 20 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: TRADE

HTTP REQUEST

```
POST /api/v5/tradingBot/recurring/amend-order-algo
```

Request Example

REQUEST PARAMETERS

Parameter	Type	Required	Description
algold	String	Yes	Algo ID
stgyName	String	Yes	New custom name for trading bot after adjustment, no more than 40 characters

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "algoid": "448965992920907776",
      "algoClOrdId": "",
      "sCode": "0",
      "sMsg": ""
    }
  ]
}
```

RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
algoClOrdId	String	Client-supplied Algo ID
sCode	String	The code of the event execution result, 0 means success
sMsg	String	Rejection message if the request is unsuccessful

## POST / STOP RECURRING BUY ORDER

A maximum of 10 orders can be stopped per request.

RATE LIMIT: 20 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: TRADE

HTTP REQUEST

```
POST /api/v5/tradingBot/recurring/stop-order-algo
```

Request Example

REQUEST PARAMETERS

Parameter	Type	Required	Description
algold	String	Yes	Algo ID

Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoClOrdId": "",
      "algoId": "1839309556514557952",
      "sCode": "0",
      "sMsg": "",
      "tag": ""
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
algoClOrdId	String	Client-supplied Algo ID
sCode	String	The code of the event execution result, 0 means success
sMsg	String	Rejection message if the request is unsuccessful
tag	String	Order tag (Deprecated)

## GET / RECURRING BUY ORDER LIST

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

### HTTP REQUEST

```
GET /api/v5/tradingBot/recurring/orders-algo-pending
```

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
algold	String	No	Algo ID
after	String	No	Pagination of data to return records earlier than the requested <code>algoId</code> .
before	String	No	Pagination of data to return records newer than the requested <code>algoId</code> .
limit	String	No	Number of results per request. The maximum is 100. The default is 100

Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoClOrdId": "",
      "algoId": "644497312047435776",
      "algoOrdType": "recurring",
      "amt": "100",
      "cTime": "1699932133373",
      "cycles": "6",
      "instType": "SPOT",
      "investmentAmt": "0",
      "investmentCcy": "USDC",
      "mktCap": "0",
      "orderType": "RECURRING"
    }
  ],
  "msg": ""
}
```

```

"period": "hourly",
"pnlRatio": "0",
"recurringDay": "",
"recurringHour": "1",
"recurringList": [
  {
    "ccy": "BTC",
    "ratio": "0.2"
  },
  {
    "ccy": "ETH",
    "ratio": "0.8"
  }
],
"recurringTime": "12",
"state": "running",
"stgyName": "stg1",
"tag": "",
"timeZone": "8",
"totalAnnRate": "0",
"totalPnl": "0",
"uTime": "1699952473152",
"tradeQuoteCcy": "USDT"
},
],
"msg": ""
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
algoCIOrdId	String	Client-supplied Algo ID
instType	String	Instrument type
cTime	String	Algo order created time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
uTime	String	Algo order updated time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
algoOrdType	String	Algo order type <code>recurring</code> : recurring buy
state	String	Algo order state <code>running</code> <code>stopping</code> <code>pause</code>
stgyName	String	Custom name for trading bot, no more than 40 characters
recurringList	Array of objects	Recurring buy info
> ccy	String	Recurring currency, e.g. <code>BTC</code>
> ratio	String	Proportion of recurring currency assets, e.g. "0.2" representing 20%
period	String	Period <code>monthly</code> <code>weekly</code> <code>daily</code> <code>hourly</code>
recurringDay	String	Recurring buy date When the period is <code>monthly</code> , the value range is an integer of [1,28] When the period is <code>weekly</code> , the value range is an integer of [1,7]

Parameter	Type	Description
recurringHour	String	Recurring buy by hourly 1/4/8/12, e.g. 4 represents "recurring buy every 4 hour"
recurringTime	String	Recurring buy time, the value range is an integer of [0,23]
timeZone	String	UTC time zone, the value range is an integer of [-12,14] e.g. "8" representing UTC+8 (East 8 District), Beijing Time
amt	String	Quantity invested per cycle
investmentAmt	String	Accumulate quantity invested
investmentCcy	String	The invested quantity unit, can only be USDT/USDC
totalPnl	String	Total P&L
totalAnnRate	String	Total annualized rate of yield
pnlRatio	String	Rate of yield
mktCap	String	Market value in unit of USDT
cycles	String	Accumulate recurring buy cycles
tag	String	Order tag
tradeQuoteCcy	String	The quote currency for trading.

#### GET / RECURRING BUY ORDER HISTORY

RATE LIMIT: 20 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: READ

#### HTTP REQUEST

GET /api/v5/tradingBot/recurring/orders-algo-history

Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
algoid	String	No	Algo ID
after	String	No	Pagination of data to return records earlier than the requested algoId.
before	String	No	Pagination of data to return records newer than the requested algoId.
limit	String	No	Number of results per request. The maximum is 100. The default is 100

Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoClOrdId": "",
      "algoId": "644496098429767680",
      "algoOrdType": "recurring",
      "amt": "100",
      "cTime": "1699931844050",
      "cycles": "0",
      "cTime": "1699931844050",
      "cTime": "1699931844050"
    }
  ]
}
```

```

"instType": "SPOT",
"investmentAmt": "0",
"investmentCcy": "USDC",
"mktCap": "0",
"period": "hourly",
"pnlRatio": "0",
"recurringDay": "",
"recurringHour": "1",
"recurringList": [
  {
    "ccy": "BTC",
    "ratio": "0.2"
  },
  {
    "ccy": "ETH",
    "ratio": "0.8"
  }
],
"recurringTime": "0",
"state": "stopped",
"stgyName": "stg1",
"tag": "",
"timeZone": "8",
"totalAnnRate": "0",
"totalPnl": "0",
"uTime": "1699932177659",
"tradeQuoteCcy": "USDT"
},
],
"msg": ""
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
algoCIOrdId	String	Client-supplied Algo ID
instType	String	Instrument type
cTime	String	Algo order created time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
uTime	String	Algo order updated time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
algoOrdType	String	Algo order type <code>recurring</code> : recurring buy
state	String	Algo order state <code>stopped</code>
stgyName	String	Custom name for trading bot, no more than 40 characters
recurringList	Array of objects	Recurring buy info
> ccy	String	Recurring currency, e.g. <code>BTC</code>
> ratio	String	Proportion of recurring currency assets, e.g. "0.2" representing 20%
period	String	Period <code>monthly</code> <code>weekly</code> <code>daily</code> <code>hourly</code>
recurringDay	String	Recurring buy date When the period is <code>monthly</code> , the value range is an integer of [1,28] When the period is <code>weekly</code> , the value range is an integer of [1,7]

Parameter	Type	Description
recurringHour	String	Recurring buy by hourly 1/4/8/12, e.g. 4 represents "recurring buy every 4 hour"
recurringTime	String	Recurring buy time, the value range is an integer of [0,23]
timeZone	String	UTC time zone, the value range is an integer of [-12,14] e.g. "8" representing UTC+8 (East 8 District), Beijing Time
amt	String	Quantity invested per cycle
investmentAmt	String	Accumulate quantity invested
investmentCcy	String	The invested quantity unit, can only be USDT/USDC
totalPnl	String	Total P&L
totalAnnRate	String	Total annualized rate of yield
pnlRatio	String	Rate of yield
mktCap	String	Market value in unit of USDT
cycles	String	Accumulate recurring buy cycles
tag	String	Order tag
tradeQuoteCcy	String	The quote currency for trading.

#### GET / RECURRING BUY ORDER DETAILS

RATE LIMIT: 20 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: READ

#### HTTP REQUEST

GET /api/v5/tradingBot/recurring/orders-algo-details

Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
algoid	String	Yes	Algo ID

Response Example

```
{
  "code": "0",
  "data": [
    {
      "algoClOrdId": "",
      "algoId": "644497312047435776",
      "algoOrdType": "recurring",
      "amt": "100",
      "cTime": "1699932133373",
      "cycles": "6",
      "instType": "SPOT",
      "investmentAmt": "0",
      "investmentCcy": "USDC",
      "mktCap": "0",
      "nextInvestTime": "1699956005500",
      "period": "hourly",
      "pnlRatio": "0",
      "recurringDay": ""
    }
  ]
}
```

```

    "recurringHour": "1",
    "recurringList": [
        {
            "avgPx": "0",
            "ccy": "BTC",
            "profit": "0",
            "px": "36683.2",
            "ratio": "0.2",
            "totalAmt": "0"
        },
        {
            "avgPx": "0",
            "ccy": "ETH",
            "profit": "0",
            "px": "2058.36",
            "ratio": "0.8",
            "totalAmt": "0"
        }
    ],
    "recurringTime": "12",
    "state": "running",
    "stgyName": "stg1",
    "tag": "",
    "timeZone": "8",
    "totalAnnRate": "0",
    "totalPnl": "0",
    "uTime": "1699952485451",
    "tradeQuoteCcy": "USDT"
}
],
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
algold	String	Algo ID
algoClOrdId	String	Client-supplied Algo ID
instType	String	Instrument type
cTime	String	Algo order created time, Unix timestamp format in milliseconds, e.g. 1597026383085
uTime	String	Algo order updated time, Unix timestamp format in milliseconds, e.g. 1597026383085
algoOrdType	String	Algo order type recurring: recurring buy
state	String	Algo order state running stopping stopped pause
stgyName	String	Custom name for trading bot, no more than 40 characters
recurringList	Array of objects	Recurring buy info
> ccy	String	Recurring buy currency, e.g. BTC
> ratio	String	Proportion of recurring currency assets, e.g. "0.2" representing 20%
> totalAmt	String	Accumulated quantity in unit of recurring buy currency
> profit	String	Profit in unit of investmentCcy
> avgPx	String	Average price of recurring buy, quote currency is investmentCcy

Parameter	Type	Description
> px	String	Current market price, quote currency is <code>investmentCcy</code>
period	String	Period <input type="radio"/> monthly <input type="radio"/> weekly <input type="radio"/> daily <input type="radio"/> hourly
recurringDay	String	Recurring buy date When the period is <code>monthly</code> , the value range is an integer of [1,28] When the period is <code>weekly</code> , the value range is an integer of [1,7]
recurringHour	String	Recurring buy by hourly <input type="radio"/> 1/ <input type="radio"/> 4/ <input type="radio"/> 8/ <input type="radio"/> 12, e.g. <code>4</code> represents "recurring buy every 4 hour"
recurringTime	String	Recurring buy time, the value range is an integer of [0,23]
timeZone	String	UTC time zone, the value range is an integer of [-12,14] e.g. "8" representing UTC+8 (East 8 District), Beijing Time
amt	String	Quantity invested per cycle
investmentAmt	String	Accumulate quantity invested
investmentCcy	String	The invested quantity unit, can only be <code>USDT</code> / <code>USDC</code>
nextInvestTime	String	Next invest time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
totalPnl	String	Total P&L
totalAnnRate	String	Total annualized rate of yield
pnlRatio	String	Rate of yield
mktCap	String	Market value in unit of <code>USDT</code>
cycles	String	Accumulate recurring buy cycles
tag	String	Order tag
tradeQuoteCcy	String	The quote currency for trading.

#### GET / RECURRING BUY SUB ORDERS

RATE LIMIT: 20 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: READ

#### HTTP REQUEST

`GET /api/v5/tradingBot/recurring/sub-orders`

Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
algoid	String	Yes	Algo ID
ordId	String	No	Sub order ID
after	String	No	Pagination of data to return records earlier than the requested <code>algoid</code> .

Parameter	Type	Required	Description
before	String	No	Pagination of data to return records newer than the requested <code>algoId</code> .
limit	String	No	Number of results per request. The maximum is 100. The default is 100

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "accFillSz": "0.045315",
      "algoClOrdId": "",
      "algoId": "560516615079727104",
      "algoOrdType": "recurring",
      "avgPx": "1765.4",
      "cTime": "1679911222200",
      "fee": "-0.000317205",
      "feeCcy": "ETH",
      "instId": "ETH-USDC",
      "instType": "SPOT",
      "ordId": "560523524230717440",
      "ordType": "market",
      "px": "-1",
      "side": "buy",
      "state": "filled",
      "sz": "80",
      "tag": "",
      "tdMode": "",
      "uTime": "1679911222207"
    },
    {
      "accFillSz": "0.00071526",
      "algoClOrdId": "",
      "algoId": "560516615079727104",
      "algoOrdType": "recurring",
      "avgPx": "27961.6",
      "cTime": "1679911222189",
      "fee": "-0.00000500682",
      "feeCcy": "BTC",
      "instId": "BTC-USDC",
      "instType": "SPOT",
      "ordId": "560523524184580096",
      "ordType": "market",
      "px": "-1",
      "side": "buy",
      "state": "filled",
      "sz": "20",
      "tag": "",
      "tdMode": "",
      "uTime": "1679911222194"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
algoid	String	Algo ID
instType	String	Instrument type
instId	String	Instrument ID
algoOrdType	String	Algo order type <code>recurring</code> : recurring buy
ordId	String	Sub order ID

Parameter	Type	Description
cTime	String	Sub order created time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
uTime	String	Sub order updated time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
tdMode	String	Sub order trade mode Margin mode : <code>cross</code> Non-Margin mode : <code>cash</code>
ordType	String	Sub order type <code>market</code> : Market order
sz	String	Sub order quantity to buy or sell
state	String	Sub order state <code>canceled</code> <code>live</code> <code>partially_filled</code> <code>filled</code> <code>canceling</code>
side	String	Sub order side <code>buy</code> <code>sell</code>
px	String	Sub order limit price If it's a market order, "-1" will be return
fee	String	Sub order fee
feeCcy	String	Sub order fee currency
avgPx	String	Sub order average filled price
accFillSz	String	Sub order accumulated fill quantity
tag	String	Order tag
algoClOrdId	String	Client-supplied Algo ID

#### WS / RECURRING BUY ORDERS CHANNEL

Retrieve recurring buy orders. Data will be pushed when triggered by events. It will also be pushed in regular interval according to subscription granularity.

#### URL PATH

/ws/v5/business (required login)

#### Request Example

```

import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/business",
        useServerTime=False
    )

```

```

await ws.start()
args = [{
    "channel": "algo-recurring-buy",
    "instType": "SPOT"
}]

await ws.subscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

await ws.unsubscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>algo-recurring-buy</code>
> instType	String	Yes	Instrument type <code>SPOT</code> <code>ANY</code>
> algold	String	No	Algo Order ID

## Successful Response Example

```
{
    "id": "1512",
    "event": "subscribe",
    "arg": {
        "channel": "algo-recurring-buy",
        "instType": "SPOT"
    },
    "connId": "a4d3ae55"
}
```

## Failure Response Example

```
{
    "id": "1512",
    "event": "error",
    "code": "60012",
    "msg": "Invalid request: {\\"op\\": \"subscribe\", \\"args\\": [{ \\"channel\\\" : \"algo-recurring-buy\", \\"instType\\\" : \"FUTURES\"}]}",
    "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message

Parameter	Type	Required	Description
event	String	Yes	Event   
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> instType	String	Yes	Instrument type
> algold	String	No	Algo Order ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

Push Data Example:

```
{
  "arg": {
    "channel": "algo-recurring-buy",
    "instType": "SPOT",
    "uid": "447*****584"
  },
  "data": [
    {
      "algoClOrdId": "",
      "algoId": "644497312047435776",
      "algoOrdType": "recurring",
      "amt": "100",
      "cTime": "1699932133373",
      "cycles": "0",
      "instType": "SPOT",
      "investmentAmt": "0",
      "investmentCcy": "USDC",
      "mktCap": "0",
      "nextInvestTime": "1699934415300",
      "pTime": "169993314691",
      "period": "hourly",
      "pnlRatio": "0",
      "recurringDay": "",
      "recurringHour": "1",
      "recurringList": [
        {
          "avgPx": "0",
          "ccy": "BTC",
          "profit": "0",
          "px": "36482",
          "ratio": "0.2",
          "totalAmt": "0"
        },
        {
          "avgPx": "0",
          "ccy": "ETH",
          "profit": "0",
          "px": "2057.54",
          "ratio": "0.8",
          "totalAmt": "0"
        }
      ],
      "recurringTime": "12",
      "state": "running",
      "stgyName": "stg1",
      "tag": "",
      "timeZone": "8",
      "totalAnnRate": "0",
      "totalPnl": "0",
      "uTime": "1699932136249",
      "tradeQuoteCcy": "USDT"
    }
  ]
}
```

```
]}  
}
```

#### RESPONSE PARAMETERS WHEN DATA IS PUSHED.

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> instType	String	Instrument type
> algold	String	Algo Order ID
> uid	String	User ID
data	Array of objects	Subscribed data
> algold	String	Algo ID
> algoClOrdId	String	Client-supplied Algo ID
> instType	String	Instrument type
> cTime	String	Algo order created time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> uTime	String	Algo order updated time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> algoOrdType	String	Algo order type <code>recurring</code> : recurring buy
> state	String	Algo order state <code>running</code> <code>stopping</code> <code>stopped</code> <code>pause</code>
> stgyName	String	Custom name for trading bot, no more than 40 characters
> recurringList	Array of objects	Recurring buy info
>> ccy	String	Recurring buy currency, e.g. <code>BTC</code>
>> ratio	String	Proportion of recurring currency assets, e.g. "0.2" representing 20%
>> totalAmt	String	Accumulated quantity in unit of recurring buy currency
>> profit	String	Profit in unit of <code>investmentCcy</code>
>> avgPx	String	Average price of recurring buy, quote currency is <code>investmentCcy</code>
>> px	String	Current market price, quote currency is <code>investmentCcy</code>
> period	String	Period <code>monthly</code> <code>weekly</code> <code>daily</code> <code>hourly</code>
> recurringDay	String	Recurring buy date When the period is <code>monthly</code> , the value range is an integer of [1,28] When the period is <code>weekly</code> , the value range is an integer of [1,7]

Parameter	Type	Description
> recurringHour	String	Recurring buy by hourly 1/4/8/12, e.g. 4 represents "recurring buy every 4 hour"
> recurringTime	String	Recurring buy time, the value range is an integer of [0,23]
> timeZone	String	UTC time zone, the value range is an integer of [-12,14] e.g. "8" representing UTC+8 (East 8 District), Beijing Time
> amt	String	Quantity invested per cycle
> investmentAmt	String	Accumulate quantity invested
> investmentCcy	String	The invested quantity unit, can only be USDT/USDC
> nextInvestTime	String	Next invest time, Unix timestamp format in milliseconds, e.g. 1597026383085
> totalPnl	String	Total P&L
> totalAnnRate	String	Total annualized rate of yield
> pnlRatio	String	Rate of yield
> mktCap	String	Market value in unit of USDT
> cycles	String	Accumulate recurring buy cycles
> tag	String	Order tag
> pTime	String	Push time of algo order information, Unix timestamp format in milliseconds, e.g. 1597026383085
> tradeQuoteCcy	String	The quote currency for trading.

## Copy Trading

Lead trading API Workflow as follows:

### 1. Apply to become a leading trader.

- The procedure can refer to How to become a lead trader;
- You can know whether you are a lead trader by checking whether `roleType` or `spotRoleType` from Get account configuration is 1.

### 2. Leading instruments:

- GET / Leading instruments can get instruments that are supported to have leading trades and the instruments that you enable leading trade. For instruments that are disabled copy trading, you can still trade normally, but copy trading will not be triggered;
- Amend leading instruments can amend your leading instruments. You need to set initial leading instruments while applying to become a leading trader. All non-leading contracts can't have position or pending orders for the current request when setting non-leading contracts as leading contracts.

### 3. Open position:

- You can open the position by placing order endpoints and channels including Place order endpoint, Place multiple orders endpoint, Place order channel, Place multiple orders channel, `tdMode` should be `spot_isolated` for `SPOT` lead trading.
- For buy/sell mode, the orders must be in the same direction as your existing positions and open orders. You can select the direction you want if the instrument does not have position and pending orders.
- For long/short mode, you can open long or open short as you want.

### 4. Close position

- You can close the position with customized price or size by placing order endpoints and channels including Place order endpoint, Place multiple orders endpoint, Place order channel, Place multiple orders channel, or close the position by Close positions / Close lead position;

- Close positions can close certain position under the current instrument(e.g. the long or short position under long/shor mode ), which can contain multiple leading positions;
- Close lead position can only close a leading position once a time. It is required to pass subPosId which can get from Get existing leading positions.

## 5. TP/SL

- TP/SL can be set by Place algo order or Place lead stop order;
- Place algo order can set TP/SL for certain position under the current instrument(e.g. the long or short position under long/shor mode ), which can contain multiple leading positions;
- Place lead stop order set set TP/SL for only a leading position once a time. It is required to pass subPosId which can get from Get existing leading positions.

### GET / EXISTING LEAD POSITIONS

Retrieve lead positions that are not closed.

Returns reverse chronological order with `openTime`

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/copytrading/current-subpositions`

Request example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	<p>Instrument type</p> <p><code>SPOT</code></p> <p><code>SWAP</code></p> <p>It returns all types by default.</p>
instId	String	No	Instrument ID, e.g. BTC-USDT-SWAP
after	String	No	Pagination of data to return records earlier than the requested <code>subPosId</code> .
before	String	No	Pagination of data to return records newer than the requested <code>subPosId</code> .
limit	String	No	Number of results per request. Maximum is 500. Default is 500.

Response example

```
{
  "code": "0",
  "data": [
    {
      "algoId": "",
      "ccy": "USDT",
      "instId": "BTC-USDT-SWAP",
      "instType": "SWAP",
      "lever": "3",
      "margin": "12.6417",
      "markPx": "38205.8",
      "mgnMode": "isolated",
      "openAvgPx": "37925.1",
      "openOrdId": "",
      "openTime": "1701231120479",
      "posSide": "net",
      "sl0OrdPx": "",
      "slTriggerPx": "",
      "subPos": "1",
      "subPosId": "649945658862370816",
      "tp0OrdPx": "",
      "tpTriggerPx": "",
      "uniqueCode": "25CD5A80241D6FE6",
    }
  ]
}
```

```

    "upl": "0.2807",
    "uplRatio": "0.0222042921442527",
    "availSubPos": "1"
},
{
    "algId": "",
    "ccy": "USDT",
    "instId": "BTC-USDT-SWAP",
    "instType": "SWAP",
    "lever": "3",
    "margin": "12.6263333333333333",
    "markPx": "38205.8",
    "mgnMode": "isolated",
    "openAvgPx": "37879",
    "openOrdId": "",
    "openTime": "1701225074786",
    "posSide": "net",
    "slOrdPx": "",
    "slTriggerPx": "",
    "subPos": "1",
    "subPosId": "649920301388038144",
    "tpOrdPx": "",
    "tpTriggerPx": "",
    "uniqueCode": "25CD5A80241D6FE6",
    "upl": "0.3268",
    "uplRatio": "0.0258824150584758",
    "availSubPos": "1"
}
],
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID, e.g. BTC-USDT-SWAP
subPosId	String	Lead position ID
posSide	String	Position side <input type="radio"/> long <input type="radio"/> short <input type="radio"/> net (Long positions have positive subPos; short positions have negative subPos)
mgnMode	String	Margin mode. <input type="radio"/> cross <input type="radio"/> isolated
lever	String	Leverage
openOrdId	String	Order ID for opening position, only applicable to lead position
openAvgPx	String	Average open price
openTime	String	Open time
subPos	String	Quantity of positions
tpTriggerPx	String	Take-profit trigger price.
slTriggerPx	String	Stop-loss trigger price.
algId	String	Stop order ID
instType	String	Instrument type
tpOrdPx	String	Take-profit order price, it is -1 for market price
slOrdPx	String	Stop-loss order price, it is -1 for market price

Parameter	Type	Description
margin	String	Margin
upl	String	Unrealized profit and loss
uplRatio	String	Unrealized profit and loss ratio
markPx	String	Latest mark price, only applicable to contract
uniqueCode	String	Lead trader unique code
ccy	String	Margin currency
availSubPos	String	Quantity of positions that can be closed

#### GET / LEAD POSITION HISTORY

Retrieve the completed lead position of the last 3 months.

Returns reverse chronological order with `subPosId`.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/copytrading/subpositions-history`

Request example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type <input type="checkbox"/> SPOT <input type="checkbox"/> SWAP It returns all types by default.
instId	String	No	Instrument ID, e.g. BTC-USDT-SWAP
after	String	No	Pagination of data to return records earlier than the requested <code>subPosId</code> .
before	String	No	Pagination of data to return records newer than the requested <code>subPosId</code> .
limit	String	No	Number of results per request. Maximum is 100. Default is 100.

Response example

```
{
  "code": "0",
  "data": [
    {
      "ccy": "USDT",
      "closeAvgPx": "37617.5",
      "closeTime": "1701188587950",
      "instId": "BTC-USDT-SWAP",
      "instType": "SWAP",
      "lever": "3",
      "margin": "37.41",
      "markPx": "38203.4",
      "mgnMode": "isolated",
      "openAvgPx": "37410",
      "openOrId": "",
      "openTime": "1701184638702",
      "pnl": "0.6225",
      "pnlRatio": "0.0166399358460306",
      "subPosId": "1701184638702"
    }
  ]
}
```

```

"posSide": "net",
"profitSharingAmt": "0.0407967",
"subPos": "3",
"closeSubPos": "2",
"type": "1",
"subPosId": "649750700213698561",
"uniqueCode": "25CD5A80241D6FE6"
},
{
  "ccy": "USDT",
  "closeAvgPx": "37617.5",
  "closeTime": "1701188587950",
  "instId": "BTC-USDT-SWAP",
  "instType": "SWAP",
  "lever": "3",
  "margin": "24.94",
  "markPx": "38203.4",
  "mgnMode": "isolated",
  "openAvgPx": "37410",
  "openOrdId": "",
  "openTime": "1701184635381",
  "pnl": "0.415",
  "pnlRatio": "0.0166399358460306",
  "posSide": "net",
  "profitSharingAmt": "0.0271978",
  "subPos": "2",
  "closeSubPos": "2",
  "type": "2",
  "subPosId": "649750686292803585",
  "uniqueCode": "25CDSA80241D6FE6"
}
],
"msg": ""
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID, e.g. BTC-USDT-SWAP
subPosId	String	Lead position ID
posSide	String	Position side    (long position has positive subPos; short position has negative subPos)
mgnMode	String	Margin mode.  
lever	String	Leverage
openOrdId	String	Order ID for opening position, only applicable to lead position
openAvgPx	String	Average open price
openTime	String	Time of opening
subPos	String	Quantity of positions
closeTime	String	Time of closing position
closeAvgPx	String	Average price of closing position
pnl	String	Profit and loss
pnlRatio	String	P&L ratio
instType	String	Instrument type

Parameter	Type	Description
margin	String	Margin
ccy	String	Currency
markPx	String	Latest mark price, only applicable to contract
uniqueCode	String	Lead trader unique code
profitSharingAmt	String	Profit sharing amount, only applicable to copy trading. Note: this parameter is already deprecated.
closeSubPos	String	Quantity of positions that is already closed
type	String	The type of closing position ①: Close position partially; ②: Close all

#### POST / PLACE LEAD STOP ORDER

Set TP/SL for the current lead position that are not closed.

#### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: USER ID

#### PERMISSION: TRADE

#### HTTP REQUEST

`POST /api/v5/copytrading/algo-order`

Request example

```
POST /api/v5/copytrading/algo-order
body
{
  "subPosId": "518541406042591232",
  "tpTriggerPx": "10000"
}
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type <a href="#">SPOT</a> <a href="#">SWAP</a> , the default value
subPosId	String	Yes	Lead position ID
tpTriggerPx	String	Conditional	Take-profit trigger price. Take-profit order price will be the market price after triggering. At least one of tpTriggerPx and slTriggerPx must be filled The take profit order will be deleted if it is 0
slTriggerPx	String	Conditional	Stop-loss trigger price. Stop-loss order price will be the market price after triggering. The stop loss order will be deleted if it is 0
tpOrdPx	String	No	Take-profit order price If the price is -1, take-profit will be executed at the market price, the default is <a href="#">-1</a> Only applicable to <a href="#">SPOT</a> lead trader
slOrdPx	String	No	Stop-loss order price If the price is -1, stop-loss will be executed at the market price, the default is <a href="#">-1</a> Only applicable to <a href="#">SPOT</a> lead trader
tpTriggerPxType	String	No	Take-profit trigger price type

Parameter	Type	Required	Description
			<p><code>last</code>: last price  <code>index</code>: index price  <code>mark</code>: mark price  Default is <code>last</code></p>
slTriggerPxType	String	No	<p>Stop-loss trigger price type  <code>last</code>: last price  <code>index</code>: index price  <code>mark</code>: mark price  Default is <code>last</code></p>
tag	String	No	<p>Order tag  A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters.</p>

#### Response example

```
{
  "code": "0",
  "data": [
    {
      "subPosId": "518560559046594560",
      "tag": ""
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
subPosId	String	Lead position ID
tag	String	Order tag

#### POST / CLOSE LEAD POSITION

You can only close a lead position once a time.

It is required to pass subPosId which can get from Get existing leading positions.

#### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: USER ID

#### PERMISSION: TRADE

#### HTTP REQUEST

```
POST /api/v5/copytrading/close-subposition
```

#### Request example

```
POST /api/v5/copytrading/close-subposition
body
{
  "subPosId": "518541406042591232",
}
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	<p>Instrument type  <code>SPOT</code>  <code>SWAP</code>, the default value</p>

Parameter	Type	Required	Description
subPosId	String	Yes	Lead position ID
ordType	String	No	Order type <input checked="" type="checkbox"/> <b>market</b> : Market order, the default value <input type="checkbox"/> <b>limit</b> : Limit order
px	String	No	Order price. Only applicable to <b>limit</b> order and <b>SPOT</b> lead trader If the price is 0, the pending order will be canceled. It is modifying order if you set <b>px</b> after placing limit order.
tag	String	No	Order tag A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters.

#### Response example

```
{
  "code": "0",
  "data": [
    {
      "subPosId": "518560559046594560",
      "tag": ""
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
subPosId	String	Lead position ID
tag	String	Order tag

#### GET / LEADING INSTRUMENTS

Retrieve instruments that are supported to lead by the platform. Retrieve instruments that the lead trader has set.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

#### HTTP REQUEST

```
GET /api/v5/copytrading/instruments
```

#### Request example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type <input checked="" type="checkbox"/> <b>SPOT</b> <input type="checkbox"/> <b>SWAP</b> , the default value

#### Response example

```
{
  "code": "0",
  "data": [
    {
      "enabled": true,
      "instId": "BTC-USDT-SWAP"
    },
  ],
}
```

```
{
  "enabled": true,
  "instId": "ETH-USDT-SWAP"
},
{
  "enabled": false,
  "instId": "ADA-USDT-SWAP"
}
],
"msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID, e.g. BTC-USDT-SWAP
enabled	Boolean	Whether instrument is a lead instrument. <code>true</code> or <code>false</code>

## POST / AMEND LEADING INSTRUMENTS

The leading trader can amend current leading instruments, need to set initial leading instruments while applying to become a leading trader. All non-leading instruments can't have position or pending orders for the current request when setting non-leading instruments as leading instruments.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

`POST /api/v5/copytrading/set-instruments`

Request example

```
POST /api/v5/copytrading/set-instruments
body
{
  "instId": "BTC-USDT-SWAP,ETH-USDT-SWAP"
}
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type <code>SPOT</code> <code>SWAP</code> , the default value
instId	String	Yes	Instrument ID, e.g. BTC-USDT-SWAP. If there are multiple instruments, separate them with commas.

The value of `instId` must include all instruments that you are going to have the lead trading with because the previous settings will be overwritten after the current request is set successfully

Response example

```
{
  "code": "0",
  "data": [
    {
      "enabled": true,
      "instId": "BTC-USDT-SWAP"
    },
    {
      "enabled": false,
      "instId": "ADA-USDT-SWAP"
    }
  ]
}
```

```

        "enabled": true,
        "instId": "ETH-USDT-SWAP"
    },
],
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID, e.g. BTC-USDT-SWAP
enabled	Boolean	Whether you set it successfully true or false

## GET / PROFIT SHARING DETAILS

The leading trader gets profits shared details for the last 3 months.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

### HTTP REQUEST

```
GET /api/v5/copytrading/profit-sharing-details
```

Request example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type SPOT SWAP It returns all types by default.
after	String	No	Pagination of data to return records earlier than the requested profitSharingId
before	String	No	Pagination of data to return records newer than the requested profitSharingId
limit	String	No	Number of results per request. Maximum is 100. Default is 100.

Response example

```

{
    "code": "0",
    "data": [
        {
            "ccy": "USDT",
            "nickName": "Potato",
            "profitSharingAmt": "0.00536",
            "profitSharingId": "148",
            "portLink": "",
            "ts": "172339200000",
            "instType": "SWAP"
        },
        {
            "ccy": "USDT",
            "nickName": "Apple",
            "profitSharingAmt": "0.00336",
            "profitSharingId": "20",
            "portLink": "",
            "ts": "172339200000",
            "instType": "SWAP"
        }
    ],
}

```

```
"msg": ""  
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	The currency of profit sharing.
profitSharingAmt	String	Profit sharing amount. It would be 0 if there is no any profit sharing.
nickName	String	Nickname of copy trader.
profitSharingId	String	Profit sharing ID.
instType	String	Instrument type
portLink	String	Portrait link
ts	String	Profit sharing time.

## GET / TOTAL PROFIT SHARING

The leading trader gets the total amount of profit shared since joining the platform.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

```
GET /api/v5/copytrading/total-profit-sharing
```

Request example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type <input type="checkbox"/> SPOT <input type="checkbox"/> SWAP It returns all types by default.

Response example

```
{  
  "code": "0",  
  "data": [  
    {  
      "ccy": "USDT",  
      "totalProfitSharingAmt": "0.6584928",  
      "instType": "SWAP"  
    }  
  ],  
  "msg": ""  
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	The currency of profit sharing.
totalProfitSharingAmt	String	Total profit sharing amount.
instType	String	Instrument type

## GET / UNREALIZED PROFIT SHARING DETAILS

The leading trader gets the profit sharing details that are expected to be shared in the next settlement cycle.

The unrealized profit sharing details will update once there copy position is closed.

### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: READ

### HTTP REQUEST

```
GET /api/v5/copytrading/unrealized-profit-sharing-details
```

Request example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type <input type="checkbox"/> SPOT <input type="checkbox"/> SWAP It returns all types by default.

Response example

```
{
  "code": "0",
  "data": [
    {
      "ccy": "USDT",
      "nickName": "Potato",
      "portLink": "",
      "ts": "1669901824779",
      "unrealizedProfitSharingAmt": "0.455472",
      "instType": "SWAP"
    },
    {
      "ccy": "USDT",
      "nickName": "Apple",
      "portLink": "",
      "ts": "1669460210113",
      "unrealizedProfitSharingAmt": "0.033608",
      "instType": "SWAP"
    }
  ],
  "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	The currency of profit sharing. e.g. USDT
unrealizedProfitSharingAmt	String	Unrealized profit sharing amount.
nickName	String	Nickname of copy trader.
instType	String	Instrument type
portLink	String	Portrait link
ts	String	Update time.

## GET / TOTAL UNREALIZED PROFIT SHARING

The leading trader gets the total unrealized amount of profit shared.

### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

**RATE LIMIT RULE: USER ID****PERMISSION: READ****HTTP REQUEST**`GET /api/v5/copytrading/total-unrealized-profit-sharing`

Request example

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
instType	String	No	Instrument type SWAP, the default value.

Response example

```
{
  "code": "0",
  "data": [
    {
      "profitSharingTs": "1705852800000",
      "totalUnrealizedProfitSharingAmt": "0.114402985553185"
    }
  ],
  "msg": ""
}
```

**RESPONSE PARAMETERS**

Parameter	Type	Description
profitSharingTs	String	The settlement time for the total unrealized profit sharing amount. Unix timestamp format in milliseconds, e.g.1597026383085
totalUnrealizedProfitSharingAmt	String	Total unrealized profit sharing amount

**POST / AMEND PROFIT SHARING RATIO**

It is used to amend profit sharing ratio.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS****RATE LIMIT RULE: USER ID****PERMISSION: TRADE****HTTP REQUEST**`POST /api/v5/copytrading/amend-profit-sharing-ratio`

Request example

```
POST /api/v5/copytrading/amend-profit-sharing-ratio
body
{
  "instType": "SWAP",
  "profitSharingRatio": "0.1"
}
```

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
instType	String	No	Instrument type SWAP

Parameter	Type	Required	Description
profitSharingRatio	String	Yes	Profit sharing ratio. 0.1 represents 10%

#### Response example

```
{
  "code": "0",
  "data": [
    {
      "result": true
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
result	Boolean	The result of setting true

#### GET / ACCOUNT CONFIGURATION

Retrieve current account configuration related to copy/lead trading.

#### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: USER ID

#### PERMISSION: READ

#### HTTP REQUEST

```
GET /api/v5/copytrading/config
```

#### Request example

#### REQUEST PARAMETERS

None

#### Response example

```
{
  "code": "0",
  "data": [
    {
      "details": [
        {
          "copyTraderNum": "1",
          "instType": "SWAP",
          "maxCopyTraderNum": "100",
          "profitSharingRatio": "0",
          "roleType": "1"
        },
        {
          "copyTraderNum": "",
          "instType": "SPOT",
          "maxCopyTraderNum": "",
          "profitSharingRatio": "",
          "roleType": "0"
        }
      ],
      "nickName": "155***9957",
      "portLink": "",
      "uniqueCode": "5506D3681454A304"
    }
  ],
  "msg": ""
}
```

}

## RESPONSE PARAMETERS

Parameter	Type	Description
uniqueCode	String	User unique code
nickName	String	Nickname
portLink	String	Portrait link
details	Array of objects	Details
> instType	String	Instrument type <input type="radio"/> SPOT <input type="radio"/> SWAP
> roleType	String	Role type <input type="radio"/> 0: General user <input type="radio"/> 1: Leading trader <input type="radio"/> 2: Copy trader
> profitSharingRatio	String	Profit sharing ratio. Only applicable to lead trader, or it will be "". 0.1 represents 10%
> maxCopyTraderNum	String	Maximum number of copy traders
> copyTraderNum	String	Current number of copy traders

## POST / FIRST COPY SETTINGS

The first copy settings for the certain lead trader. You need to first copy settings after stopping copying.

### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: TRADE

### HTTP REQUEST

POST /api/v5/copytrading/first-copy-settings

Request example

```
POST /api/v5/copytrading/first-copy-settings
body
{
  "instType": "SWAP",
  "uniqueCode": "25CD5A80241D6FE6",
  "copyMgnMode": "cross",
  "copyInstIdType": "copy",
  "copyMode": "ratio_copy",
  "copyRatio": "1",
  "copyTotalAmt": "500",
  "subPosCloseType": "copy_close"
}
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type <input type="radio"/> SWAP, the default value
uniqueCode	String	Yes	Lead trader unique code A combination of case-sensitive alphanumerics, all numbers and the length is 16 characters, e.g.

Parameter	Type	Required	Description
			213E8C92DC61EFAC
copyMgnMode	String	Yes	Copy margin mode <code>cross</code> : cross <code>isolated</code> : isolated <code>copy</code> : Use the same margin mode as lead trader when opening positions
copyInstIdType	String	Yes	Copy contract type setted <code>custom</code> : custom by <code>instId</code> which is required; <code>copy</code> : Keep your contracts consistent with this trader by automatically adding or removing contracts when they do
instId	String	Conditional	Instrument ID. If there are multiple instruments, separate them with commas.
copyMode	String	No	Copy mode <code>fixed_amount</code> : set the same fixed amount for each order, and <code>copyAmt</code> is required; <code>ratio_copy</code> : set amount as a multiple of the lead trader's order value, and <code>copyRatio</code> is required The default is <code>fixed_amount</code>
copyTotalAmt	String	Yes	Maximum total amount in USDT. The maximum total amount you'll invest at any given time across all orders in this copy trade You won't copy new orders if you exceed this amount
copyAmt	String	Conditional	Copy amount per order in USDT.
copyRatio	String	Conditional	Copy ratio per order.
tpRatio	String	No	Take profit per order. 0.1 represents 10%
slRatio	String	No	Stop loss per order. 0.1 represents 10%
slTotalAmt	String	No	Total stop loss in USDT for trader. If your net loss (total profit - total loss) reaches this amount, you'll stop copying this trader
subPosCloseType	String	Yes	Action type for open positions <code>market_close</code> : immediately close at market price <code>copy_close</code> : close when trader closes <code>manual_close</code> : close manually The default is <code>copy_close</code>
tag	String	No	Order tag A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters.

#### Response example

```
{
  "code": "0",
  "data": [
    {
      "result": true
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
result	Boolean	The result of setting <code>true</code>

You need to use this endpoint to amend copy settings

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

`POST /api/v5/copytrading/amend-copy-settings`

Request example

```
POST /api/v5/copytrading/amend-copy-settings
body
{
  "instType": "SWAP",
  "uniqueCode": "25CD5A80241D6FE6",
  "copyMgnMode": "cross",
  "copyInstIdType": "copy",
  "copyMode": "ratio_copy",
  "copyRatio": "1",
  "copyTotalAmt": "500",
  "subPosCloseType": "copy_close"
}
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type <code>SWAP</code>
uniqueCode	String	Yes	Lead trader unique code A combination of case-sensitive alphanumerics, all numbers and the length is 16 characters, e.g. 213E8C92DC61EFAC
copyMgnMode	String	Yes	Copy margin mode <code>cross</code> : cross <code>isolated</code> : isolated <code>copy</code> : Use the same margin mode as lead trader when opening positions
copyInstIdType	String	Yes	Copy contract type setted <code>custom</code> : custom by <code>instId</code> which is required; <code>copy</code> : Keep your contracts consistent with this trader by automatically adding or removing contracts when they do
instId	String	Conditional	Instrument ID. If there are multiple instruments, separate them with commas.
copyMode	String	No	Copy mode <code>fixed_amount</code> : set the same fixed amount for each order, and <code>copyAmt</code> is required; <code>ratio_copy</code> : set amount as a multiple of the lead trader's order value, and <code>copyRatio</code> is required The default is <code>fixed_amount</code>
copyTotalAmt	String	Yes	Maximum total amount in USDT. The maximum total amount you'll invest at any given time across all orders in this copy trade You won't copy new orders if you exceed this amount
copyAmt	String	Conditional	Copy amount per order in USDT
copyRatio	String	Conditional	Copy ratio per order.
tpRatio	String	No	Take profit per order. 0.1 represents 10%
slRatio	String	No	Stop loss per order. 0.1 represents 10%

Parameter	Type	Required	Description
slTotalAmt	String	No	Total stop loss in USDT for trader. If your net loss (total profit - total loss) reaches this amount, you'll stop copying this trader
subPosCloseType	String	Yes	Action type for open positions <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">market_close</span> : immediately close at market price <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">copy_close</span> : close when trader closes <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; margin-right: 10px;">manual_close</span> : close manually The default is <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px;">copy_close</span>
tag	String	No	Order tag A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters.

Response example

```
{
  "code": "0",
  "data": [
    {
      "result": true
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
result	Boolean	The result of setting <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px;">true</span>

#### POST / STOP COPYING

You need to use this endpoint to stop copy trading

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

```
POST /api/v5/copytrading/stop-copy-trading
```

Request example

```
POST /api/v5/copytrading/stop-copy-trading
body
{
  "instType": "SWAP",
  "uniqueCode": "25CD5A80241D6FE6",
  "subPosCloseType": "manual_close"
}
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px;">SWAP</span>
uniqueCode	String	Yes	Lead trader unique code A combination of case-sensitive alphanumerics, all numbers and the length is 16 characters, e.g. 213E8C92DC61EFAC

Parameter	Type	Required	Description
subPosCloseType	String	Yes	Action type for open positions, it is required if you have related copy position <div style="display: flex; justify-content: space-around;"> <span>market_close</span>: immediately close at market price           <span>copy_close</span>: close when trader closes           <span>manual_close</span>: close manually         </div>

Response example

```
{
  "code": "0",
  "data": [
    {
      "result": true
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
result	Boolean	The result of setting <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px; display: inline-block;">true</div>

#### GET / COPY SETTINGS

Retrieve the copy settings about certain lead trader.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

#### HTTP REQUEST

```
GET /api/v5/copytrading/copy-settings
```

Request example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px; display: inline-block;">SWAP</div>
uniqueCode	String	Yes	Lead trader unique code A combination of case-sensitive alphanumerics, all numbers and the length is 16 characters, e.g. 213E8C92DC61EFAC

Response example

```
{
  "code": "0",
  "data": [
    {
      "ccy": "USDT",
      "copyAmt": "",
      "copyInstIdType": "copy",
      "copyMgnMode": "isolated",
      "copyMode": "ratio_copy",
      "copyRatio": "1",
      "copyState": "1",
      "copyTotalAmt": "500",
      "instIds": [
        {
          "enabled": "1",
          "instId": "1"
        }
      ]
    }
  ]
}
```

```

    "instId": "ADA-USDT-SWAP"
  },
  {
    "enabled": "1",
    "instId": "YFII-USDT-SWAP"
  }
],
"slRatio": "",
"slTotalAmt": "",
"subPosCloseType": "copy_close",
"tpRatio": "",
"tag": ""
}
],
"msg": ""
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
copyMode	String	Copy mode fixed_amount ratio_copy
copyAmt	String	Copy amount in USDT per order.
copyRatio	String	Copy ratio per order.
copyTotalAmt	String	Maximum total amount in USDT. The maximum total amount you'll invest at any given time across all orders in this copy trade
tpRatio	String	Take profit per order. 0.1 represents 10%
slRatio	String	Stop loss per order. 0.1 represents 10%
copyInstIdType	String	Copy contract type setted custom: custom by instId which is required; copy: Keep your contracts consistent with this trader by automatically adding or removing contracts when they do
instIds	Array of objects	Instrument list. It will return all lead contracts of the lead trader
> instId	String	Instrument ID
> enabled	String	Whether copying this instId 0 1
slTotalAmt	String	Total stop loss in USDT for trader.
subPosCloseType	String	Action type for open positions market_close: immediately close at market price copy_close: close when trader closes manual_close: close manually
copyMgnMode	String	Copy margin mode cross: CROSS isolated: isolated copy: Use the same margin mode as lead trader when opening positions
ccy	String	Margin currency
copyState	String	Current copy state 0: non-copy, 1: copy
tag	String	Order tag

Retrieve my lead traders.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

```
GET /api/v5/copytrading/current-lead-traders
```

Request example

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
instType	String	No	Instrument type SWAP, the default value

Response example

```
{
  "code": "0",
  "data": [
    {
      "beginCopyTime": "1701224821936",
      "ccy": "USDT",
      "copyTotalAmt": "500",
      "copyTotalPnl": "0",
      "leadMode": "public",
      "margin": "1.89395",
      "nickName": "Trader9527",
      "portLink": "",
      "profitSharingRatio": "0.08",
      "todayPnl": "0",
      "uniqueCode": "25CD5A80241D6FE6",
      "upl": "0"
    }
  ],
  "msg": ""
}
```

**RESPONSE PARAMETERS**

Parameter	Type	Description
portLink	String	Portrait link
nickName	String	Nick name
margin	String	Margin for copy trading
copyTotalAmt	String	Copy total amount
copyTotalPnl	String	Copy total pnl
uniqueCode	String	Lead trader unique code
ccy	String	margin currency
profitSharingRatio	String	Profit sharing ratio. 0.1 represents 10%
beginCopyTime	String	Begin copying time. Unix timestamp format in milliseconds, e.g.1597026383085
upl	String	Unrealized profit & loss
todayPnl	String	Today pnl

Parameter	Type	Description
leadMode	String	Lead mode <code>public</code> <code>private</code>

#### GET / COPY TRADING CONFIGURATION

Public endpoint. Retrieve copy trading parameter configuration information of copy settings

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

**PERMISSION: READ**

#### HTTP REQUEST

`GET /api/v5/copytrading/public-config`

Request example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type <code>SWAP</code> , the default value

Response example

```
{
  "code": "0",
  "data": [
    {
      "maxCopyAmt": "1000",
      "maxCopyRatio": "100",
      "maxCopyTotalAmt": "30000",
      "maxSlRatio": "0.75",
      "maxTpRatio": "1.5",
      "minCopyAmt": "20",
      "minCopyRatio": "0.01"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
maxCopyAmt	String	Maximum copy amount per order in USDT when you are using copy mode <code>fixed_amount</code>
minCopyAmt	String	Minimum copy amount per order in USDT when you are using copy mode <code>fixed_amount</code>
maxCopyTotalAmt	String	Maximum copy total amount under the certain lead trader, the minimum is the same with <code>minCopyAmt</code>
minCopyRatio	String	Minimum ratio per order when you are using copy mode <code>ratio_copy</code>
maxCopyRatio	String	Maximum ratio per order when you are using copy mode <code>ratio_copy</code>
maxTpRatio	String	Maximum ratio of taking profit per order, the minimum is 0
maxSlRatio	String	Maximum ratio of stopping loss per order, the minimum is 0

#### GET / LEAD TRADER RANKS

Public endpoint. Retrieve lead trader ranks.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

## HTTP REQUEST

GET /api/v5/copytrading/public-lead-traders

Request example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type SWAP, the default value
sortType	String	No	Sort type overview: overview, the default value pnl: profit and loss aum: assets under management win_ratio: win ratio pnl_ratio: pnl ratio current_copy_trader_pnl: current copy trader pnl
state	String	No	Lead trader state 0: All lead traders, the default, including vacancy and non-vacancy 1: lead traders who have vacancy
minLeadDays	String	No	Minimum lead days 1: 7 days 2: 30 days 3: 90 days 4: 180 days
minAssets	String	No	Minimum assets in USDT
maxAssets	String	No	Maximum assets in USDT
minAum	String	No	Minimum assets in USDT under management.
maxAum	String	No	Maximum assets in USDT under management.
dataVer	String	No	Data version. It is 14 numbers. e.g. 20231010182400. Generally, it is used for pagination A new version will be generated every 10 minutes. Only last 5 versions are stored The default is latest version. If it is not exist, error will not be throwed and the latest version will be used.
page	String	No	Page for pagination
limit	String	No	Number of results per request. The maximum is 20; the default is 10

Response example

```
{
  "code": "0",
  "data": [
    {
      "dataVer": "20231129213200",
      "ranks": [
        {
          "accCopyTraderNum": "3536",
          "aum": "1509265.3238761567721365",
          "ccy": "USDT",
          "copyState": "0",
          "copyTraderNum": "999",
          "leadDays": "156",
          "maxCopyTraderNum": "1000",
          "nickName": "Crypto to the moon",
          "pnl": "48805.1105999999972258",
          "pnlRatio": "1.6898",
          "pnlRatios": [
            {
              "copyTraderNum": "999",
              "pnl": "48805.1105999999972258",
              "pnlRatio": "1.6898"
            }
          ]
        }
      ]
    }
  ]
}
```

```

        },
        "beginTs": "1701187200000",
        "pnlRatio": "1.6744"
    },
    {
        "beginTs": "1700755200000",
        "pnlRatio": "1.649"
    }
],
"portLink": "https://static.okx.com/cdn/okex/users/headimages/20230624/f49a683aaf5949ea88b01bbc771fb9fc",
"traderInsts": [
    "ICP-USDT-SWAP",
    "MINA-USDT-SWAP"
],
"uniqueCode": "540D011FDACCB47A",
"winRatio": "0.6957"
}
],
"totalPage": "1"
}
],
"msg": ""
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
dataVer	String	Data version
totalPage	String	Total number of pages
ranks	Array of objects	The rank information of lead traders
> aum	String	assets under management
> copyState	String	Current copy state 0: non-copy, 1: copy
> maxCopyTraderNum	String	Maximum number of copy traders
> copyTraderNum	String	Current number of copy traders
> accCopyTraderNum	String	Accumulated number of copy traders
> portLink	String	Portrait link
> nickName	String	Nick name
> ccy	String	Margin currency
> uniqueCode	String	Lead trader unique code
> winRatio	String	Win ratio, 0.1 represents 10%
> leadDays	String	Lead days
> traderInsts	Array of strings	Contract list which lead trader is leading
> pnl	String	Pnl (in USDT) of last 90 days.
> pnlRatio	String	Pnl ratio of last 90 days. 0.1 represents 10%
> pnlRatios	Array of objects	Pnl ratios
>> beginTs	String	Begin time of pnl ratio on that day

Parameter	Type	Description
>> pnlRatio	String	Pnl ratio on that day

#### GET / LEAD TRADER WEEKLY PNL

Public endpoint. Retrieve lead trader weekly pnl. Results are returned in counter chronological order.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

**PERMISSION: READ**

**HTTP REQUEST**

GET /api/v5/copytrading/public-weekly-pnl

Request example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type SWAP, the default value
uniqueCode	String	Yes	Lead trader unique code A combination of case-sensitive alphanumerics, all numbers and the length is 16 characters, e.g. 213E8C92DC61EFAC

Response example

```
{
  "code": "0",
  "data": [
    {
      "beginTs": "1701014400000",
      "pnl": "-2.8428",
      "pnlRatio": "-0.0106"
    },
    {
      "beginTs": "1700409600000",
      "pnl": "81.8446",
      "pnlRatio": "0.3036"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
beginTs	String	Begin time of pnl ratio on that week
pnl	String	Pnl on that week
pnlRatio	String	Pnl ratio on that week

#### GET / LEAD TRADER DAILY PNL

Public endpoint. Retrieve lead trader daily pnl. Results are returned in counter chronological order.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

**PERMISSION: READ**

**HTTP REQUEST**

GET /api/v5/copytrading/public-pnl

Request example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type SWAP, the default value
uniqueCode	String	Yes	Lead trader unique code A combination of case-sensitive alphanumerics, all numbers and the length is 16 characters, e.g. 213E8C92DC61EFAC
lastDays	String	Yes	Last days ①: last 7 days ②: last 30 days ③: last 90 days ④: last 365 days

Response example

```
{  
  "code": "0",  
  "data": [  
    {  
      "beginTs": "1701100800000",  
      "pnl": "97.3309",  
      "pnlRatio": "0.3672"  
    },  
    {  
      "beginTs": "1701014400000",  
      "pnl": "96.7755",  
      "pnlRatio": "0.3651"  
    }  
,  
  ],  
  "msg": ""  
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
beginTs	String	Begin time on that day
pnl	String	Accumulated pnl
pnlRatio	String	Accumulated pnl ratio

#### GET / LEAD TRADER STATS

Public endpoint. Key data related to lead trader performance.

#### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: IP

#### PERMISSION: READ

#### HTTP REQUEST

GET /api/v5/copytrading/public-stats

Request example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type SWAP, the default value
uniqueCode	String	Yes	Lead trader unique code A combination of case-sensitive alphanumerics, all numbers and the length is 16 characters, e.g. 213E8C92DC61EFAC
lastDays	String	Yes	Last days 1: last 7 days 2: last 30 days 3: last 90 days 4: last 365 days

#### Response example

```
{
  "code": "0",
  "data": [
    {
      "avgSubPosNotional": "213.1038",
      "ccy": "USDT",
      "curCopyTraderPnl": "96.8071",
      "investAmt": "265.095252476476294",
      "lossDays": "1",
      "profitDays": "2",
      "winRatio": "0.6667"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
winRatio	String	Win ratio
profitDays	String	Profit days
lossDays	String	Loss days
curCopyTraderPnl	String	Current copy trader pnl (USDT)
avgSubPosNotional	String	Average lead position notional (USDT)
investAmt	String	Investment amount (USDT)
ccy	String	Margin currency

#### GET / LEAD TRADER CURRENCY PREFERENCES

Public endpoint. The most frequently traded crypto of this lead trader. Results are sorted by ratio from large to small.

#### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: IP

#### PERMISSION: READ

#### HTTP REQUEST

```
GET /api/v5/copytrading/public-preference-currency
```

#### Request example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type SWAP, the default value
uniqueCode	String	Yes	Lead trader unique code A combination of case-sensitive alphanumerics, all numbers and the length is 16 characters, e.g. 213E8C92DC61EFAC

#### Response example

```
{
  "code": "0",
  "data": [
    {
      "ccy": "ETH",
      "ratio": "0.8881"
    },
    {
      "ccy": "BTC",
      "ratio": "0.0666"
    },
    {
      "ccy": "YFI",
      "ratio": "0.0453"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency
ratio	String	Ratio. 0.1 represents 10%

#### GET / LEAD TRADER CURRENT LEAD POSITIONS

Public endpoint. Get current leading positions of lead trader

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

**PERMISSION: READ**

**HTTP REQUEST**

```
GET /api/v5/copytrading/public-current-subpositions
```

#### Request example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type SWAP, the default value.
uniqueCode	String	Yes	Lead trader unique code A combination of case-sensitive alphanumerics, all numbers and the length is 16 characters, e.g. 213E8C92DC61EFAC
after	String	No	Pagination of data to return records earlier than the requested <code>subPosId</code> .
before	String	No	Pagination of data to return records newer than the requested <code>subPosId</code> .

Parameter	Type	Required	Description
limit	String	No	Number of results per request. Maximum is 100. Default is 100.

#### Response example

```
{
  "code": "0",
  "data": [
    {
      "ccy": "USDT",
      "instId": "ETH-USDT-SWAP",
      "instType": "SWAP",
      "lever": "5",
      "margin": "16.23304",
      "markPx": "2027.31",
      "mgnMode": "isolated",
      "openAvgPx": "2029.13",
      "openTime": "1701144639417",
      "posSide": "short",
      "subPos": "4",
      "subPosId": "64958293098104064",
      "uniqueCode": "D9ADEAB33AE9EABD",
      "upl": "0.0728",
      "uplRatio": "0.0044846806266725"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID, e.g. BTC-USDT-SWAP
subPosId	String	Lead position ID
posSide	String	Position side <input type="radio"/> long <input type="radio"/> short <input type="radio"/> net (Long positions have positive subPos; short positions have negative subPos)
mgnMode	String	Margin mode. <input type="radio"/> cross <input type="radio"/> isolated
lever	String	Leverage
openAvgPx	String	Average open price
openTime	String	Open time
subPos	String	Quantity of positions
instType	String	Instrument type
margin	String	Margin
upl	String	Unrealized profit and loss
uplRatio	String	Unrealized profit and loss ratio
markPx	String	Latest mark price, only applicable to contract
uniqueCode	String	Lead trader unique code
ccy	String	Currency

## GET / LEAD TRADER LEAD POSITION HISTORY

Public endpoint. Retrieve the lead trader completed leading position of the last 3 months.

Returns reverse chronological order with `subPosId`.

### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: IP

### PERMISSION: READ

### HTTP REQUEST

```
GET /api/v5/copytrading/public-subpositions-history
```

Request example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type <code>SWAP</code> , the default value.
uniqueCode	String	Yes	Lead trader unique code A combination of case-sensitive alphanumerics, all numbers and the length is 16 characters, e.g. 213E8C92DC61EFAC
after	String	No	Pagination of data to return records earlier than the requested <code>subPosId</code> .
before	String	No	Pagination of data to return records newer than the requested <code>subPosId</code> .
limit	String	No	Number of results per request. Maximum is 100. Default is 100.

Response example

```
{
  "code": "0",
  "data": [
    {
      "ccy": "USDT",
      "closeAvgPx": "28385.9",
      "closeTime": "1697709137162",
      "instId": "BTC-USDT-SWAP",
      "instType": "SWAP",
      "lever": "20",
      "margin": "4.245285",
      "mgnMode": "isolated",
      "openAvgPx": "28301.9",
      "openTime": "1697698048031",
      "pnl": "0.252",
      "pnlRatio": "0.05935997229868",
      "posSide": "long",
      "subPos": "3",
      "subPosId": "635126416883355648",
      "uniqueCode": "9A8534AB09862774"
    }
  ],
  "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID, e.g. BTC-USDT-SWAP
subPosId	String	Lead position ID
posSide	String	Position side <code>long</code>

Parameter	Type	Description
		<span>short</span> <span>net</span> (long position has positive subPos; short position has negative subPos)
mgnMode	String	Margin mode. <span>cross</span> <span>isolated</span>
lever	String	Leverage
openAvgPx	String	Average open price
openTime	String	Time of opening
subPos	String	Quantity of positions
closeTime	String	Time of closing position
closeAvgPx	String	Average price of closing position
pnl	String	Profit and loss
pnlRatio	String	P&L ratio
instType	String	Instrument type
margin	String	Margin
ccy	String	Currency
uniqueCode	String	Lead trader unique code

#### GET / COPY TRADERS

Public endpoint. Retrieve copy trader coming from certain lead trader. Return according to pnl from high to low

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/copytrading/public-copy-traders`

Request example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	No	Instrument type <span>SWAP</span> , the default value
uniqueCode	String	Yes	Lead trader unique code A combination of case-sensitive alphanumerics, all numbers and the length is 16 characters, e.g. 213E8C92DC61EFAC
limit	String	No	Number of results per request. The maximum is <span>100</span> ; The default is <span>100</span>

Response example

```
{
  "code": "0",
  "data": [
    {
      "ccy": "USDT",
      "copyTotalPnl": "2060.12242",
      "instType": "SWAP"
    }
  ]
}
```

```

"copyTraderNumChg": "1",
"copyTraderNumChgRatio": "0.5",
"copyTraders": [
  {
    "beginCopyTime": "1686125051000",
    "nickName": "bre***@gmail.com",
    "pnl": "1076.77388",
    "portLink": ""
  },
  {
    "beginCopyTime": "1698133811000",
    "nickName": "MrYanDao505",
    "pnl": "983.34854",
    "portLink": "https://static.okx.com/cdn/okex/users/headimages/20231010/fd31f45e99fe41f7bb219c0b53ae0ada"
  }
]
},
"msg": ""
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
copyTotalPnl	String	Total copy trader profit and loss
ccy	String	The currency name of profit and loss
copyTraderNumChg	String	Number change in last 7 days
copyTraderNumChgRatio	String	Ratio change in last 7 days
copyTraders	Array of objects	Copy trader information
> beginCopyTime	String	Begin copying time. Unix timestamp format in milliseconds, e.g.1597026383085
> nickName	String	Nick name
> portLink	String	Copy trader portrait link
> pnl	String	Copy trading profit and loss

#### WS / LEAD TRADING NOTIFICATION CHANNEL

The notification when failing to lead trade.

#### URL PATH

/ws/v5/business (required login)

#### Request Example

```

import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/business",
        useServerTime=False
    )
    await ws.start()

```

```

args = [{  
    "channel": "copytrading-lead-notification",  
    "instType": "SWAP"  
}]  
  
await ws.subscribe(args, callback=callbackFunc)  
await asyncio.sleep(10)  
  
await ws.unsubscribe(args, callback=callbackFunc)  
await asyncio.sleep(10)  
  
asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>copytrading-lead-notification</code>
> instType	String	Yes	Instrument type <code>SWAP</code>
> instId	String	No	Instrument ID

## Successful Response Example

```
{
    "id": "1512",
    "event": "subscribe",
    "arg": {
        "channel": "copytrading-lead-notification",
        "instType": "SWAP"
    },
    "connId": "aa993428"
}
```

## Failure Response Example

```
{
    "id": "1512",
    "event": "error",
    "code": "60012",
    "msg": "Invalid request: {\\"op\\": \\"subscribe\\", \\"args\\": [{ \\"channel\\": \\"copytrading-lead-notification\\", \\"instType\\": \\"FUTURES\\\" }]}",
    "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event <code>subscribe</code>

Parameter	Type	Required	Description
			<div style="display: flex; justify-content: space-around; align-items: center;"> <span>unsubscribe</span> <span>error</span> </div>
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> instType	String	Yes	Instrument type SWAP
> instId	String	No	Instrument ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

Push Data Example:

```
{
  "arg": {
    "channel": "copytrading-lead-notification",
    "instType": "SWAP",
    "uid": "525627088439549953"
  },
  "data": [
    {
      "infoType": "2",
      "instId": "",
      "instType": "SWAP",
      "maxLeadTraderNum": "3",
      "minLeadEq": "",
      "posSide": "",
      "side": "",
      "subPosId": "667695035433385984",
      "uniqueCode": "3AF72F63E3EAD701"
    }
  ]
}
```

#### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> uid	String	User Identifier
> instType	String	Instrument type
data	Array of objects	Subscribed data
> instType	String	Instrument type
> infoType	String	Information type <ul style="list-style-type: none"> <li>1: lead trading failed due to touch max position limitation</li> <li>2: lead trading failed due to touch the maximum daily number of lead trading</li> <li>3: lead trading failed due to your USDT equity less than the minimum USDT equity of lead trading</li> </ul>
> subPosId	String	Lead position ID
> uniqueCode	String	Lead trader unique code

Parameter	Type	Description
> instId	String	Instrument ID
> side	String	Side <code>buy</code> <code>sell</code>
> posSide	String	Position side <code>long</code> <code>short</code> <code>net</code>
> maxLeadTraderNum	String	Maximum daily number of lead trading.
> minLeadEq	String	Minimum USDT equity of lead trading.

## Market Data

The API endpoints of `Market Data` do not require authentication.

There are multiple services for market data, and each service has an independent cache. A random service will be requested for every request. So for two requests, it's expected that the data obtained in the second request is earlier than the first request.

### GET / TICKERS

Retrieve the latest price snapshot, best bid/ask price, and trading volume in the last 24 hours. Best ask price may be lower than the best bid price during the pre-open period.

#### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: IP

#### HTTP REQUEST

`GET /api/v5/market/tickers`

#### Request Example

```
import okx.MarketData as MarketData

flag = "0" # Production trading:0 , demo trading:1

marketDataAPI = MarketData.MarketAPI(flag=flag)

# Retrieve the latest price snapshot, best bid/ask price, and trading volume in the last 24 hours
result = marketDataAPI.get_tickers(
    instType="SWAP"
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	Yes	Instrument type <code>SPOT</code> <code>SWAP</code> <code>FUTURES</code> <code>OPTION</code>
instFamily	String	No	Instrument family Applicable to <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "instType": "SWAP",
      "instId": "LTC-USD-SWAP",
      "last": "9999.99",
      "lastSz": "1",
      "askPx": "9999.99",
      "askSz": "11",
      "bidPx": "8888.88",
      "bidSz": "5",
      "open24h": "9000",
      "high24h": "10000",
      "low24h": "8888.88",
      "volCcy24h": "2222",
      "vol24h": "2222",
      "sodUtc0": "0.1",
      "sodUtc8": "0.1",
      "ts": "1597026383085"
    },
    {
      "instType": "SWAP",
      "instId": "BTC-USD-SWAP",
      "last": "9999.99",
      "lastSz": "1",
      "askPx": "9999.99",
      "askSz": "11",
      "bidPx": "8888.88",
      "bidSz": "5",
      "open24h": "9000",
      "high24h": "10000",
      "low24h": "8888.88",
      "volCcy24h": "2222",
      "vol24h": "2222",
      "sodUtc0": "0.1",
      "sodUtc8": "0.1",
      "ts": "1597026383085"
    }
  ]
}
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type
instId	String	Instrument ID
last	String	Last traded price
lastSz	String	Last traded size. 0 represents there is no trading volume
askPx	String	Best ask price
askSz	String	Best ask size
bidPx	String	Best bid price
bidSz	String	Best bid size
open24h	String	Open price in the past 24 hours
high24h	String	Highest price in the past 24 hours
low24h	String	Lowest price in the past 24 hours
volCcy24h	String	24h trading volume, with a unit of <a href="#">currency</a> . If it is a <a href="#">derivatives</a> contract, the value is the number of base currency. e.g. the unit is BTC for BTC-USD-SWAP and BTC-

Parameter	Type	Description
		USDT-SWAP If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in quote currency.
vol24h	String	24h trading volume, with a unit of <code>contract</code> . If it is a <code>derivatives</code> contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in base currency.
sodUtc0	String	Open price in the UTC 0
sodUtc8	String	Open price in the UTC 8
ts	String	Ticker data generation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## GET / TICKER

Retrieve the latest price snapshot, best bid/ask price, and trading volume in the last 24 hours. Best ask price may be lower than the best bid price during the pre-open period.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

### HTTP REQUEST

`GET /api/v5/market/ticker`

#### Request Example

```
import okx.MarketData as MarketData

flag = "0" # Production trading:0 , demo trading:1

marketDataAPI = MarketData.MarketAPI(flag=flag)

# Retrieve the latest price snapshot, best bid/ask price, and trading volume in the last 24 hours
result = marketDataAPI.get_ticker(
    instId="BTC-USD-SWAP"
)
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USD-SWAP</code>

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "instType": "SWAP",
      "instId": "BTC-USD-SWAP",
      "last": "9999.99",
      "lastSz": "0.1",
      "askPx": "9999.99",
      "askSz": "11",
      "bidPx": "8888.88",
      "bidSz": "5",
      "open24h": "9000",
      "high24h": "10000",
      "low24h": "8888.88",
      "volCcy24h": "2222",
      "vol24h": "2222",
      "sodUtc0": "2222",
      "sodUtc8": "2222",
      "ts": "1597026383085"
    }
  ]
}
```

```
        "ts": "1597026383085"
```

```
}
```

```
]
```

```
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type
instId	String	Instrument ID
last	String	Last traded price
lastSz	String	Last traded size. 0 represents there is no trading volume
askPx	String	Best ask price
askSz	String	Best ask size
bidPx	String	Best bid price
bidSz	String	Best bid size
open24h	String	Open price in the past 24 hours
high24h	String	Highest price in the past 24 hours
low24h	String	Lowest price in the past 24 hours
volCcy24h	String	24h trading volume, with a unit of <code>currency</code> . If it is a <code>derivatives</code> contract, the value is the number of base currency. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in quote currency.
vol24h	String	24h trading volume, with a unit of <code>contract</code> . If it is a <code>derivatives</code> contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in base currency.
sodUtc0	String	Open price in the UTC 0
sodUtc8	String	Open price in the UTC 8
ts	String	Ticker data generation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code> .

## GET / ORDER BOOK

Retrieve order book of the instrument. The data will be updated once every 50 milliseconds. Best ask price may be lower than the best bid price during the pre-open period.

This endpoint does not return data immediately. Instead, it returns the latest data once the server-side cache has been updated.

**RATE LIMIT: 40 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

## HTTP REQUEST

```
GET /api/v5/market/books
```

Request Example

```
import okx.MarketData as MarketData

flag = "0" # Production trading:0 , demo trading:1

marketDataAPI = MarketData.MarketAPI(flag=flag)

# Retrieve order book of the instrument
```

```

result = marketDataAPI.get_orderbook(
    instId="BTC-USDT"
)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT</code>
sz	String	No	Order book depth per side. Maximum 400, e.g. 400 bids + 400 asks Default returns to ① depth data

## Response Example

```

{
  "code": "0",
  "msg": "",
  "data": [
    {
      "asks": [
        [
          [
            "41006.8",
            "0.60038921",
            "0",
            "1"
          ]
        ],
        [
          [
            "41006.3",
            "0.30178218",
            "0",
            "2"
          ]
        ],
        "ts": "1629966436396"
      ]
    }
  ]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
asks	Array of Arrays	Order book on sell side
bids	Array of Arrays	Order book on buy side
ts	String	Order book generation time

An example of the array of asks and bids values: ["411.8", "10", "0", "4"]

- "411.8" is the depth price
- "10" is the quantity at the price (number of contracts for derivatives, quantity in base currency for Spot and Spot Margin)
- "0" is part of a deprecated feature and it is always "0"
- "4" is the number of orders at the price.

The order book data will be updated around once a second during the call auction.

## GET / FULL ORDER BOOK

Retrieve order book of the instrument. The data will be updated once a second. Best ask price may be lower than the best bid price during the pre-open period.

This endpoint does not return data immediately. Instead, it returns the latest data once the server-side cache has been updated.

#### RATE LIMIT: 10 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: IP

#### HTTP REQUEST

```
GET /api/v5/market/books-full
```

#### Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT</code>
sz	String	No	Order book depth per side. Maximum 5000, e.g. 5000 bids + 5000 asks Default returns to <code>1</code> depth data.

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "asks": [
        [
          [
            "41006.8",
            "0.60038921",
            "1"
          ]
        ],
        "bids": [
          [
            "41006.3",
            "0.30178218",
            "2"
          ]
        ],
        "ts": "1629966436396"
      ]
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
asks	Array of Arrays	Order book on sell side
bids	Array of Arrays	Order book on buy side
ts	String	Order book generation time

An example of the array of asks and bids values: `["411.8", "10", "4"]`

- `"411.8"` is the depth price
- `"10"` is the quantity at the price (number of contracts for derivatives, quantity in base currency for Spot and Spot Margin)
- `"4"` is the number of orders at the price.

The order book data will be updated around once a second during the call auction.

Retrieve the candlestick charts. This endpoint can retrieve the latest 1,440 data entries. Charts are returned in groups based on the requested bar.

#### RATE LIMIT: 40 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: IP

#### HTTP REQUEST

GET /api/v5/market/candles

#### Request Example

```
import okx.MarketData as MarketData

flag = "0" # Production trading:0 , demo trading:1

marketDataAPI = MarketData.MarketAPI(flag=flag)

# Retrieve the candlestick charts
result = marketDataAPI.get_candlesticks(
    instId="BTC-USDT"
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT</code>
bar	String	No	Bar size, the default is <code>1m</code> e.g. [1m/3m/5m/15m/30m/1H/2H/4H] UTC+8 opening price k-line: [6H/12H/1D/2D/3D/1W/1M/3M] UTC+0 opening price k-line: [6Hutc/12Hutc/1Dutc/2Dutc/3Dutc/1Wutc/1Mutc/3Mutc]
after	String	No	Pagination of data to return records earlier than the requested <code>ts</code>
before	String	No	Pagination of data to return records newer than the requested <code>ts</code> . The latest data will be returned when using <code>before</code> individually
limit	String	No	Number of results per request. The maximum is <code>300</code> . The default is <code>100</code> .

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    [
      "1597026383085",
      "3.721",
      "3.743",
      "3.677",
      "3.708",
      "8422410",
      "22698348.04828491",
      "12698348.04828491",
      "0"
    ],
    [
      "1597026383085",
      "3.731",
      "3.799",
      "3.494",
      "3.72",
      "24912403",
      "67632347.24399722",
      "37632347.24399722",
      "1"
    ]
  ]
}
```

]  
}

## RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Opening time of the candlestick, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
o	String	Open price
h	String	Highest price
l	String	Lowest price
c	String	Close price
vol	String	Trading volume, with a unit of <code>contract</code> . If it is a <code>derivatives</code> contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in base currency.
volCcy	String	Trading volume, with a unit of <code>currency</code> . If it is a <code>derivatives</code> contract, the value is the number of base currency. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in quote currency.
volCcyQuote	String	Trading volume, the value is the quantity in quote currency e.g. The unit is USDT for BTC-USDT and BTC-USDT-SWAP; The unit is USD for BTC-USD-SWAP
confirm	String	The state of candlesticks. <code>0</code> : K line is uncompleted <code>1</code> : K line is completed

The first candlestick data may be incomplete, and should not be polled repeatedly.

The data returned will be arranged in an array like this: `[ts,o,h,l,c,vol,volCcy,volCcyQuote,confirm]`.

For the current cycle of k-line data, when there is no transaction, the opening high and closing low default take the closing price of the previous cycle.

## GET / CANDLESTICKS HISTORY

Retrieve history candlestick charts from recent years (It is last 3 months supported for 1s candlestick).

### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: IP

### HTTP REQUEST

`GET /api/v5/MarketData/history-candles`

### Request Example

```
import okx.MarketData as MarketData

flag = "0" # Production trading:0 , demo trading:1

marketDataAPI = MarketData.MarketAPI(flag=flag)

# Retrieve history candlestick charts from recent years
result = marketDataAPI.get_history_candlesticks(
    instId="BTC-USDT"
)
```

```
)  
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT</code>
after	String	No	Pagination of data to return records earlier than the requested <code>ts</code>
before	String	No	Pagination of data to return records newer than the requested <code>ts</code> . The latest data will be returned when using <code>before</code> individually
bar	String	No	Bar size, the default is <code>1m</code> e.g. [1s/1m/3m/5m/15m/30m/1H/2H/4H] UTC+8 opening price k-line: [6H/12H/1D/2D/3D/1W/1M/3M] UTC+0 opening price k-line: [6Hutc/12Hutc/1Dutc/2Dutc/3Dutc/1Wutc/1Mutc/3Mutc]
limit	String	No	Number of results per request. The maximum is <code>300</code> . The default is <code>100</code> .

## Response Example

```
{  
  "code": "0",  
  "msg": "",  
  "data": [  
    [  
      "1597026383085",  
      "3.721",  
      "3.743",  
      "3.677",  
      "3.708",  
      "8422410",  
      "22698348.04828491",  
      "12698348.04828491",  
      "1"  
    ],  
    [  
      "1597026383085",  
      "3.731",  
      "3.799",  
      "3.494",  
      "3.72",  
      "24912403",  
      "67632347.24399722",  
      "37632347.24399722",  
      "1"  
    ]  
  ]  
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Opening time of the candlestick, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
o	String	Open price
h	String	Highest price
l	String	Lowest price
c	String	Close price
vol	String	Trading volume, with a unit of <code>contract</code> . If it is a <code>derivatives</code> contract, the value is the number of contracts.

Parameter	Type	Description
		If it is <b>SPOT</b> / <b>MARGIN</b> , the value is the quantity in base currency.
volCcy	String	Trading volume, with a unit of <b>currency</b> . If it is a <b>derivatives</b> contract, the value is the number of base currency. If it is <b>SPOT</b> / <b>MARGIN</b> , the value is the quantity in quote currency.
volCcyQuote	String	Trading volume, the value is the quantity in quote currency e.g. The unit is USDT for BTC-USDT and BTC-USDT-SWAP; The unit is USD for BTC-USD-SWAP
confirm	String	The state of candlesticks 0: K line is uncompleted 1: K line is completed

The data returned will be arranged in an array like this: [ts,o,h,l,c,vol,volCcy,volCcyQuote,confirm]

1s candle is not supported by OPTION, but it is supported by other business lines (SPOT, MARGIN, FUTURES and SWAP)

## GET / TRADES

Retrieve the recent transactions of an instrument.

**RATE LIMIT: 100 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

### HTTP REQUEST

GET /api/v5/market/trades

#### Request Example

```
import okx.MarketData as MarketData

flag = "0" # Production trading:0 , demo trading:1

marketDataAPI = MarketData.MarketAPI(flag=flag)

# Retrieve the recent transactions of an instrument
result = marketDataAPI.get_trades(
    instId="BTC-USDT"
)
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <b>BTC-USDT</b>
limit	String	No	Number of results per request. The maximum is <b>500</b> ; The default is <b>100</b>

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "instId": "BTC-USDT",
```

```

        "side": "sell",
        "sz": "0.00001",
        "source": "0",
        "px": "29963.2",
        "tradeId": "242720720",
        "ts": "1654161646974"
    },
    {
        "instId": "BTC-USDT",
        "side": "sell",
        "sz": "0.00001",
        "source": "0",
        "px": "29964.1",
        "tradeId": "242720719",
        "ts": "1654161641568"
    }
]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID
tradId	String	Trade ID
px	String	Trade price
sz	String	Trade quantity For spot trading, the unit is base currency For <b>FUTURES</b> / <b>SWAP</b> / <b>OPTION</b> , the unit is contract.
side	String	Trade side of taker <b>buy</b> <b>sell</b>
source	String	Order source 0: normal
ts	String	Trade time, Unix timestamp format in milliseconds, e.g. <b>1597026383085</b> .

Up to 500 most recent historical public transaction data can be retrieved.

## GET / TRADES HISTORY

Retrieve the recent transactions of an instrument from the last 3 months with pagination.

### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: IP

### HTTP REQUEST

**GET /api/v5/market/history-trades**

### Request Example

```

import okx.MarketData as MarketData

flag = "0" # Production trading:0 , demo trading:1

marketDataAPI = MarketData.MarketAPI(flag=flag)

# Retrieve the recent transactions of an instrument from the last 3 months with pagination
result = marketDataAPI.get_history_trades(
    instId="BTC-USD-SWAP"
)

```

```
)
```

```
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT</code>
type	String	No	Pagination Type <code>1</code> : tradeld <code>2</code> : timestamp The default is <code>1</code>
after	String	No	Pagination of data to return records earlier than the requested tradeld or ts.
before	String	No	Pagination of data to return records newer than the requested tradeld. Do not support timestamp for pagination. The latest data will be returned when using <code>before</code> individually
limit	String	No	Number of results per request. The maximum and default both are <code>100</code>

## Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "instId": "BTC-USDT",
      "side": "sell",
      "sz": "0.00001",
      "source": "0",
      "px": "29963.2",
      "tradeId": "242720720",
      "ts": "1654161646974"
    },
    {
      "instId": "BTC-USDT",
      "side": "sell",
      "sz": "0.00001",
      "source": "0",
      "px": "29964.1",
      "tradeId": "242720719",
      "ts": "1654161641568"
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID
tradeld	String	Trade ID
px	String	Trade price
sz	String	Trade quantity For spot trading, the unit is base currency For <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code> , the unit is contract.
side	String	Trade side of taker <code>buy</code> <code>sell</code>
source	String	Order source <code>0</code> : normal

Parameter	Type	Description
ts	String	Trade time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code> .

## GET / OPTION TRADES BY INSTRUMENT FAMILY

Retrieve the recent transactions of an instrument under same instFamily. The maximum is 100.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

### HTTP REQUEST

`GET /api/v5/market/option/instrument-family-trades`

Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instFamily	String	Yes	Instrument family, e.g. BTC-USD Applicable to <code>OPTION</code>

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "vol24h": "103381",
      "tradeInfo": [
        {
          "instId": "BTC-USD-221111-17750-C",
          "side": "sell",
          "sz": "1",
          "px": "0.0075",
          "tradeId": "20",
          "ts": "1668090715058"
        },
        {
          "instId": "BTC-USD-221111-17750-C",
          "side": "sell",
          "sz": "91",
          "px": "0.01",
          "tradeId": "19",
          "ts": "1668090421062"
        }
      ],
      "optType": "C"
    },
    {
      "vol24h": "144499",
      "tradeInfo": [
        {
          "instId": "BTC-USD-230127-10000-P",
          "side": "sell",
          "sz": "82",
          "px": "0.019",
          "tradeId": "23",
          "ts": "1668090967057"
        },
        {
          "instId": "BTC-USD-221111-16250-P",
          "side": "sell",
          "sz": "102",
          "px": "0.0045",
          "tradeId": "24",
          "ts": "1668090885050"
        }
      ],
      "optType": "P"
    }
  ]
}
```

```

    }
]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
vol24h	String	24h trading volume, with a unit of contract.
optType	String	Option type, C: Call P: Put
tradeInfo	Array of objects	The list trade data
> instId	String	The Instrument ID
> tradeId	String	Trade ID
> px	String	Trade price
> sz	String	Trade quantity. The unit is contract.
> side	String	Trade side <code>buy</code> <code>sell</code>
> ts	String	Trade time, Unix timestamp format in milliseconds, e.g. 1597026383085.

## GET / OPTION TRADES

The maximum is 100.

### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: IP

### HTTP REQUEST

```
GET /api/v5/public/option-trades
```

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Conditional	Instrument ID, e.g. BTC-USD-221230-4000-C, Either <code>instId</code> or <code>instFamily</code> is required. If both are passed, <code>instId</code> will be used.
instFamily	String	Conditional	Instrument family, e.g. BTC-USD
optType	String	No	Option type, C: Call P: put

Response Example

```
{
  "code": "0",
  "data": [
    {
      "fillVol": "0.24415013671875",
      "fwdPx": "16676.907614127158",
      "idxPx": "16667",
      "instFamily": "BTC-USD",
      "instId": "BTC-USD-221230-16600-P",
      "markPx": "0.006308943261227884",
      "optType": "P",
      "px": "0.005",
      "side": "sell",
      "sz": "30",
      "tradeId": "65",
      "ts": "1597026383085"
    }
  ]
}
```

```
        "ts": "1672225112048"
    }
],
"msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID
instFamily	String	Instrument family
tradeId	String	Trade ID
px	String	Trade price
> sz	String	Trade quantity. The unit is contract.
side	String	Trade side <input type="button" value="buy"/> <input type="button" value="sell"/>
optType	String	Option type, C: Call P: Put
fillVol	String	Implied volatility while trading (Correspond to trade price)
fwdPx	String	Forward price while trading
idxPx	String	Index price while trading
markPx	String	Mark price while trading
ts	String	Trade time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

### GET / 24H TOTAL VOLUME

The 24-hour trading volume is calculated on a rolling basis.

**RATE LIMIT: 2 REQUESTS PER 2 SECONDS**

#### RATE LIMIT RULE: IP

## HTTP REQUEST

GET /api/v5/market/platform-24-volume

### Request Example

```
import okx.MarketData as MarketData

flag = "0" # Production trading:0 , demo trading:1

marketDataAPI = MarketData.MarketAPI(flag=flag)

# Retrieve 24 total volume
result = marketDataAPI.get_volume()

print(result)
```

### Response Example

```
{  
  "code": "0",  
  "msg": "",  
  "data": [  
    {  
      "volCny": "230900886396766",  
      "volUsd": "34462818865189"
```

```
        "ts": "1657856040389"
```

```
}
```

```
]
```

```
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
volUsd	String	24-hour total trading volume from the order book trading in "USD"
volCny	String	24-hour total trading volume from the order book trading in "CNY"
ts	String	Data return time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## GET / CALL AUCTION DETAILS

Retrieve call auction details.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

**HTTP REQUEST**

```
GET /api/v5/market/call-auction-details
```

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT</code>

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "instId": "ONDO-USDC",
      "unmatchedSz": "9988764",
      "eqPx": "0.6",
      "matchedSz": "44978",
      "state": "continuous_trading",
      "auctionEndTime": "172654200000",
      "ts": "172654200007"
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID
eqPx	String	Equilibrium price
matchedSz	String	Matched size for both buy and sell The unit is in base currency
unmatchedSz	String	Unmatched size
auctionEndTime	String	Call auction end time. Unix timestamp in milliseconds.

Parameter	Type	Description
state	String	Trading state of the symbol <code>call_auction</code> <code>continuous_trading</code>
ts	String	Data generation time. Unix timestamp in milliseconds.

During call auction, users can get the updates of equilibrium price, matched size, unmatched size, and auction end time. The data will be updated around once a second. The endpoint returns the actual open price, matched size, and unmatched size when the call auction ends. For symbols that never go through call auction, the endpoint will also return results but with state always as `continuous\_trading` and other fields as 0 or empty.

## WS / TICKERS CHANNEL

Retrieve the last traded price, bid price, ask price and 24-hour trading volume of instruments. Best ask price may be lower than the best bid price during the pre-open period.

The fastest rate is 1 update/100ms. There will be no update if the event is not triggered. The events which can trigger update: trade, the change on best ask/bid.

### URL PATH

/ws/v5/public

#### Request Example

```
import asyncio

from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/public")
    await ws.start()
    args = [
        {"channel": "tickers",
         "instId": "BTC-USDT"
    }]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels

Parameter	Type	Required	Description
> channel	String	Yes	Channel name tickers
> instId	String	Yes	Instrument ID

## Successful Response Example

```
{  
  "id": "1512",  
  "event": "subscribe",  
  "arg": {  
    "channel": "tickers",  
    "instId": "BTC-USDT"  
  },  
  "connId": "a4d3ae55"  
}
```

## Failure Response Example

```
{  
  "id": "1512",  
  "event": "error",  
  "code": "60012",  
  "msg": "Invalid request: {\\"op\\": \\"subscribe\\", \\"argss\\": [{ \\"channel\\": \\"tickers\\", \\"instId\\": \\"LTC-USD-200327\\"}]}",  
  "connId": "a4d3ae55"  
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event subscribe unsubscribe error
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> instId	String	Yes	Instrument ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

## Push Data Example

```
{  
  "arg": {  
    "channel": "tickers",  
    "instId": "BTC-USDT"  
  },  
  "data": [  
    {  
      "instType": "SPOT",  
      "instId": "BTC-USDT",  
      "last": "9999.99",  
      "lastSz": "0.1",  
      "askPx": "9999.99",  
      "askSz": "11",  
      "bidPx": "9999.98",  
      "bidSz": "11",  
      "avgPx": "9999.98",  
      "highPx": "9999.99",  
      "lowPx": "9999.98",  
      "volume": 11  
    }  
  ]  
}
```

```

        "bidPx": "8888.88",
        "bidSz": "5",
        "open24h": "9000",
        "high24h": "10000",
        "low24h": "8888.88",
        "volCcy24h": "2222",
        "vol24h": "2222",
        "sodUtc0": "2222",
        "sodUtc8": "2222",
        "ts": "1597026383085"
    }
]
}

```

#### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> instId	String	Instrument ID
data	Array of objects	Subscribed data
> instType	String	Instrument type
> instId	String	Instrument ID
> last	String	Last traded price
> lastSz	String	Last traded size. 0 represents there is no trading volume
> askPx	String	Best ask price
> askSz	String	Best ask size
> bidPx	String	Best bid price
> bidSz	String	Best bid size
> open24h	String	Open price in the past 24 hours
> high24h	String	Highest price in the past 24 hours
> low24h	String	Lowest price in the past 24 hours
> volCcy24h	String	24h trading volume, with a unit of <code>currency</code> . If it is a <code>derivatives</code> contract, the value is the number of base currency. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in quote currency.
> vol24h	String	24h trading volume, with a unit of <code>contract</code> . If it is a <code>derivatives</code> contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in base currency.
> sodUtc0	String	Open price in the UTC 0
> sodUtc8	String	Open price in the UTC 8
> ts	String	Ticker data generation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

#### WS / CANDLESTICKS CHANNEL

Retrieve the candlesticks data of an instrument. the push frequency is the fastest interval 1 second push the data.

#### URL PATH

## Request Example

```

import asyncio

from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/business")
    await ws.start()
    args = [
        {
            "channel": "candle1D",
            "instId": "BTC-USDT"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>candle3M</code> <code>candle1M</code> <code>candle1W</code> <code>candle1D</code> <code>candle2D</code> <code>candle3D</code> <code>candle5D</code> <code>candle12H</code> <code>candle6H</code> <code>candle4H</code> <code>candle2H</code> <code>candle1H</code> <code>candle30m</code> <code>candle15m</code> <code>candle5m</code> <code>candle3m</code> <code>candle1m</code> <code>candle1s</code> <code>candle3Mutc</code> <code>candle1Mutc</code> <code>candle1Wutc</code> <code>candle1Dutc</code>

Parameter	Type	Required	Description
> instId	String	Yes	Instrument ID

#### Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "candle1D",
    "instId": "BTC-USDT"
  },
  "connId": "a4d3ae55"
}
```

#### Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \"subscribe\", \\"argss\\\": [{ \\"channel\\\" : \"candle1D\", \\"instId\\\" : \"BTC-USD-191227\" }]}",
  "connId": "a4d3ae55"
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event   
arg	Object	No	Subscribed channel
> channel	String	yes	channel name
> instId	String	Yes	Instrument ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

#### Push Data Example

```
{
  "arg": {
    "channel": "candle1D",
    "instId": "BTC-USDT"
  },
  "data": [
    [
      "1597026383085",
      "8533.02",
      "8533.02",
      "8533.02",
      "8533.02",
      "8533.02"
    ]
  ]
}
```

```

    "8553.74",
    "8527.17",
    "8548.26",
    "45247",
    "529.5858061",
    "5529.5858061",
    "0"
]
]
}

```

#### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> instId	String	Instrument ID
data	Array of Arrays	Subscribed data
> ts	String	Opening time of the candlestick, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> o	String	Open price
> h	String	Highest price
> l	String	Lowest price
> c	String	Close price
> vol	String	Trading volume, with a unit of <code>contract</code> . If it is a <code>derivatives</code> contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in base currency.
> volCcy	String	Trading volume, with a unit of <code>currency</code> . If it is a <code>derivatives</code> contract, the value is the number of base currency. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in quote currency.
> volCcyQuote	String	Trading volume, the value is the quantity in quote currency e.g. The unit is <code>USDT</code> for <code>BTC-USDT</code> and <code>BTC-USDT-SWAP</code> The unit is <code>USD</code> for <code>BTC-USD-SWAP</code>
> confirm	String	The state of candlesticks <code>0</code> : K line is uncompleted <code>1</code> : K line is completed

#### WS / TRADES CHANNEL

Retrieve the recent trades data. Data will be pushed whenever there is a trade. Every update may aggregate multiple trades.

The message is sent only once per taker order, per filled price. The count field is used to represent the number of aggregated matches.

#### URL PATH

`/ws/v5/public`

#### Request Example

```

import asyncio

from okx.websocket.WsPublicAsync import WsPublicAsync

```

```

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/public")
    await ws.start()
    args = [
        {
            "channel": "trades",
            "instId": "BTC-USDT"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation subscribe unsubscribe
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name trades
> instId	String	Yes	Instrument ID

### Successful Response Example

```
{
    "id": "1512",
    "event": "subscribe",
    "arg": {
        "channel": "trades",
        "instId": "BTC-USDT"
    },
    "connId": "a4d3ae55"
}
```

### Failure Response Example

```
{
    "id": "1512",
    "event": "error",
    "code": "60012",
    "msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"trades\", \"instId\" : \"BTC-USD-191227\"}]}",
    "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event   
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> instId	String	Yes	Instrument ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

### Push Data Example

```
{
  "arg": {
    "channel": "trades",
    "instId": "BTC-USDT"
  },
  "data": [
    {
      "instId": "BTC-USDT",
      "tradeId": "130639474",
      "px": "42219.9",
      "sz": "0.12060306",
      "side": "buy",
      "ts": "1630048897897",
      "count": "3",
      "source": "0",
      "seqId": 1234
    }
  ]
}
```

### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> instId	String	Instrument ID
data	Array of objects	Subscribed data
> instId	String	Instrument ID, e.g. 
> tradeId	String	The last trade ID in the trades aggregation
> px	String	Trade price
> sz	String	Trade quantity For spot trading, the unit is base currency For    , the unit is contract.
> side	String	Trade side of taker 

Parameter	Type	Description
		<code>sell</code>
> ts	String	Filled time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> count	String	The count of trades aggregated
> source	String	Order source <code>0</code> : normal orders
> seqId	Integer	Sequence ID of the current message.

#### Aggregation function description:

1. The system will send only one message per taker order, per filled price. The `count` field will be used to represent the number of aggregated matches.
2. The `tradId` field in the message becomes the last trade ID in the aggregation.
3. When the `count` = 1, it means the taker order matches only one maker order with the specific price.
4. When the `count` > 1, it means the taker order matches multiple maker orders with the same price. For example, if `tradId` = 123 and `count` = 3, it means the message aggregates the trades of `tradId` = 123, 122, and 121. Maker side has filled multiple orders.
5. Users can use this information to compare with data from the `trades-all` channel.
6. Order book and the aggregated trades data are still published sequentially.

The seqId may be the same for different trade updates that occur at the same time.

## WS / ALL TRADES CHANNEL

Retrieve the recent trades data. Data will be pushed whenever there is a trade. Every update contain only one trade.

### URL PATH

/ws/v5/business

#### Request Example

```
import asyncio

from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/business")
    await ws.start()
    args = [
        {
            "channel": "trades-all",
            "instId": "BTC-USDT"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>trades-all</code>
> instId	String	Yes	Instrument ID

#### Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "trades-all",
    "instId": "BTC-USDT"
  },
  "connId": "a4d3ae55"
}
```

#### Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\"op\": \"subscribe\", \"args\": [{\"channel\": \"trades-all\", \"instId\": \"BTC-USD-191227\"}]}",
  "connId": "a4d3ae55"
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event <code>subscribe</code> <code>unsubscribe</code> <code>error</code>
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> instId	String	Yes	Instrument ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

#### Push Data Example

```
{
  "arg": {
    "channel": "trades-all",
    "instId": "BTC-USDT"
  },
  "data": [
    {
      "instId": "BTC-USDT",
      "tradeId": "130639474",
      "px": "42219.9",
      "sz": "0.12060306",
      "side": "buy",
      "source": "0",
      "ts": "1630048897897"
    }
  ]
}
```

## PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> instId	String	Instrument ID
data	Array of objects	Subscribed data
> instId	String	Instrument ID, e.g. <code>BTC-USDT</code>
> tradeId	String	Trade ID
> px	String	Trade price
> sz	String	Trade quantity For spot trading, the unit is base currency For <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code> , the unit is contract.
> side	String	Trade direction <code>buy</code> <code>sell</code>
> source	String	Order source <code>0</code> : normal
> ts	String	Filled time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## WS / ORDER BOOK CHANNEL

Retrieve order book data. Best ask price may be lower than the best bid price during the pre-open period.

Use `books` for 400 depth levels, `books5` for 5 depth levels, `bbo-tbt` tick-by-tick 1 depth level, `books50-12-tbt` tick-by-tick 50 depth levels, and `books-12-tbt` for tick-by-tick 400 depth levels.

- `books`: 400 depth levels will be pushed in the initial full snapshot. Incremental data will be pushed every 100 ms for the changes in the order book during that period of time.
- `books5`: 5 depth levels snapshot will be pushed in the initial push. Snapshot data will be pushed every 100 ms when there are changes in the 5 depth levels snapshot.
- `bbo-tbt`: 1 depth level snapshot will be pushed in the initial push. Snapshot data will be pushed every 10 ms when there are changes in the 1 depth level snapshot.
- `books-12-tbt`: 400 depth levels will be pushed in the initial full snapshot. Incremental data will be pushed every 10 ms for the changes in the order book during that period of time.
- `books50-12-tbt`: 50 depth levels will be pushed in the initial full snapshot. Incremental data will be pushed every 10 ms for the changes in the order book during that period of time.
- The push sequence for order book channels within the same connection and trading symbols is fixed as: `bbo-tbt` -> `books-12-tbt` -> `books50-12-tbt` -> `books` -> `books5`.

- Users can not simultaneously subscribe to `books-12-tbt` and `books50-12-tbt/books` channels for the same trading symbol.
  - For more details, please refer to the changelog 2024-07-17

Only API users who are VIP5 and above in trading fee tier are allowed to subscribe to "books-l2-tbt" 400 depth channels

Only API users who are VIP4 and above in trading fee tier are allowed to subscribe to "books50-l2-tbt" 50 depth channels

Identity verification refers to [Login](#)

#### URL PATH

`/ws/v5/public`

Request Example

```
import asyncio

from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/public")
    await ws.start()
    args = [
        {
            "channel": "books",
            "instId": "BTC-USDT"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>books</code> <code>books5</code> <code>bbo-tbt</code> <code>books50-12-tbt</code> <code>books-12-tbt</code>
> instId	String	Yes	Instrument ID

Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "books",
    "instId": "BTC-USDT"
  },
  "connId": "a4d3ae55"
}
```

### Failure example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \\"subscribe\\", \\"args\\": [{ \\"channel\\": \\"books\\", \\"instId\\": \\"BTC-USDT-191227\\\"}]}",
  "connId": "a4d3ae55"
}
```

### RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event   
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> instId	String	Yes	Instrument ID
msg	String	No	Error message
code	String	No	Error code
connId	String	Yes	WebSocket connection ID

### Push Data Example: Full Snapshot

```
{
  "arg": {
    "channel": "books",
    "instId": "BTC-USDT"
  },
  "action": "snapshot",
  "data": [
    {
      "asks": [
        ["8476.98", "415", "0", "13"],
        ["8477", "7", "0", "2"],
        ["8477.34", "85", "0", "1"],
        ["8477.56", "1", "0", "1"],
        ["8505.84", "8", "0", "1"],
        ["8506.37", "85", "0", "1"],
        ["8506.49", "2", "0", "1"],
        ["8506.96", "100", "0", "2"]
      ],
      "bids": [
        ["8476.97", "256", "0", "12"],
        ["8475.55", "101", "0", "1"],
        ["8475.54", "100", "0", "1"],
        ["8475.3", "1", "0", "1"],
        ["8447.32", "6", "0", "1"]
      ]
    }
  ]
}
```

```

        [ "8447.02", "246", "0", "1" ],
        [ "8446.83", "24", "0", "1" ],
        [ "8446", "95", "0", "3" ]
    ],
    "ts": "1597026383085",
    "checksum": -855196043,
    "prevSeqId": -1,
    "seqId": 123456
}
]
}

```

### Push Data Example: Incremental Data

```

{
    "arg": {
        "channel": "books",
        "instId": "BTC-USDT"
    },
    "action": "update",
    "data": [
        {
            "asks": [
                [ "8476.98", "415", "0", "13" ],
                [ "8477", "7", "0", "2" ],
                [ "8477.34", "85", "0", "1" ],
                [ "8477.56", "1", "0", "1" ],
                [ "8505.84", "8", "0", "1" ],
                [ "8506.37", "85", "0", "1" ],
                [ "8506.49", "2", "0", "1" ],
                [ "8506.96", "100", "0", "2" ]
            ],
            "bids": [
                [ "8476.97", "256", "0", "12" ],
                [ "8475.55", "101", "0", "1" ],
                [ "8475.54", "100", "0", "1" ],
                [ "8475.3", "1", "0", "1" ],
                [ "8447.32", "6", "0", "1" ],
                [ "8447.02", "246", "0", "1" ],
                [ "8446.83", "24", "0", "1" ],
                [ "8446", "95", "0", "3" ]
            ],
            "ts": "1597026383085",
            "checksum": -855196043,
            "prevSeqId": 123456,
            "seqId": 123457
        }
    ]
}

```

### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> instId	String	Instrument ID
action	String	Push data action, incremental data or full snapshot. <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">snapshot</div> : full <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">update</div> : incremental
data	Array of objects	Subscribed data
> asks	Array of Arrays	Order book on sell side
> bids	Array of Arrays	Order book on buy side
> ts	String	Order book generation time, Unix timestamp format in milliseconds, e.g. <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">1597026383085</div>

Parameter	Type	Description
> checksum	Integer	Checksum, implementation details below
> prevSeqId	Integer	Sequence ID of the last sent message. Only applicable to <code>books</code> , <code>books-12-tbt</code> , <code>books50-12-tbt</code>
> seqId	Integer	Sequence ID of the current message, implementation details below

An example of the array of asks and bids values: `["411.8", "10", "0", "4"]`

- `"411.8"` is the depth price
- `"10"` is the quantity at the price (number of contracts for derivatives, quantity in base currency for Spot and Spot Margin)
- `"0"` is part of a deprecated feature and it is always `"0"`
- `"4"` is the number of orders at the price.

If you need to subscribe to many 50 or 400 depth level channels, it is recommended to subscribe through multiple websocket connections, with each of less than 30 channels.

The order book data will be updated around once a second during the call auction.

## SEQUENCE ID

`seqId` is the sequence ID of the market data published. The set of sequence ID received by users is the same if users are connecting to the same channel through multiple websocket connections. Each `instId` has an unique set of sequence ID. Users can use `prevSeqId` and `seqId` to build the message sequencing for incremental order book updates. Generally the value of `seqId` is larger than `prevSeqId`. The `prevSeqId` in the new message matches with `seqId` of the previous message. The smallest possible sequence ID value is 0, except in snapshot messages where the `prevSeqId` is always -1.

Exceptions:

1. If there are no updates to the depth for an extended period(Around 60 seconds), for the channel that always updates snapshot data, OKX will send the latest snapshot, for the channel that has incremental data, OKX will send a message with `'asks': [], 'bids': []` to inform users that the connection is still active. `seqId` is the same as the last sent message and `prevSeqId` equals to `seqId`. 2. The sequence number may be reset due to maintenance, and in this case, users will receive an incremental message with `seqId` smaller than `prevSeqId`. However, subsequent messages will follow the regular sequencing rule.

## EXAMPLE

1. Snapshot message: `prevSeqId = -1, seqId = 10`
2. Incremental message 1 (normal update): `prevSeqId = 10, seqId = 15`
3. Incremental message 2 (no update): `prevSeqId = 15, seqId = 15`
4. Incremental message 3 (sequence reset): `prevSeqId = 15, seqId = 3`
5. Incremental message 4 (normal update): `prevSeqId = 3, seqId = 5`

## CHECKSUM

This mechanism can assist users in checking the accuracy of depth data.

## MERGING INCREMENTAL DATA INTO FULL DATA

After subscribing to the incremental load push (such as `books` 400 levels) of Order Book Channel, users first receive the initial full load of market depth. After the incremental load is subsequently received, update the local full load.

1. If there is the same price, compare the size. If the size is 0, delete this depth data. If the size changes, replace the original data.
2. If there is no same price, sort by price (bid in descending order, ask in ascending order), and insert the depth information into the full load.

## CALCULATE CHECKSUM

Use the first 25 bids and asks in the full load to form a string (where a colon connects the price and size in an ask or a bid), and then calculate the CRC32 value (32-bit signed integer).

Calculate Checksum

1. More than 25 levels of bid and ask

A full load of market depth (only 2 levels of data are shown here, while 25 levels of data should actually be intercepted):

```
{  
  "bids": [  
    ["3366.1", "7", "0", "3"],  
    ["3366", "6", "3", "4"]  
  ],  
  "asks": [  
    ["3366.8", "9", "10", "3"],  
    ["3368", "8", "3", "4"]  
  ]  
}
```

Check string:

"3366.1:7:3366.8:9:3366:6:3368:8"

2. Less than 25 levels of bid or ask

A full load of market depth:

```
{  
  "bids": [  
    ["3366.1", "7", "0", "3"]  
  ],  
  "asks": [  
    ["3366.8", "9", "10", "3"],  
    ["3368", "8", "3", "4"],  
    ["3372", "8", "3", "4"]  
  ]  
}
```

Check string:

"3366.1:7:3366.8:9:3368:8:3372:8"

1. When the bid and ask depth data exceeds 25 levels, each of them will intercept 25 levels of data, and the string to be checked is queued in a way that the bid and ask depth data are alternately arranged.

Such as: `bid[price:size]:ask[price:size]:bid[price:size]:ask[price:size]...`

2. When the bid or ask depth data is less than 25 levels, the missing depth data will be ignored.

Such as: `bid[price:size]:ask[price:size]:asks[price:size]:asks[price:size]...`

Push Data Example of bbo-tbt channel

```
{  
  "arg": {  
    "channel": "bbo-tbt",  
    "instId": "BCH-USDT-SWAP"  
  },  
  "data": [  
    {  
      "asks": [  
        [  
          "111.06", "55154", "0", "2"  
        ]  
      ],  
      "bids": [  
        [  
          "111.05", "57745", "0", "2"  
        ]  
      ],  
      "ts": "1670324386802",  
      "seqId": 363996337  
    }  
  ]  
}
```

Push Data Example of books5 channel

```
{
  "arg": {
    "channel": "books5",
    "instId": "BCH-USDT-SWAP"
  },
  "data": [
    {
      "asks": [
        ["111.06", "55154", "0", "2"],
        ["111.07", "53276", "0", "2"],
        ["111.08", "72435", "0", "2"],
        ["111.09", "70312", "0", "2"],
        ["111.1", "67272", "0", "2"]]
      ],
      "bids": [
        ["111.05", "57745", "0", "2"],
        ["111.04", "57109", "0", "2"],
        ["111.03", "69563", "0", "2"],
        ["111.02", "71248", "0", "2"],
        ["111.01", "65090", "0", "2"]]
      ],
      "instId": "BCH-USDT-SWAP",
      "ts": "1670324386802",
      "seqId": 363996337
    }
  ]
}
```

## WS / OPTION TRADES CHANNEL

Retrieve the recent trades data. Data will be pushed whenever there is a trade. Every update contain only one trade.

### URL PATH

/ws/v5/public

#### Request Example

```
import asyncio

from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/public")
    await ws.start()
    args = [
        {"channel": "option-trades",
         "instType": "OPTION",
         "instFamily": "BTC-USD"
        }]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.

Parameter	Type	Required	Description
op	String	Yes	<a href="#">subscribe</a> <a href="#">unsubscribe</a>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <a href="#">option-trades</a>
> instType	String	Yes	Instrument type, <a href="#">OPTION</a>
> instId	String	Conditional	Instrument ID, e.g. BTC-USD-221230-4000-C, Either <a href="#">instId</a> or <a href="#">instFamily</a> is required. If both are passed, <a href="#">instId</a> will be used.
> instFamily	String	Conditional	Instrument family, e.g. BTC-USD

#### Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "option-trades",
    "instType": "OPTION",
    "instFamily": "BTC-USD"
  },
  "connId": "a4d3ae55"
}
```

#### Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \\"subscribe\\", \\"args\\": [{ \\"channel\\": \\"option-trades\\"]}]}",
  "connId": "a4d3ae55"
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	<a href="#">subscribe</a> <a href="#">unsubscribe</a> <a href="#">error</a>
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name <a href="#">status</a>
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

#### Push Data Example

```
{
  "arg": {
    "channel": "option-trades",
```

```

        "instType": "OPTION",
        "instFamily": "BTC-USD"
    },
    "data": [
        {
            "fillVol": "0.5066007836914062",
            "fwdPx": "16469.69928595038",
            "idxPx": "16537.2",
            "instFamily": "BTC-USD",
            "instId": "BTC-USD-230224-18000-C",
            "markPx": "0.04690107010619562",
            "optType": "C",
            "px": "0.045",
            "side": "sell",
            "sz": "2",
            "tradeId": "38",
            "ts": "1672286551080"
        }
    ]
}

```

#### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
data	Array of objects	Subscribed data
> instId	String	Instrument ID
> instFamily	String	Instrument family
> tradeId	String	Trade ID
> px	String	Trade price
> sz	String	Trade quantity. The unit is contract.
> side	String	Trade side  
> optType	String	Option type, C: Call P: Put
> fillVol	String	Implied volatility while trading (Correspond to trade price)
> fwdPx	String	Forward price while trading
> idxPx	String	Index price while trading
> markPx	String	Mark price while trading
> ts	String	Trade time, Unix timestamp format in milliseconds, e.g. <a href="#">1597026383085</a> .

#### WS / CALL AUCTION DETAILS CHANNEL

Retrieve call auction details.

#### URL PATH

/ws/v5/public

#### Request Example

```
import asyncio
```

```

from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/public")
    await ws.start()
    args = [
        {"channel": "call-auction-details",
        "instId": "ONDO-USDC"
    }]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>call-auction-details</code>
> instId	String	Yes	Instrument ID

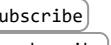
## Successful Response Example

```
{
    "id": "1512",
    "event": "subscribe",
    "arg": {
        "channel": "call-auction-details",
        "instId": "ONDO-USDC"
    },
    "connId": "a4d3ae55"
}
```

## Failure Response Example

```
{
    "id": "1512",
    "event": "error",
    "code": "60012",
    "msg": "Invalid request: {\\"op\\": \"subscribe\", \\"argss\\": [{ \\"channel\\\" : \"call-auction-details\", \\"instId\\\" : \"BTC-USD-191227\"}]}",
    "connId": "a4d3ae55"
}
```

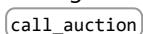
## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event   
arg	Object	No	Subscribed channel
> channel	String	yes	channel name
> instId	String	Yes	Instrument ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

### Push Data Example

```
{
  "arg": {
    "channel": "call-auction-details",
    "instId": "ONDO-USDC"
  },
  "data": [
    {
      "instId": "ONDO-USDC",
      "unmatchedSz": "9988764",
      "eqPx": "0.6",
      "matchedSz": "44978",
      "state": "continuous_trading",
      "auctionEndTime": "1726542000000",
      "ts": "172654200007"
    }
  ]
}
```

### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> instId	String	Instrument ID
data	Array of objects	Subscribed data
> instId	String	Instrument ID
> eqPx	String	Equilibrium price
> matchedSz	String	Matched size for both buy and sell The unit is in base currency
> unmatchedSz	String	Unmatched size
> auctionEndTime	String	Call auction end time. Unix timestamp in milliseconds.
> state	String	Trading state of the symbol 

Parameter	Type	Description
		continuous_trading
> ts	String	Data generation time. Unix timestamp in milliseconds.

During call auction, users can get the updates of equilibrium price, matched size, unmatched size, and auction end time. The data will be updated around once a second. When call auction ends, this channel will push the last message, returning the actual open price, matched size, and unmatched size, with trading state as `continuous\_trading`.

## SBE Market Data

### OVERVIEW

OKX supports Simple Binary Encoding (SBE) for data returned from the following WebSocket channels:

- WS / Trades channel: `trades`
- WS / Order book channel: `bbo-tbt` and `books-12-tbt`

### XML SCHEMA

The SBE XML schema is now available for download:

[Download XML Schema](#)

### GENERAL INFORMATION

- Access is restricted to API users with **VIP5** trading fee tier or higher for production, and **VIP1** or higher for demo trading.
- SBE channels will use a new WebSocket URL.  
Live trading: `wss://ws.okx.com:8443/ws/v5/public-sbe`  
Demo trading: `wss://wspap.okx.com:8443/ws/v5/public-sbe`
- Both JSON and SBE format data will be available on the same connection, distinguishable by WebSocket frame type. opcode `1` indicates JSON, while opcode `2` indicates SBE.
- Prices and quantities will be encoded as exponential decimals, using a signed integer mantissa and signed exponent. For example, a mantissa of 123456 and a exponent of -4 represents 12.3456 (actual value = mantissa \* 10 ^ exponent).
- The SBE protocol will use `instIdCode`, an integer will be provided by Get instruments to represent trading instruments. Users must map `instIdCode` to `instId`, noting that `instIdCode` will change if a trading symbol is relisted.
- `tsUs` and `outTime` come from different servers, so their relative order is not guaranteed.
- `tsUs` is in microseconds format but only accurate to milliseconds. The microseconds-format timestamp is obtained by appending 000 to the millisecond timestamp. For example, if the millisecond timestamp is 1726233600001, the related microseconds-format timestamp (tsUs) will be 1726233600001000.

### INTEGRATION INFORMATION

- To log in, transmit your API key and signature in the WebSocket connection header.
  - The connection requests must contain the following headers:
    - `OK-ACCESS-KEY` The API key as a String.
    - `OK-ACCESS-SIGN` The Base64-encoded signature.
    - `OK-ACCESS-TIMESTAMP` Unix Epoch time in seconds. e.g : `1751335333`
    - `OK-ACCESS-PASSPHRASE` The passphrase you specified when creating the API key.
  - The `OK-ACCESS-SIGN` header is generated as follows:
    - Create a pre-hash string of timestamp + method + requestPath
    - Prepare the SecretKey.
    - Sign the pre-hash string with the SecretKey using the HMAC SHA256.
    - Encode the signature in the Base64 format. Example: `sign=CryptoJS.enc.Base64.stringify(CryptoJS.HmacSHA256(timestamp + 'GET' + '/users/self/verify', SecretKey))`
    - Example of timestamp: `const timestamp = '' + Date.now() / 1,000`, e.g. `1704876947`
    - Method: always 'GET'.
    - RequestPath : always '/users/self/verify'
  - The response HTTP code of `101` indicates the successful login.
  - The response HTTP code `401`, along with an error message in the response body, indicates a failed login. The error message will be in JSON format.
- Subscription request must be sent in JSON format. The response will also be in JSON format, and can be identified by opcode `1`.

- The protocol is similar to existing JSON-formatted subscription requests/response.
- The difference is that `instIdCode` should be used instead of `instId`.
- The notice event is supported in JSON format:
- The WebSocket server will send a ping frame with opcode `9` every 20 seconds after receiving a pong frame.
  - If the WebSocket server does not receive a pong frame back from the connection within 60 seconds, the connection will be disconnected.
  - Upon receiving a ping, you must respond with a pong frame using opcode `10`, along with a copy of the ping's payload as soon as possible (payload will be a random numerical text like 11446744073709551615).
  - Unsolicited pong frames are permitted but will not prevent disconnection. It is advisable that the payload for these pong frames be empty.
- For `trades`, `bbo-tbt` and `books-12-tbt` channels, data will be returned in binary format and can be identified by opcode `2`, distinguishable by template ID. Key differences compared to existing JSON-formatted connections include:
  - For the `trades` channel, the `seqId` will be returned.
  - For the `bbo-tbt` channel, it usually provides real-time data, but under system overload, data loss can occur, varying by different connection.
  - For the `books-12-tbt`:
    - When prices and quantities decimals change, an exponent update (template ID: 1002) will occur with previous sequence ID and sequence ID, identifiable by template ID. This must be processed to maintain the sequence ID consistency.
    - The checksum will no longer be included.
    - There will be no initial order book snapshot after subscription. Instead, OKX will provide a REST API endpoint that returns SBE binary data for the initial 400 levels snapshot. This endpoint will buffer requests and return data only when a new snapshot is generated, approximately every 500 ms.
- The relationship between the channel and event is not one-to-one. The `books-12-tbt` contains two types of events. The mapping is outlined below.

Channel	XML Template ID and message name
bbo-tbt	1000: BboTbtChannelEvent
books-12-tbt	1001: BooksL2TbtChannelEvent 1002: BooksL2TbtExponentUpdateEvent
books-12-tbt-elp (It is not enabled)	1003: BooksL2TbtElpChannelEvent 1004: BooksL2TbtElpExponentUpdateEvent
trades	1005: TradesChannelEvent

- How to manage a local order book correctly

1. Open a SBE WebSocket connection and subscribe to `books-12-tbt`.
2. Buffer the events received from the stream. Record the `prevSeqId` of the first event you received.

Note: For template ID 1002, the event is an exponent update, containing only exponent update information without ask or bid data. For template ID 1001, the data includes both asks and bids.
3. Get a depth snapshot from `/books-sbe`, e.g. <https://www.okx.com/api/v5/market/books-sbe?instIdCode=12345&source=0>
4. If the `seqId` from the snapshot is strictly less than the `prevSeqId` from step 2, go back to step 3.
5. In the buffered events, discard any event where stream `seqId` is  $\leq$  snapshot `seqId` of the snapshot.
6. The first buffered event should satisfy the condition: stream `prevSeqId`  $\leq$  snapshot `seqId`  $<$  stream `seqId`.
7. Set your local order book to the snapshot. Its sequence ID is snapshot `seqId`.
8. Apply the update procedure below to all buffered events, and then to all subsequent events received.
  - If the template ID is 1002 (BooksL2TbtExponentUpdateEvent), only update the exponents without bid and ask data. If the template ID is 1001 (BooksL2TbtChannelEvent), follow the process outlined below.
  - For each price level in bids and asks, set the new quantity in the order book:
    - If the price level does not exist in the order book, insert it with new quantity.
    - If the quantity is zero, remove the price level from the order book.
  - Set the order book sequence ID to the latest sequence ID (`seqId`) in the processed event.

Note: Not all snapshot `seqId` will appear in the `books-12-tbt` channels.

- Sequence ID

`seqId` is the sequence ID of the market data published. The set of sequence ID received by users is the same if users are connecting to the same channel through multiple websocket connections. Each `instIdCode` has an unique set of sequence ID. Users can use `prevSeqId` and `seqId` to build the message sequencing for incremental order book updates. Generally the value of `seqId` is larger than `prevSeqId`. The `prevSeqId` in the new message

matches with `seqId` of the previous message. The smallest possible sequence ID value is 0, except in snapshot messages where the `prevSeqId` is always -1.

## Exceptions:

1. If there are no updates to the depth for an extended period(Around 60 seconds), for the channel that always updates snapshot data, OKX will send the latest snapshot, for the channel that has incremental data, OKX will send a message with `numInGroup`: 0 to inform users that the connection is still active. `seqId` is the same as the last sent message and `prevSeqId` equals to `seqId`.
2. The sequence number may be reset due to maintenance, and in this case, users will receive an incremental message with `seqId` smaller than `prevSeqId`. However, subsequent messages will follow the regular sequencing rule.

## EXAMPLE

1. Incremental message 1 (normal update): `prevSeqId` = 10, `seqId` = 15
2. Incremental message 2 (no update): `prevSeqId` = 15, `seqId` = 15
3. Incremental message 3 (sequence reset): `prevSeqId` = 15, `seqId` = 3
4. Incremental message 4 (normal update): `prevSeqId` = 3, `seqId` = 5

## SBE ORDER BOOK

It is a public endpoint, returning SBE binary data for the initial 400 levels snapshot. This endpoint will buffer requests and return data only when a new snapshot is generated, approximately every 500 ms.

Note: If the request fails, the error message will be provided in JSON format.

For the HTTP request header, it doesn't need to be set to `application/sbe`; however, the response header will be `Content-Type`: `application/sbe` if the request is successful, and `Content-Type`: `application/json` if the request fails.

## RATE LIMIT: 10 REQUESTS PER 10 SECONDS

## RATE LIMIT RULE: IP + INSTIDCODE

## HTTP REQUEST

```
GET /api/v5/market/books-sbe
```

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instIdCode	Integer	Yes	Instrument ID code
source	Integer	Yes	The source of order book. 0: normal

Response Example

## RESPONSE PARAMETERS

Please refer to the `SnapshotDepthResponseEvent` with ID `1006` in the XML schema.

## NEW ERROR CODE

Error Code	HTTP Status	Error Message
60034	401	Only users who are {0} and above in trading fee tier are allowed to use this URL.

## UPGRADE

- In general, only compatible upgrades are made, such as adding a new field. In these cases, the XML schema ID remains unchanged, while the schema version is incremented.
- If a breaking change is needed, a new XML schema with a new schema ID will be released at least 1–2 months in advance. Before the end of the transition period, you'll need to support both the old and new schemas, based on their schema ID and version.

# Block Trading

## Block Trading Workflow

A block trade is a **large sized, privately negotiated** transaction that allows traders to execute spot, perpetuals, futures, options and a combination of instruments (multi leg) which are traded **outside the order book** and at a **mutually agreed price** between the counter-parties. Once the transaction economics have been agreed upon, it will be submitted to OKX to be seamlessly margined, cleared and executed.

### Basic Concepts

1. **RFQs** - Request for Quote sent by the Taker to Maker(s). It captures the quantity, instrument or multi instrument strategy that a Taker wants to trade.
2. **Quotes** - Quotes are created by the *Maker* in response to a requested RFQ.
3. **Trades** - Trades occur when the *Taker* successfully *executes* upon a makers quote to an RFQ.

### High Level Workflow

To trade as either Taker or Maker, users need to deposit at least 100,000 USD into their trading account. In addition, to become a Maker, Please complete the form to access block trading.

1. Taker creates an RFQ and selects which counterparties to broadcast the RFQ to.
2. Multiple Maker(s) send a two way quote as a response to the RFQ.
3. Taker chooses to execute upon the best quote and the trade is sent to OKX for clearing & settlement.
4. Taker & Maker receive confirmation of the trade's execution.
5. Trade economics are published to market feed. (minus counterparty info)

**Self-trade Prevention** Users cannot send RFQ requests to themselves.

### Taker's Perspective

1. Taker creates an RFQ using `POST /api/v5/rfq/create-rfq`. Taker can pull available instruments via `GET /api/v5/public/instruments` and available counterparties from `GET /api/v5/rfq/counterparties`.
2. Taker can cancel an RFQ anytime until it becomes inactive with `POST /api/v5/rfq/cancel-rfq`.
3. Maker, who is a requested counterparty to the RFQ, and is notified over the `rfqs` WebSocket channel, can provide a Quote to the RFQ.
4. Taker, who will be notified of quotes from the `quotes` WebSocket channel, can execute upon the best Quote with `POST /api/v5/rfq/execute-quote`.
5. Taker will receive confirmation of the trade's successful execution on the `struc-block-trades` and `rfqs` WebSocket channel.
6. Taker will also receive confirmation of the trade being completed on the `public-struc-block-trades` WebSocket channel as well as all other block trades on OKX.

### Maker's Perspective

1. Maker is notified about a new RFQ who they are a counterparty to, on the `rfqs` WebSocket channel.
2. Maker can create a one way or two way Quote using `POST /api/v5/rfq/create-quote`.
3. Maker can cancel an existing quote anytime until it becomes inactive with `POST /api/v5/rfq/cancel-quote`.
4. Taker chooses to execute upon an available Quote.
5. Maker will receive updates of their Quote from the `quotes` WebSocket channel.
6. Maker will receive confirmation of the successful execution of their Quote from the `struc-block-trades` and `quotes` WebSocket channel.
7. Maker will receive confirmation of the trade being completed on the `public-struc-block-trades` WebSocket channel as well as all other block trades on OKX.

## REST API

Block trading is not supported under spot mode.

### GET COUNTERPARTIES

Retrieves the list of counterparties that the user is permitted to trade with.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

## RATE LIMIT RULE: USER ID

### PERMISSION: READ

#### HTTP REQUEST

```
GET /api/v5/rfq/counterparties
```

#### Request Example

```
import okx.BlockTrading as BlockTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

blockTradingAPI = BlockTrading.BlockTradingAPI(apikey, secretkey, passphrase, False, flag)

# Get counterparties
result = blockTradingAPI.counterparties()
print(result)
```

#### REQUEST PARAMETERS

None

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "traderName" : "Satoshi Nakamoto",
      "traderCode" : "SATOSHI",
      "type" : ""
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
traderName	String	The long formative username of trader or entity on the platform.
traderCode	String	A unique identifier of maker which will be publicly visible on the platform. All RFQ and Quote endpoints will use this as the unique counterparty identifier.
type	String	The counterparty type. <span style="border: 1px solid #ccc; padding: 0 2px;">LP</span> refers to API connected auto market makers.

#### CREATE RFQ

Creates a new RFQ

Please select trading bot "WAGMI" as the counterparty when submitting RFQs in demo trading.

Prices provided on RFQs by the trading bot are for reference only.

To learn more, please visit Support center > FAQ > Trading > Liquid marketplace > Demo trading

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS; 80 REQUESTS PER 12 HOURS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

## HTTP REQUEST

POST /api/v5/rfq/create-rfq

### Request Example

```
import okx.BlockTrading as BlockTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

blockTradingAPI = BlockTrading.BlockTradingAPI(apikey, secretkey, passphrase, False, flag)

# Create RFQ
result = blockTradingAPI.create_rfq(
    anonymous=True,
    counterparties=[
        "Trader1",
        "Trader2"
    ],
    clRfqId= "rfq01",
    legs=[
        {
            "sz": "25",
            "side": "buy",
            "posSide": "long",
            "tdMode": "cross",
            "ccy": "USDT",
            "instId": "BTC-USD-221208-100000-C"
        },
        {
            "sz": "150",
            "side": "buy",
            "posSide": "long",
            "tdMode": "cross",
            "ccy": "USDT",
            "instId": "ETH-USDT",
            "tgtCcy": "base_ccy"
        }
    ]
)
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
counterparties	Array of strings	Yes	The trader code(s) of the counterparties who receive the RFQ. Can be found via /api/v5/rfq/counterparties/
anonymous	Boolean	No	Submit RFQ on a disclosed or anonymous basis. Valid values are <code>true</code> or <code>false</code> . If not specified, the default value is <code>false</code> . When anonymous = true, the taker's identify is not disclosed to maker even after trade execution.
clRfqId	String	No	Client-supplied RFQ ID. A combination of case-sensitive alpha-numeric, all numbers, or all letters of up to 32 characters.
tag	String	No	RFQ tag. The block trade associated with the RFQ will have the same tag. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters.
allowPartialExecution	Boolean	No	Whether the RFQ can be partially filled provided that the shape of legs stays the same. Valid values are <code>true</code> or <code>false</code> . <code>false</code> by default.

Parameter	Type	Required	Description
legs	Array of objects	Yes	An Array of objects containing each leg of the RFQ. Maximum 15 legs can be placed per request
> instId	String	Yes	The Instrument ID of each leg. Example : "BTC-USDT-SWAP"
> tdMode	String	No	<p>Trade mode</p> <p>Margin mode: <code>cross</code> <code>isolated</code></p> <p>Non-Margin mode: <code>cash</code>.</p> <p>If not provided, tdMode will inherit default values set by the system shown below:</p> <p>Futures mode &amp; SPOT: <code>cash</code></p> <p>Buy options in Futures mode and Multi-currency Margin: <code>isolated</code></p> <p>Other cases: <code>cross</code></p>
> ccy	String	No	<p>Margin currency.</p> <p>Only applicable to <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code>. The parameter will be ignored in other scenarios.</p>
> sz	String	Yes	The size of each leg
> lmtPx	String	No	<p>Taker expected price for the RFQ</p> <p>If provided, RFQ trade will be automatically executed if the price from the quote is better than or equal to the price specified until the RFQ is canceled or expired.</p> <p>This field has to be provided for all legs to have the RFQ automatically executed, or leave empty for all legs, otherwise request will be rejected.</p> <p>The auto execution side depends on the leg side of the RFQ.</p> <p>For <code>SPOT/MARGIN/FUTURES/SWAP</code>, lmtPx will be in unit of the quote ccy.</p> <p>For <code>OPTION</code>, lmtPx will be in unit of settle ccy.</p> <p>The field will not be disclosed to counterparties.</p>
> side	String	Yes	The direction of each leg. Valid values can be <code>buy</code> or <code>sell</code> .
> posSide	String	No	<p>Position side.</p> <p>The default is <code>net</code> in the net mode. It can only be <code>long</code> or <code>short</code> in the long/short mode.</p> <p>If not specified, users in long/short mode always open new positions.</p> <p>Only applicable to <code>FUTURES/SWAP</code>.</p>
> tgtCcy	String	No	<p>Defines the unit of the "sz" attribute.</p> <p>Only applicable to instType = SPOT.</p> <p>The valid enumerations are <code>base_ccy</code> and <code>quote_ccy</code>. When not specified, this is equal to <code>base_ccy</code> by default.</p>
> tradeQuoteCcy	String	No	<p>The quote currency used for trading. Only applicable to SPOT.</p> <p>The default value is the quote currency of the instId, for example: for <code>BTC-USD</code>, the default is <code>USD</code>.</p>
acctAlloc	Array of objects	No	Account level allocation of the RFQ
> acct	String	Yes	The name of the allocated account of the RFQ.
> legs	Array of objects	Yes	The allocated legs of the account.
>> sz	String	Yes	The allocated size of each leg
>> instId	String	Yes	The Instrument ID of each leg. Example : "BTC-USDT-SWAP"
>> tdMode	String	No	Trade mode
>> ccy	String	No	Margin currency
>> posSide	String	No	Position side

## Response Example

```
{  
  "code": "0",  
  "msg": "",  
  "data": [  
    {  
      "cTime": "1611033737572",  
      "uTime": "1611033737572",  
      "traderCode": "SATOSHI",  
      "tag": "123456",  
      "rfqId": "22534",  
      "clRfqId": "rfq01",  
      "allowPartialExecution": false,  
      "state": "active",  
      "validUntil": "1611033857557",  
      "counterparties": [  
        "Trader1",  
        "Trader2"  
      ],  
      "legs": [  
        {  
          "instId": "BTC-USD-221208-100000-C",  
          "tdMode": "cross",  
          "ccy": "USDT",  
          "sz": "25",  
          "side": "buy",  
          "posSide": "long",  
          "tgtCcy": ""  
        },  
        {  
          "instId": "ETH-USDT",  
          "tdMode": "cross",  
          "ccy": "USDT",  
          "sz": "150",  
          "side": "buy",  
          "posSide": "long",  
          "tgtCcy": "base_ccy",  
          "tradeQuoteCcy": "USDT"  
        }  
      ]  
    }  
  ]  
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, 0 means success.
msg	String	The error message, not empty if the code is not 0.
data	Array of objects	Array of objects containing the results of the RFQ creation.
> cTime	String	The timestamp the RFQ was created. Unix timestamp format in milliseconds.
> uTime	String	The timestamp the RFQ was last updated. Unix timestamp format in milliseconds.
> state	String	The status of the RFQ. Valid values can be active, canceled, pending_fill, filled, expired, traded_away, failed. traded_away only applies to Maker
> counterparties	Array of strings	The list of counterparties traderCode the RFQ was broadcast to.
> validUntil	String	The timestamp the RFQ expires. Unix timestamp format in milliseconds. If all legs are options, the RFQ will expire after 10 minutes; otherwise, the RFQ will expire after 2 minutes.

Parameter	Type	Description
> clRfqlId	String	Client-supplied RFQ ID. This attribute is treated as client sensitive information. It will not be exposed to the Maker, only return empty string.
> tag	String	RFQ tag. The block trade associated with the RFQ will have the same tag.
> allowPartialExecution	Boolean	Whether the RFQ can be partially filled provided that the shape of legs stays the same.
> traderCode	String	A unique identifier of taker.
> rfqlId	String	The unique identifier of the RFQ generated by system.
> legs	Array of objects	An Array of objects containing each leg of the RFQ.
>> instId	String	Instrument ID, e.g. BTC-USDT-SWAP
>> tdMode	String	<p>Trade mode</p> <p>Margin mode: <code>cross</code> <code>isolated</code></p> <p>Non-Margin mode: <code>cash</code>.</p> <p>If not provided, tdMode will inherit default values set by the system shown below:</p> <p>Futures mode &amp; SPOT: <code>cash</code></p> <p>Buy options in Futures mode and Multi-currency Margin: <code>isolated</code></p> <p>Other cases: <code>cross</code></p>
>> ccy	String	<p>Margin currency.</p> <p>Only applicable to <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code>. The parameter will be ignored in other scenarios.</p>
>> sz	String	Size of the leg in contracts or spot.
>> side	String	The direction of the leg. Valid values can be buy or sell.
>> posSide	String	<p>Position side.</p> <p>The default is <code>net</code> in the net mode. If not specified, return "", which is equivalent to net.</p> <p>It can only be <code>long</code> or <code>short</code> in the long/short mode. If not specified, return "", which corresponds to the direction that opens new positions for the trade (buy =&gt; long, sell =&gt; short).</p> <p>Only applicable to FUTURES/SWAP.</p>
>> tgtCcy	String	<p>Defines the unit of the "sz" attribute.</p> <p>Only applicable to instType = SPOT.</p> <p>The valid enumerations are <code>base_ccy</code> and <code>quote_ccy</code>. When not specified this is equal to <code>base_ccy</code> by default.</p>
>> tradeQuoteCcy	String	<p>The quote currency used for trading. Only applicable to SPOT.</p> <p>The default value is the quote currency of the instId, for example: for <code>BTC-USD</code>, the default is <code>USD</code>.</p>
> groupId	String	<p>Group RFQ ID</p> <p>Only applicable to group RFQ, return "" for normal RFQ</p>
> acctAlloc	Array of objects	Account level allocation of the RFQ
>> acct	String	The name of the allocated account of the RFQ
>> sCode	String	The code of the event execution result, 0 means success
>> sMsg	String	Rejection message if the request is unsuccessful
>> legs	Array of objects	The allocated legs of the account
>>> instId	String	Instrument ID

Parameter	Type	Description
>>> sz	String	The calculated size of each leg of allocated account
>>> tdMode	String	Trade mode
>>> ccy	String	Margin currency
>>> posSide	String	Position side

### Group RFQ introduction

1. Only a master account can conduct group RFQ and the available scope of allocated subaccounts is its normal and managed subaccounts.
2. Users will pass in acctAlloc request parameter to indicate the details of group RFQ account allocation, account name, instrument ID, allocated size, etc. master account is also allowed and should be indicated as "0". For tdMode, ccy and posSide fields, they will inherit the system default value if you leave them empty.
3. Add groupId, acctAlloc as a new response parameter.
4. The upper limit of the number of allocated subaccounts is 10. You will receive error code 70516 if you exceed the upper limit.
5. For each symbol, the total size of RFQ legs in all accounts should be equal to its combined amount in the group RFQ. If not, you will receive error code 70514.
6. For each sub-account, the ratio of a leg's size to the group RFQ must be the same across all symbols. If not, you will receive error code 70515.

Here is an example:

1. Parent RFQ legs
  1. Symbol: BTC-USDT, size: 50, symbol: ETH-USDT, size: 100
2. Child RFQ legs, happy case
  1. Acct1: symbol: BTC-USDT, size: 30, symbol: ETH-USDT, size: 60 (ratio: 0.6)
  2. Acct2: symbol: BTC-USDT, size: 20, symbol: ETH-USDT, size: 40 (ratio: 0.4)
3. Child RFQ legs, bad case
  1. Acct1: symbol: BTC-USDT, size: 30, symbol: ETH-USDT, size: 50
  2. Acct2: symbol: BTC-USDT, size: 20, symbol: ETH-USDT, size: 50
  3. The total size is equal. But the ratio is not equal for different legs per subaccount.
7. For allowPartialExecution field, it will be ignored even though users pass it in. For a group RFQ, allowPartialExecution will always be true, since taker can not determine whether the RFQ can be partially or fully filled if any subaccount fails. Thus, makers should regard it as a RFQ that can be partially filled.
8. Group RFQ will not be created if any subaccount fails.

### CANCEL RFQ

Cancel an existing active RFQ that you have created previously.

#### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: USER ID

#### PERMISSION: TRADE

#### HTTP REQUEST

```
POST /api/v5/rfq/cancel-rfq
```

#### Request Example

```
import okx.BlockTrading as BlockTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

blockTradingAPI = BlockTrading.BlockTradingAPI(apikey, secretkey, passphrase, False, flag)

# Cancel RFQ
```

```

result = blockTradingAPI.cancel_rfq(
    rfqId="22535",
    clRfqId="rfq001"
)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
rfqId	String	Conditional	RFQ ID created .
clRfqId	String	Conditional	Client-supplied RFQ ID. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. Either rfqId or clRfqId is required. If both are passed, rfqId will be used.

## Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "rfqId": "22535",
      "clRfqId": "rfq001",
      "sCode": "0",
      "sMsg": ""
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, 0 means success.
msg	String	The error message, not empty if the code is not 0.
data	Array of objects	Array of objects containing the results
> rfqId	String	RFQ ID
> clRfqId	String	Client-supplied RFQ ID.
> sCode	String	The code of the event execution result, 0 means success.
> sMsg	String	Rejection message if the request is unsuccessful.

## CANCEL MULTIPLE RFQS

Cancel one or multiple active RFQ(s) in a single batch. Maximum 100 RFQ orders can be canceled per request.

### RATE LIMIT: 2 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: TRADE

### HTTP REQUEST

```
POST /api/v5/rfq/cancel-batch-rfqs
```

## Request Example

```

import okx.BlockTrading as BlockTrading

# API initialization

```

```

apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

blockTradingAPI = BlockTrading.BlockTradingAPI(apikey, secretkey, passphrase, False, flag)

# Cancel multiple RFQs
result = blockTradingAPI.cancel_batch_rfqs(
    rfqIds=[
        "2201",
        "2202",
        "2203"
    ],
    clRfqIds=[
        "r1",
        "r2",
        "r3"
    ],
)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
rfqIds	Array of strings	Conditional	RFQ IDs .
clRfqIds	Array of strings	Conditional	Client-supplied RFQ IDs. Either <code>rfqIds</code> or <code>clRfqIds</code> is required. If both attributes are sent, <code>rfqIds</code> will be used as primary identifier.

Success - All requested RFQs canceled

```
{
    "code": "0",
    "msg": "",
    "data": [
        {
            "rfqId": "2201",
            "clRfqId": "r1",
            "sCode": "0",
            "sMsg": ""
        },
        {
            "rfqId": "2202",
            "clRfqId": "r2",
            "sCode": "0",
            "sMsg": ""
        },
        {
            "rfqId": "2203",
            "clRfqId": "r3",
            "sCode": "0",
            "sMsg": ""
        }
    ]
}
```

Partial cancellation

```
{
    "code": "2",
    "msg": "Bulk operation partially",
    "data": [
        {
            "rfqId": "2201",
            "clRfqId": "r1",
            "sCode": "0",
            "sMsg": "Partial cancellation"
        }
    ]
}
```

```

        "sCode":"70000",
        "sMsg":"RFQ does not exist."
    },
    {
        "rfqId":"2202",
        "clRfqId":"r2",
        "sCode":"0",
        "sMsg":""
    },
    {
        "rfqId":"2203",
        "clRfqId":"r3",
        "sCode":"0",
        "sMsg":""
    }
]
}

```

## Failure example

```

{
    "code": "1",
    "msg": "Operation failed.",
    "data": [
        {
            "rfqId": "2201",
            "clRfqId": "r1",
            "sCode": "70000",
            "sMsg": "RFQ does not exist."
        },
        {
            "rfqId": "2202",
            "clRfqId": "r2",
            "sCode": "70000",
            "sMsg": "RFQ does not exist."
        },
        {
            "rfqId": "2203",
            "clRfqId": "r3",
            "sCode": "70000",
            "sMsg": "RFQ does not exist."
        }
    ]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, <code>0</code> means success.
msg	String	The error message, not empty if the code is not 0.
data	Array of objects	Array of objects containing the results
> rfqId	String	RFQ ID
> clRfqId	String	Client-supplied RFQ ID.
> sCode	String	The code of the event execution result, <code>0</code> means success.
> sMsg	String	Rejection message if the request is unsuccessful.

## CANCEL ALL RFQS

Cancels all active RFQs.

**RATE LIMIT: 2 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

## PERMISSION: TRADE

### HTTP REQUEST

```
POST /api/v5/rfq/cancel-all-rfqs
```

#### Request Example

```
import okx.BlockTrading as BlockTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

blockTradingAPI = BlockTrading.BlockTradingAPI(apikey, secretkey, passphrase, False, flag)

# Cancel all RFQs
result = blockTradingAPI.cancel_all_rfqs()
print(result)
```

### REQUEST PARAMETERS

None

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ts": "1697026383085"
    }
  ]
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, 0 means success.
msg	String	The error message, not empty if the code is not 0.
data	Array of objects	Array of objects containing the results
> ts	String	The timestamp of successful cancellation. Unix timestamp format in milliseconds, e.g. 1597026383085.

### EXECUTE QUOTE

Executes a Quote. It is only used by the creator of the RFQ

### RATE LIMIT: 2 REQUESTS PER 3 SECONDS

### RATE LIMIT RULE: USER ID

## PERMISSION: TRADE

### HTTP REQUEST

```
POST /api/v5/rfq/execute-quote
```

#### Request Example

```
import okx.BlockTrading as BlockTrading

# API initialization
apikey = "YOUR_API_KEY"
```

```

secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

blockTradingAPI = BlockTrading.BlockTradingAPI(apikey, secretkey, passphrase, False, flag)

# Execute quote
result = blockTradingAPI.execute_quote(
    rfqid="22540",
    quoteId="84073"
)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
rfqid	String	Yes	RFQ ID .
quotoid	String	Yes	Quote ID.
legs	Array of objects	No	An Array of objects containing the execution size of each leg of the RFQ. The ratio of the leg sizes needs to be the same as the RFQ. *Note: <code>tgtCcy</code> and <code>side</code> of each leg will be same as ones in the RFQ. <code>px</code> will be the same as the ones in the Quote.
> instId	String	Yes	The Instrument ID, for example: "BTC-USDT-SWAP".
> sz	String	Yes	The size of each leg

## Response Example

```

{
  "code":"0",
  "msg":"",
  "data":[
    {
      "blockTdId":"180184",
      "rfqid":"1419",
      "clRfqId":"r0001",
      "quoteId":"1046",
      "clQuoteId":"q0001",
      "tag":"123456",
      "tTraderCode":"Trader1",
      "mTraderCode":"Trader2",
      "cTime":"1649670009",
      "legs":[
        {
          "px":"0.1",
          "sz":"25",
          "instId":"BTC-USD-20220114-13250-C",
          "side":"sell",
          "fee":"-1.001",
          "feeCcy":"BTC",
          "tradeId":"10211"
        },
        {
          "px":"0.2",
          "sz":"25",
          "instId":"BTC-USDT",
          "side":"buy",
          "fee":"-1.001",
          "feeCcy":"BTC",
          "tradeId":"10212"
        }
      ]
    }
  ]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, <code>0</code> means success.
msg	String	The error message, not empty if the code is not 0.
data	Array of objects	Array of objects containing the results
> cTime	String	The execution time for the trade. Unix timestamp in milliseconds.
> rfqid	String	RFQ ID.
> clRfqId	String	Client-supplied RFQ ID. This attribute is treated as client sensitive information. It will not be exposed to the Maker, only return empty string.
> quoteld	String	Quote ID.
> clQuoteld	String	Client-supplied Quote ID. This attribute is treated as client sensitive information. It will not be exposed to the Taker, only return empty string.
> blockTdId	String	Block trade ID.
> tag	String	RFQ tag.
> tTraderCode	String	A unique identifier of the taker. Empty if the anonymous parameter of the RFQ is set to be <code>true</code> .
> mTraderCode	String	A unique identifier of the maker. Empty if the anonymous parameter of the Quote is set to be <code>true</code> .
> legs	Array of objects	Legs of trade
>> instId	String	Instrument ID, e.g. BTC-USDT-SWAP
>> px	String	The price the leg executed
>> sz	String	Size of the leg in contracts or spot.
>> side	String	The direction of the leg from the Takers perspective. Valid value can be buy or sell.
>> fee	String	Fee for the individual leg. Negative fee represents the user transaction fee charged by the platform. Positive fee represents rebate.
>> feeCcy	String	Fee currency. To be read in conjunction with fee
>> tradId	String	Last traded ID.
> acctAlloc	Array of objects	Account level allocation of the RFQ
>> acct	String	The name of the allocated account of the RFQ.
>> blockTdId	String	Block trade ID
>> sCode	String	The code of the event execution result, 0 means success
>> sMsg	String	Rejection message if the request is unsuccessful
>> legs	Array of objects	The allocated legs of the account.
>>> instId	String	The Instrument ID of each leg. Example : "BTC-USDT-SWAP"

Parameter	Type	Description
>>> sz	String	The size of each account leg is filled.
>>> fee	String	The fee of each account level leg
>>> feeCcy	String	Fee currency. To be read in conjunction with fee
>>> tradeld	String	Last traded ID of each account leg

### Group RFQ introduction

1. Takers are not allowed to partially execute the quote for group RFQ. You will receive error code 70507 if you don't pass in the full leg size.
2. Parent RFQ leg size will be the summation of the filled size of each child RFQ leg size while fee should also be the summation.
3. The blockTId of parent RFQ and the tradeld of parent RFQ legs will be empty. But there will be subaccount breakdown attached with blockTId and tradeld populated.

## GET QUOTE PRODUCTS

Retrieve the products which makers want to quote and receive RFQs for, and the corresponding price and size limit.

### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: READ

### HTTP REQUEST

```
GET /api/v5/rfq/maker-instrument-settings
```

### Request Example

### REQUEST PARAMETERS

None

### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "instType": "OPTION",
      "includeAll": true,
      "data": [
        {
          "instFamily": "BTC-USD",
          "maxBlockSz": "10000",
          "makerPxBand": "5"
        },
        {
          "instFamily": "SOL-USD",
          "maxBlockSz": "100000",
          "makerPxBand": "15"
        }
      ]
    },
    {
      "instType": "FUTURES",
      "includeAll": false,
      "data": [
        {
          "instFamily": "BTC-USD",
          "maxBlockSz": "10000",
          "makerPxBand": "5"
        },
        {
          "instFamily": "ETH-USDT",
          "maxBlockSz": "100000",
          "makerPxBand": "15"
        }
      ]
    }
  ]
}
```

```

        "maxBlockSz": "100000",
        "makerPxBand": "15"
    }
],
{
    "instType": "SWAP",
    "includeAll": false,
    "data": [
        {
            "instFamily": "BTC-USD",
            "maxBlockSz": "10000",
            "makerPxBand": "5"
        },
        {
            "instFamily": "ETH-USDT"
        }
    ]
},
{
    "instType": "SPOT",
    "includeAll": false,
    "data": [
        {
            "instId": "BTC-USDT"
        },
        {
            "instId": "TRX-USDT"
        }
    ]
}
]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, <code>0</code> means success.
msg	String	The error message, not empty if the code is not <code>0</code> .
data	Array of objects	Return data of the request.
> instType	String	Type of instrument. Valid value can be <code>FUTURES</code> , <code>OPTION</code> , <code>SWAP</code> or <code>SPOT</code> .
> includeAll	Boolean	Receive all instruments or not under specific instType setting. Valid value can be boolean ( <code>True</code> / <code>False</code> ). By default, the value will be <code>false</code> .
> data	Array of objects	Elements of the instType.
>> instFamily	String	Instrument family. Required for <code>FUTURES</code> , <code>OPTION</code> and <code>SWAP</code> only.
>> instId	String	Instrument ID. Required for <code>SPOT</code> only.
>> maxBlockSz	String	Max trade quantity for the product(s). For <code>FUTURES</code> , <code>OPTION</code> and <code>SWAP</code> , the max quantity of the RFQ/Quote is in unit of contracts. For <code>SPOT</code> , this parameter is in base currency.
>> makerPxBand	String	Price bands in unit of ticks, measured against mark price. Setting makerPxBand to 1 tick means: If Bid price > Mark + 1 tick, it will be stopped If Ask price < Mark - 1 tick, It will be stopped

## SET QUOTE PRODUCTS

Customize the products which makers want to quote and receive RFQs for, and the corresponding price and size limit.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

## PERMISSION: TRADE

## HTTP REQUEST

POST /api/v5/rfq/maker-instrument-settings

## Request Example

```

import okx.BlockTrading as BlockTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

blockTradingAPI = BlockTrading.BlockTradingAPI(apikey, secretkey, passphrase, False, flag)

# Set quote products
data =[{
    "instType": "OPTION",
    "data": [
        {
            "uly": "BTC-USD",
            "maxBlockSz": "10000",
            "makerPxBand": "5"
        },
        {
            "uly": "SOL-USD",
            "maxBlockSz": "100000",
            "makerPxBand": "15"
        }
    ]
}]

result = blockTradingAPI.set_marker_instrument(
    data
)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	Yes	Type of instrument. Valid value can be <code>FUTURES</code> , <code>OPTION</code> , <code>SWAP</code> or <code>SPOT</code> .
includeAll	Boolean	No	Receive all instruments or not under specific instType setting. Valid value can be boolean ( <code>True</code> / <code>False</code> ). By default, the value will be <code>false</code> .
data	Array of objects	Yes	Elements of the instType.
> instFamily	String	Conditional	Instrument family. Required for <code>FUTURES</code> , <code>OPTION</code> and <code>SWAP</code> only.
> instId	String	Conditional	Instrument ID. Required for <code>SPOT</code> only.
> maxBlockSz	String	No	Max trade quantity for the product(s). For <code>FUTURES</code> , <code>OPTION</code> and <code>SWAP</code> , the max quantity of the RFQ/Quote is in unit of contracts. For <code>SPOT</code> , this parameter is in base currency.
> makerPxBand	String	No	Price bands in unit of ticks, measured against mark price. Setting makerPxBand to 1 tick means: If Bid price > Mark + 1 tick, it will be stopped If Ask price < Mark - 1 tick, It will be stopped

## Response Example

```
{
  "code": "0",
  "msg": "OK"
}
```

```

"msg":"",
"data": [
  {
    "result":true
  }
]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, <code>0</code> means success.
msg	String	The error message, not empty if the code is not <code>0</code> .
data	Array of objects	Array of objects containing the results.
> result	Boolean	Result of the request Valid value is <code>true</code> or <code>false</code> .

## RESET MMP STATUS

Reset the MMP status to be inactive.

### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: TRADE

### HTTP REQUEST

```
POST /api/v5/rfq/mmp-reset
```

#### Request Example

```

import okx.BlockTrading as BlockTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

blockTradingAPI = BlockTrading.BlockTradingAPI(apikey, secretkey, passphrase, False, flag)

# Reset MMP status
result = blockTradingAPI.reset_mmp()
print(result)

```

## REQUEST PARAMETERS

None

#### Response Example

```

{
  "code":"0",
  "msg":"",
  "data": [
    {
      "ts":"1597026383085"
    }
  ]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, 0 means success.
msg	String	The error message, not empty if the code is not 0.
data	Array of objects	Array of objects containing the results.
> ts	String	The timestamp of re-setting successfully. Unix timestamp format in milliseconds, e.g. 1597026383085.

## SET MMP

This endpoint is used to set MMP configure and only applicable to block trading makers

**RATE LIMIT: 1 REQUEST PER 10 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

`POST /api/v5/rfq/mmp-config`

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
timeInterval	String	Yes	Time window (ms). MMP interval where monitoring is done. "0" means disable MMP. Maximum time interval is 600,000.
frozenInterval	String	Yes	Frozen period (ms). "0" means the trade will remain frozen until you request "Reset MMP Status" to unfrozen.
countLimit	String	Yes	Limit in number of execution attempts.

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "frozenInterval": "2000",
      "countLimit": "100",
      "timeInterval": "5000"
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
timeInterval	String	Time window (ms). MMP interval where monitoring is done
frozenInterval	String	Frozen period (ms).
countLimit	String	Limit in number of execution attempts

Group RFQ introduction

For RFQ makers, the execution attempt of group RFQ will only count once towards MMP regardless of how many account allocations involved.

## GET MMP CONFIG

This endpoint is used to get MMP configure information and only applicable to block trading market makers

### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: READ

### HTTP REQUEST

```
GET /api/v5/rfq/mmp-config
```

Request Example

### REQUEST PARAMETERS

none

Response Example

```
{
  "code": "0",
  "data": [
    {
      "frozenInterval": "2000",
      "mmpFrozen": true,
      "mmpFrozenUntil": "1000",
      "countLimit": "10",
      "timeInterval": "5000"
    }
  ],
  "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
timeInterval	String	Time window (ms). MMP interval where monitoring is done "0" means MMP is disabled
frozenInterval	String	Frozen period (ms). If it is "0", the trade will remain frozen until manually reset and <code>mmpFrozenUntil</code> will be "".
countLimit	String	Limit in number of execution attempts
mmpFrozen	Boolean	Whether MMP is currently triggered. <code>true</code> or <code>false</code>
mmpFrozenUntil	String	If frozenInterval is not "0" and mmpFrozen = True, it is the time interval (in ms) when MMP is no longer triggered, otherwise ""

## CREATE QUOTE

Allows the user to Quote an RFQ that they are a counterparty to. The user **MUST** quote the entire RFQ and not part of the legs or part of the quantity. Partial quoting is not allowed.

### RATE LIMIT: 50 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: TRADE

### HTTP REQUEST

```
POST /api/v5/rfq/create-quote
```

Request Example

```
import okx.BlockTrading as BlockTrading

# API initialization
apikey = "YOUR_API_KEY"
```

```

secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

blockTradingAPI = BlockTrading.BlockTradingAPI(apikey, secretkey, passphrase, False, flag)

# Create quote
result = blockTradingAPI.create_quote(
    rfqid="22539",
    clQuoteid="q001",
    anonymous=True,
    quoteSide="buy",
    expiresIn="30",
    legs=[
        {
            "px": "39450.0",
            "sz": "200000",
            "instId": "BTC-USDT-SWAP",
            "side": "buy"
        }
    ]
)
print(result)

```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
rfqid	String	Yes	RFQ ID .
clQuoteid	String	No	Client-supplied Quote ID. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
tag	String	No	Quote tag. The block trade associated with the Quote will have the same tag. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters.
anonymous	Boolean	No	Submit Quote on a disclosed or anonymous basis. Valid value is <code>true</code> or <code>false</code> . <code>false</code> by default.
quoteSide	String	Yes	The trading direction of the Quote. Its value can be <code>buy</code> or <code>sell</code> . For example, if quoteSide is <code>buy</code> , all the legs are executed in their leg sides; otherwise, all the legs are executed in the opposite of their leg sides.
expiresIn	String	No	Seconds that a quote expires in. Must be an integer between 10-120. Default is 60.
legs	Array of objects	Yes	The legs of the Quote.
> instId	String	Yes	The instrument ID of quoted leg.
> tdMode	String	No	Trade mode Margin mode: <code>cross</code> <code>isolated</code> Non-Margin mode: <code>cash</code> . If not provided, tdMode will inherit default values set by the system shown below: Futures mode mode & SPOT: <code>cash</code> Buy options in Futures mode and Multi-currency Margin: <code>isolated</code> Other cases: <code>cross</code>
> ccy	String	No	Margin currency. Only applicable to <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code> . The parameter will be ignored in other scenarios.
> sz	String	Yes	Size of the leg in contracts or spot.
> px	String	Yes	The price of the leg.

Parameter	Type	Required	Description
> side	String	Yes	The direction of the leg. Valid values can be buy or sell.
> posSide	String	No	Position side. The default is <code>net</code> in the net mode. It can only be <code>long</code> or <code>short</code> in the long/short mode. If not specified, users in long/short mode always open new positions. Only applicable to <code>FUTURES</code> / <code>SWAP</code> .
> tgtCcy	String	No	Defines the unit of the "sz" attribute. Only applicable to <code>instType = SPOT</code> . The valid enumerations are <code>base_ccy</code> and <code>quote_ccy</code> . When not specified this is equal to <code>base_ccy</code> by default.
> tradeQuoteCcy	String	No	The quote currency used for trading. Only applicable to SPOT. The default value is the quote currency of the instId, for example: for <code>BTC-USD</code> , the default is <code>USD</code> .

### Response Example

```
{
  "code": "",
  "msg": "",
  "data": [
    {
      "validUntil": "1608997227834",
      "uTime": "1608267227834",
      "cTime": "1608267227834",
      "legs": [
        {
          "px": "46000",
          "sz": "25",
          "instId": "BTC-USD-220114-25000-C",
          "tdMode": "cross",
          "ccy": "USDT",
          "side": "sell",
          "posSide": "long",
          "tgtCcy": ""
        },
        {
          "px": "4000",
          "sz": "25",
          "instId": "ETH-USD-220114-25000-C",
          "tdMode": "cross",
          "ccy": "USDT",
          "side": "buy",
          "posSide": "long",
          "tgtCcy": ""
        }
      ],
      "quoteId": "25092",
      "rfqId": "18753",
      "tag": "123456",
      "quoteSide": "sell",
      "state": "active",
      "reason": "mmp_canceled",
      "clQuoteId": "",
      "clRfqId": "",
      "traderCode": "Aksha"
    }
  ]
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, <code>0</code> means success.
msg	String	The error message, not empty if the code is not 0.

Parameter	Type	Description
data	Array of objects	Array of objects containing the results
> cTime	String	The timestamp the Quote was created, Unix timestamp format in milliseconds.
> uTime	String	The timestamp the Quote was last updated, Unix timestamp format in milliseconds.
> state	String	The status of the quote. Valid values can be <code>active</code> <code>cancelled</code> <code>pending_fill</code> <code>filled</code> <code>expired</code> or <code>failed</code> .
> reason	String	Reasons of state. Valid values can be <code>mmp_cancelled</code> .
> validUntil	String	The timestamp the Quote expires. Unix timestamp format in milliseconds.
> rfqid	String	RFQ ID
> clRfqId	String	Client-supplied RFQ ID. This attribute is treated as client sensitive information. It will not be exposed to the Maker, only return empty string.
> quoteld	String	Quote ID.
> clQuoteld	String	Client-supplied Quote ID. This attribute is treated as client sensitive information. It will not be exposed to the Taker, only return empty string.
> tag	String	Quote tag. The block trade associated with the Quote will have the same tag.
> traderCode	String	A unique identifier of maker.
> quoteSide	String	The trading direction of the Quote. Its value can be <code>buy</code> or <code>sell</code> . For example, if quoteSide is <code>buy</code> , all the legs are executed in their leg sides; otherwise, all the legs are executed in the opposite of their leg sides.
> legs	Array of objects	The legs of the Quote.
>> instId	String	Instrument ID, e.g. <code>BTC-USDT-SWAP</code>
>> tdMode	String	Trade mode Margin mode: <code>cross</code> <code>isolated</code> Non-Margin mode: <code>cash</code> . If not provided, tdMode will inherit default values set by the system shown below: Futures mode & SPOT: <code>cash</code> Buy options in Futures mode and Multi-currency Margin: <code>isolated</code> Other cases: <code>cross</code>
>> ccy	String	Margin currency. Only applicable to <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code> . The parameter will be ignored in other scenarios.
>> sz	String	Size of the leg in contracts or spot.
>> px	String	The price of the leg.
>> side	String	The direction of the leg. Valid values can be buy or sell.
>> posSide	String	Position side. The default is <code>net</code> in the net mode. If not specified, return "", which is equivalent to net. It can only be <code>long</code> or <code>short</code> in the long/short mode. If not specified, return "", which corresponds to the direction that opens new positions for the trade (buy => long, sell => short). Only applicable to FUTURES/SWAP.
>> tgtCcy	String	Defines the unit of the "sz" attribute. Only applicable to instType = SPOT.

Parameter	Type	Description
		The valid enumerations are <code>base_ccy</code> and <code>quote_ccy</code> . When not specified this is equal to <code>base_ccy</code> by default.
<code>&gt;&gt; tradeQuoteCcy</code>	String	The quote currency used for trading. Only applicable to SPOT. The default value is the quote currency of the instId, for example: for <code>BTC-USD</code> , the default is <code>USD</code> .

## CANCEL QUOTE

Cancels an existing active Quote you have created in response to an RFQ.

**RATE LIMIT: 50 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

`POST /api/v5/rfq/cancel-quote`

### Request Example

```
import okx.BlockTrading as BlockTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

blockTradingAPI = BlockTrading.BlockTradingAPI(apikey, secretkey, passphrase, False, flag)

# Cancel quote
result = blockTradingAPI.cancel_quote(
    quoteId="007",
    clQuoteId="Bond007"
)
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
<code>quotId</code>	String	Conditional	Quote ID.
<code>clQuotId</code>	String	Conditional	Client-supplied Quote ID. Either <code>quoteId</code> or <code>clQuoteId</code> is required. If both <code>clQuoteId</code> and <code>quoteId</code> are passed, <code>quoteId</code> will be treated as primary identifier.
<code>rfqId</code>	String	No	RFQ ID.

### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "quoteId": "007",
      "clQuoteId": "Bond007",
      "sCode": "0",
      "sMsg": ""
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, <code>0</code> means success.
msg	String	The error message, not empty if the code is not 0.
data	Array of objects	Array of objects containing the results
> quoteId	String	Quote ID
> clQuoteId	String	Client-supplied Quote ID.
> sCode	String	The code of the event execution result, <code>0</code> means success.
> sMsg	String	Rejection message if the request is unsuccessful.

#### CANCEL MULTIPLE QUOTES

Cancel one or multiple active Quote(s) in a single batch. Maximum 100 quote orders can be canceled per request.

**RATE LIMIT: 2 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

`POST /api/v5/rfq/cancel-batch-quotes`

Request Example

```
import okx.BlockTrading as BlockTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

blockTradingAPI = BlockTrading.BlockTradingAPI(apikey, secretkey, passphrase, False, flag)

# Cancel multiple quotes
result = blockTradingAPI.cancel_batch_quotes(
    quoteIds=["1150", "1151", "1152"],
    clQuoteIds=["q1", "q2", "q3"]
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
quotids	Array of strings	Conditional	Quote IDs .
clQuotids	Array of strings	Conditional	Client-supplied Quote IDs. Either <code>quoteIds</code> or <code>clQuoteIds</code> is required. If both attributes are sent, <code>quoteIds</code> will be used as primary identifier.

Success - All requested Quotes canceled

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "quoteId": "1150",
      "clQuoteId": "q1",
      "sCode": "0",
      "sMsg": ""
    }
  ]
}
```

```

    "sMsg":"",
},
{
    "quoteId":"1151",
    "clQuoteId":"q2",
    "sCode":"0",
    "sMsg":""
},
{
    "quoteId":"1152",
    "clQuoteId":"q3",
    "sCode":"0",
    "sMsg":""
}
]
}

```

## Partial cancellation

```

{
    "code":"2",
    "msg":"Bulk operation partially succeeded.",
    "data": [
        {
            "quoteId":"1150",
            "clQuoteId":"q1",
            "sCode":"0",
            "sMsg":""
        },
        {
            "quoteId":"1151",
            "clQuoteId":"q2",
            "sCode":"70001",
            "sMsg":"Quote does not exist."
        },
        {
            "quoteId":"1152",
            "clQuoteId":"q3",
            "sCode":"70001",
            "sMsg":"Quote does not exist."
        }
    ]
}

```

## Failure example

```

{
    "code":"1",
    "msg":"Operation failed.",
    "data": [
        {
            "quoteId":"1150",
            "clQuoteId":"q1",
            "sCode":"70001",
            "sMsg":"Quote does not exist."
        },
        {
            "quoteId":"1151",
            "clQuoteId":"q2",
            "sCode":"70001",
            "sMsg":"Quote does not exist."
        },
        {
            "quoteId":"1151",
            "clQuoteId":"q3",
            "sCode":"70001",
            "sMsg":"Quote does not exist."
        }
    ]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, <code>0</code> means success.
msg	String	The error message, not empty if the code is not 0.
data	Array of objects	Array of objects containing the results
> quotelid	String	Quote ID
> clQuotelid	String	Client-supplied Quote ID.
> sCode	String	The code of the event execution result, <code>0</code> means success.
> sMsg	String	Rejection message if the request is unsuccessful.

## CANCEL ALL QUOTES

Cancels all active Quotes.

### RATE LIMIT: 2 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: USER ID

#### PERMISSION: TRADE

#### HTTP REQUEST

POST /api/v5/rfq/cancel-all-quotes

#### Request Example

```
import okx.BlockTrading as BlockTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

blockTradingAPI = BlockTrading.BlockTradingAPI(apikey, secretkey, passphrase, False, flag)

# Cancel all quotes
result = blockTradingAPI.cancel_all_quotes()
print(result)
```

## REQUEST PARAMETERS

None

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ts": "1697026383085"
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, <code>0</code> means success.

Parameter	Type	Description
msg	String	The error message, not empty if the code is not 0.
data	Array of objects	Array of objects containing the results
> ts	String	The timestamp of cancellation successfully. Unix timestamp format in milliseconds, e.g. 1597026383085.

#### CANCEL ALL AFTER

Cancel all quotes after the countdown timeout.

#### RATE LIMIT: 1 REQUEST PER SECOND

#### RATE LIMIT RULE: USER ID

#### PERMISSION: TRADE

#### HTTP REQUEST

`POST /api/v5/rfq/cancel-all-after`

Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
timeOut	String	Yes	The countdown for quotes cancellation, with second as the unit. Range of value can be 0, [10, 120]. Setting timeOut to 0 disables Cancel All After.

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "triggerTime": "1587971460",
      "ts": "1587971400"
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
triggerTime	String	The time the cancellation is triggered. triggerTime=0 means Cancel All After is disabled.
ts	String	The time the request is received.

Users are recommended to send a request to the exchange every second. When the cancel all after is triggered, the trading engine will cancel quotes on behalf of the client one by one and this operation may take up to a few seconds. This feature is intended as a protection mechanism for clients only and clients should not use this feature as part of their trading strategies.

#### GET RFQS

Retrieves details of RFQs that the user is a counterparty to (either as the creator or the receiver of the RFQ).

#### RATE LIMIT: 2 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: USER ID

#### PERMISSION: READ

## HTTP REQUEST

GET /api/v5/rfq/rfqs

### Request Example

```
import okx.BlockTrading as BlockTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

blockTradingAPI = BlockTrading.BlockTradingAPI(apikey, secretkey, passphrase, False, flag)

# Retrieves details of RFQs that the user is a counterparty to
result = blockTradingAPI.get_rfqs()
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
rfqid	String	No	RFQ ID .
clRfqId	String	No	Client-supplied RFQ ID. If both <code>clRfqId</code> and <code>rfqid</code> are passed, <code>rfqid</code> will be treated as primary identifier
state	String	No	The status of the RFQ. Valid values can be <code>active</code> <code>canceled</code> <code>pending_fill</code> <code>filled</code> <code>expired</code> <code>failed</code> <code>traded_away</code> . <code>traded_away</code> only applies to Maker
beginId	String	No	Start rfq id the request to begin with. Pagination of data to return records newer than the requested rfqid, not including beginId
endId	String	No	End rfq id the request to end with. Pagination of data to return records earlier than the requested rfqid, not including endId
limit	String	No	Number of results per request. The maximum is 100 which is also the default value.

### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "rfqid": "123456",
      "clRfqId": "",
      "tag": "123456",
      "traderCode": "VITALIK",
      "validUntil": "1650969031817",
      "allowPartialExecution": false,
      "state": "filled",
      "flowType": "",
      "counterparties": [
        "SATOSHI"
      ],
      "legs": [
        {
          "instId": "BTC-USDT",
          "tdMode": "cross",
          "ccy": "USDT",
          "side": "buy",
          "posSide": "long",
          "sz": "25",
          "tgtCcy": "base_ccy",
          "tradeQuoteCcy": "USDT"
        }
      ],
      "status": "filled"
    }
  ]
}
```

```

    "cTime": "1650968131817",
    "uTime": "1650968164944"
},
{
    "rfqId": "1234567",
    "clRfqId": "",
    "tag": "1234567",
    "traderCode": "VITALIK",
    "validUntil": "1650967623729",
    "state": "filled",
    "flowType": "",
    "counterparties": [
        "SATOSHI"
    ],
    "legs": [
        {
            "instId": "BTC-USDT",
            "tdMode": "cross",
            "ccy": "USDT",
            "side": "buy",
            "posSide": "long",
            "sz": "1500000",
            "tgtCcy": "quote_ccy",
            "tradeQuoteCcy": "USDT"
        }
    ],
    "cTime": "1650966723729",
    "uTime": "1650966816577"
}
]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, <code>0</code> means success.
msg	String	The error message, not empty if the code is not 0.
data	Array of objects	Array of objects containing the results of the RFQ creation.
> cTime	String	The timestamp the RFQ was created. Unix timestamp format in milliseconds.
> uTime	String	The timestamp the RFQ was last updated. Unix timestamp format in milliseconds.
> state	String	The status of the RFQ. Valid values can be <code>active</code> <code>canceled</code> <code>pending_fill</code> <code>filled</code> <code>expired</code> <code>failed</code> <code>traded_away</code> . <code>traded_away</code> only applies to Maker
> counterparties	Array of strings	The list of counterparties traderCode the RFQ was broadcasted to.
> validUntil	String	The timestamp the RFQ expires. Unix timestamp format in milliseconds.
> clRfqId	String	Client-supplied RFQ ID. This attribute is treated as client sensitive information. It will not be exposed to the Maker, only return empty string.
> tag	String	RFQ tag. The block trade associated with the RFQ will have the same tag.
> flowType	String	Identify the type of the RFQ. Only applicable to Makers, return "" for Takers
> traderCode	String	A unique identifier of taker. Empty if the anonymous parameter of the RFQ is set to be <code>true</code> .
> rfqId	String	RFQ ID.

Parameter	Type	Description
> allowPartialExecution	Boolean	Whether the RFQ can be partially filled provided that the shape of legs stays the same. Valid value is <code>true</code> or <code>false</code> , <code>false</code> by default.
> legs	Array of objects	Legs of RFQ
>> instId	String	Instrument ID, e.g. BTC-USDT-SWAP
>> tdMode	String	<p>Trade mode</p> <p>Margin mode: <code>cross</code> <code>isolated</code></p> <p>Non-Margin mode: <code>cash</code>.</p> <p>If not provided, tdMode will inherit default values set by the system shown below:</p> <p>Futures mode &amp; SPOT: <code>cash</code></p> <p>Buy options in Futures mode and Multi-currency Margin: <code>isolated</code></p> <p>Other cases: <code>cross</code></p>
>> ccy	String	<p>Margin currency.</p> <p>Only applicable to <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code>. The parameter will be ignored in other scenarios.</p>
>> sz	String	Size of the leg in contracts or spot.
>> side	String	The direction of the leg. Valid values can be buy or sell.
>> posSide	String	<p>Position side.</p> <p>The default is <code>net</code> in the net mode. If not specified, return "", which is equivalent to net.</p> <p>It can only be <code>long</code> or <code>short</code> in the long/short mode. If not specified, return "", which corresponds to the direction that opens new positions for the trade (buy =&gt; long, sell =&gt; short).</p> <p>Only applicable to <code>FUTURES</code> / <code>SWAP</code>.</p>
>> tgtCcy	String	<p>Defines the unit of the "sz" attribute.</p> <p>Only applicable to instType = SPOT.</p> <p>The valid enumerations are <code>base_ccy</code> and <code>quote_ccy</code>. When not specified this is equal to <code>base_ccy</code> by default.</p>
>> tradeQuoteCcy	String	<p>The quote currency used for trading. Only applicable to SPOT.</p> <p>The default value is the quote currency of the instId, for example: for <code>BTC-USD</code>, the default is <code>USD</code>.</p>
> groupId	String	<p>Group RFQ ID</p> <p>Only applicable to group RFQ, return "" for normal RFQ</p>
> acctAlloc	Array of objects	<p>Account level allocation of the RFQ</p> <p>This is only applicable to the taker.</p>
>> acct	String	The name of the allocated account of the RFQ.
>> legs	Array of objects	The allocated legs of the account.
>>> instId	String	The Instrument ID of each leg. Example : "BTC-USDT-SWAP"
>>> sz	String	The allocated size of each leg.
>>> tdMode	String	Trade mode
>>> ccy	String	Margin currency
>>> posSide	String	Position side

## Group RFQ introduction

1. allowPartialExecution field is always true for group RFQ for taker and maker.

2. Add a new response parameter `acctAlloc` with all account allocation the same as the initial request, but it is only applicable to takers.
3. Add a new response parameter `groupId`, applicable to both takers and makers.
4. For group RFQ state,
  1. if any allocated account is pending execution, then `pending_fill`
  2. otherwise,
    1. if any allocated account is filled, then `filled`
    2. if all allocated accounts are failed, then `failed`

## GET QUOTES

Retrieve all Quotes that the user is a counterparty to (either as the creator or the receiver).

**RATE LIMIT: 2 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/rfq/quotes`

Request Example

```
import okx.BlockTrading as BlockTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

blockTradingAPI = BlockTrading.BlockTradingAPI(apikey, secretkey, passphrase, False, flag)

# Retrieve all Quotes that the user is a counterparty to
result = blockTradingAPI.get_quotes()
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
<code>rfqid</code>	String	No	RFQ ID .
<code>clRfqId</code>	String	No	Client-supplied RFQ ID. If both <code>clRfqId</code> and <code>rfqid</code> are passed, <code>rfqid</code> will be treated as primary identifier.
<code>quoteld</code>	String	No	Quote ID
<code>clQuoteld</code>	String	No	Client-supplied Quote ID. If both <code>clQuoteld</code> and <code>quoteld</code> are passed, <code>quoteld</code> will be treated as primary identifier
<code>state</code>	String	No	The status of the quote. Valid values can be <code>active</code> <code>canceled</code> <code>pending_fill</code> <code>filled</code> <code>expired</code> or <code>failed</code> .
<code>beginId</code>	String	No	Start quote id the request to begin with. Pagination of data to return records newer than the requested quoteld, not including beginId
<code>endId</code>	String	No	End quote id the request to end with. Pagination of data to return records earlier than the requested quoteld, not including endId
<code>limit</code>	String	No	Number of results per request. The maximum is 100 which is also the default value.

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
```

```

    "validUntil": "1608997227834",
    "uTime": "1608267227834",
    "cTime": "1608267227834",
    "legs": [
        {
            "px": "46000",
            "sz": "25",
            "instId": "BTC-USD-220114-25000-C",
            "tdMode": "cross",
            "ccy": "USDT",
            "side": "sell",
            "posSide": "long",
            "tgtCcy": ""
        },
        {
            "px": "45000",
            "sz": "25",
            "instId": "BTC-USDT",
            "tdMode": "cross",
            "ccy": "USDT",
            "side": "buy",
            "posSide": "long",
            "tgtCcy": "base_ccy",
            "tradeQuoteCcy": "USDT"
        }
    ],
    "quoteId": "25092",
    "rfqId": "18753",
    "quoteSide": "sell",
    "state": "canceled",
    "reason": "mmp_canceled",
    "clQuoteId": "cq001",
    "clRfqId": "cr001",
    "tag": "123456",
    "traderCode": "Trader1"
}
]
}
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, <code>0</code> means success.
msg	String	The error message, not empty if the code is not 0.
data	Array of objects	Array of objects containing the results of the Quote creation.
> cTime	String	The timestamp the Quote was created, Unix timestamp format in milliseconds.
> uTime	String	The timestamp the Quote was last updated, Unix timestamp format in milliseconds.
> state	String	The status of the quote. Valid values can be <code>active</code> <code>canceled</code> <code>pending_fill</code> <code>filled</code> <code>expired</code> or <code>failed</code> .
> reason	String	Reasons of state. Valid values can be <code>mmp_canceled</code> .
> validUntil	String	The timestamp the Quote expires. Unix timestamp format in milliseconds.
> rfqId	String	RFQ ID.
> clRfqId	String	Client-supplied RFQ ID. This attribute is treated as client sensitive information. It will not be exposed to the Maker, only return empty string.
> quoteld	String	Quote ID.
> clQuoteld	String	Client-supplied Quote ID. This attribute is treated as client sensitive information. It will not be exposed to the Taker, only return empty string.

Parameter	Type	Description
> tag	String	Quote tag. The block trade associated with the Quote will have the same tag.
> traderCode	String	A unique identifier of maker. Empty If the anonymous parameter of the Quote is set to be <code>true</code> .
> quoteSide	String	Top level direction of Quote. Its value can be buy or sell.
> legs	Array of objects	The legs of the Quote.
>> instId	String	The instrument ID of the quoted leg.
>> tdMode	String	<p>Trade mode Margin mode: <code>cross</code> <code>isolated</code> Non-Margin mode: <code>cash</code>. If not provided, tdMode will inherit default values set by the system shown below: Futures mode &amp; SPOT: <code>cash</code> Buy options in Futures mode and Multi-currency Margin: <code>isolated</code> Other cases: <code>cross</code></p>
>> ccy	String	Margin currency. Only applicable to <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code> . The parameter will be ignored in other scenarios.
>> sz	String	Size of the leg in contracts or spot.
>> px	String	The price of the leg.
>> side	String	The direction of the leg. Valid values can be buy or sell.
>> posSide	String	<p>Position side. The default is <code>net</code> in the net mode. If not specified, return "", which is equivalent to net. It can only be <code>long</code> or <code>short</code> in the long/short mode. If not specified, return "", which corresponds to the direction that opens new positions for the trade (buy =&gt; long, sell =&gt; short). Only applicable to <code>FUTURES</code>/<code>SWAP</code>.</p>
>> tgtCcy	String	<p>Defines the unit of the "sz" attribute. Only applicable to <code>instType = SPOT</code>. The valid enumerations are <code>base_ccy</code> and <code>quote_ccy</code>. When not specified this is equal to <code>base_ccy</code> by default.</p>
>> tradeQuoteCcy	String	<p>The quote currency used for trading. Only applicable to SPOT. The default value is the quote currency of the <code>instId</code>, for example: for <code>BTC-USD</code>, the default is <code>USD</code>.</p>

## GET TRADES

Retrieves the executed trades that the user is a counterparty to (either as the creator or the receiver).

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/rfq/trades`

Request Example

```
import okx.BlockTrading as BlockTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

blockTradingAPI = BlockTrading.BlockTradingAPI(apikey, secretkey, passphrase, False, flag)
```

```
# Retrieves the executed trades that the user is a counterparty to
result = blockTradingAPI.get_trades()
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
rfqid	String	No	RFQ ID .
clRfqId	String	No	Client-supplied RFQ ID. If both <code>clRfqId</code> and <code>rfqid</code> are passed, <code>rfqid</code> will be treated as primary identifier
quoteid	String	No	Quote ID
blockTdId	String	No	Block trade ID
clQuoteid	String	No	Client-supplied Quote ID. If both <code>clQuoteid</code> and <code>quoteid</code> are passed, <code>quoteid</code> will be treated as primary identifier
beginId	String	No	The starting rfq id the request to begin with. Pagination of data to return records newer than the requested blockTdId, not including beginId.
endId	String	No	The last rfq id the request to end with Pagination of data to return records earlier than the requested blockTdId, not including endId.
beginTs	String	No	Filter trade execution time with a begin timestamp (UTC timezone). Unix timestamp format in milliseconds, e.g. 1597026383085
endTs	String	No	Filter trade execution time with an end timestamp (UTC timezone). Unix timestamp format in milliseconds, e.g. 1597026383085
limit	String	No	Number of results per request. The maximum is 100 which is also the default value. If the number of trades in the requested range is bigger than 100, the latest 100 trades in the range will be returned.
isSuccessful	Boolean	No	Whether the trade is filled successfully. <code>true</code> : the default value. <code>false</code> .

## Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "rfqId": "123456",
      "clRfqId": "",
      "quoteId": "0T53420",
      "clQuoteId": "",
      "blockTdId": "439127542058958848",
      "tag": "123456",
      "isSuccessful": true,
      "errorCode": "",
      "legs": [
        {
          "instId": "BTC-USDT",
          "side": "sell",
          "sz": "0.666",
          "px": "100",
          "tradeId": "439127542058958850",
          "fee": "-0.0333",
          "feeCcy": "USDT",
          "tradeQuoteCcy": "USDT"
        }
      ],
      "cTime": "1650968164900",
      "tTraderCode": "SATS",
      "mTraderCode": "MIKE"
    }
  ]
}
```

```

},
{
  "rfqId": "1234567",
  "clRfqId": "",
  "quoteId": "0T533T0",
  "clQuoteId": "",
  "blockTdId": "439121886014849024",
  "tag": "123456",
  "isSuccessful": true,
  "errorCode": "",
  "legs": [
    {
      "instId": "BTC-USDT",
      "side": "sell",
      "sz": "0.532",
      "px": "100",
      "tradeId": "439121886014849026",
      "fee": "-0.0266",
      "feeCcy": "USDT",
      "tradeQuoteCcy": "USDT"
    }
  ],
  "cTime": "1650966816550",
  "tTraderCode": "SATS",
  "mTraderCode": "MIKE"
}
]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, <code>0</code> means success.
msg	String	The error message, not empty if the code is not 0.
data	Array of objects	Array of objects containing the results of the block trade.
> cTime	String	The time the trade was executed. Unix timestamp in milliseconds.
> rfqId	String	RFQ ID.
> clRfqId	String	Client-supplied RFQ ID. This attribute is treated as client sensitive information. It will not be exposed to the Maker, only return empty string.
> quotId	String	Quote ID.
> clQuotId	String	Client-supplied Quote ID. This attribute is treated as client sensitive information. It will not be exposed to the Taker, only return empty string.
> blockTdId	String	Block trade ID.
> tag	String	Trade tag. The block trade will have the tag of the RFQ or Quote it corresponds to.
> tTraderCode	String	A unique identifier of the Taker. Empty if the anonymous parameter of the RFQ is set to be <code>true</code> .
> mTraderCode	String	A unique identifier of the Maker. Empty if the anonymous parameter of the Quote is set to be <code>true</code> .
> isSuccessful	Boolean	Whether the trade is filled successfully
> errorCode	String	Error code for unsuccessful trades. It is "" for successful trade.
> legs	Array of objects	Legs of trade
>> instId	String	Instrument ID, e.g. <code>BTC-USDT-SWAP</code>

Parameter	Type	Description
>> px	String	The price the leg executed
>> sz	String	Size of the leg in contracts or spot.
>> side	String	The direction of the leg. Valid value can be buy or sell.
>> fee	String	Fee. Negative number represents the user transaction fee charged by the platform. Positive number represents rebate.
>> feeCcy	String	Fee currency
>> tradeld	String	Last traded ID.
>> tradeQuoteCcy	String	The quote currency used for trading. Only applicable to SPOT. The default value is the quote currency of the instId, for example: for <code>(BTC-USD)</code> , the default is <code>(USD)</code> .
> acctAlloc	Array of objects	Applicable to both taker, maker
>> blockTdId	String	Block trade ID
>> errorCode	String	Error code for unsuccessful trades. It is "0" for successful trade.
>> acct	String	The name of the allocated account of the RFQ Only applicable to taker, return "" to makers
>> legs	Array of objects	The allocated legs of the account.
>>> instId	String	The Instrument ID of each leg. Example : "BTC-USDT-SWAP"
>>> sz	String	Filled size
>>> tradeld	String	Trade ID
>>> fee	String	Fee
>>> feeCcy	String	Fee currency

#### Group RFQ introduction

1. This endpoint is at parent RFQ level and contains account allocation. For parent RFQ, we should return the actual executed size, i.e. failed execution size should not be included in the parent RFQ level.
2. For account allocation, we should include both filled and failed child RFQ but add an errorCode to indicate whether a child RFQ is filled.
3. Trade results will only be returned to group RFQ creator. Allocated subaccounts and MSAs will not see trade results. Allocated accounts are expected to get these trades through trading bills.
4. Trades data will only be returned after all child RFQs are executed.
5. For parent RFQ isSuccessful field,
  1. it will return true if any child RFQs are filled
  2. otherwise, if all child RFQ fails, it will return false
6. Parent RFQ blockTdId or legs tradeld will be empty. However, account allocation breakdown will be offered and blockTdId/tradeld will be attached.

#### GET BLOCK TICKERS

Retrieve the latest block trading volume in the last 24 hours.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

## HTTP REQUEST

GET /api/v5/market/block-tickers

### Request Example

```
import okx.MarketData as MarketData

flag = "0" # Production trading:0 , demo trading:1

marketDataAPI = MarketData.MarketAPI(flag=flag)

# Retrieve the latest block trading volume in the last 24 hours
result = marketDataAPI.get_block_tickers(
    instType="SPOT"
)
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	Yes	Instrument type <input type="button" value="SPOT"/> <input type="button" value="SWAP"/> <input type="button" value="FUTURES"/> <input type="button" value="OPTION"/>
instFamily	String	No	Instrument family, e.g. <input type="button" value="BTC-USD"/> Applicable to <input type="button" value="FUTURES"/> <input type="button" value="SWAP"/> <input type="button" value="OPTION"/>

### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "instType": "SWAP",
      "instId": "LTC-USD-SWAP",
      "volCcy24h": "2222",
      "vol24h": "2222",
      "ts": "1597026383085"
    },
    {
      "instType": "SWAP",
      "instId": "BTC-USD-SWAP",
      "volCcy24h": "2222",
      "vol24h": "2222",
      "ts": "1597026383085"
    }
  ]
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID
instType	String	Instrument type
volCcy24h	String	24h trading volume, with a unit of <input type="button" value="currency"/> If it is a <input type="button" value="derivatives"/> contract, the value is the number of base currency. If it is <input type="button" value="SPOT"/> <input type="button" value="MARGIN"/> , the value is the quantity in quote currency.

Parameter	Type	Description
vol24h	String	24h trading volume, with a unit of <code>contract</code> . If it is a <code>derivatives</code> contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in base currency.
ts	String	Block ticker data generation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

#### GET BLOCK TICKER

Retrieve the latest block trading volume in the last 24 hours.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

**HTTP REQUEST**

`GET /api/v5/market/block-ticker`

Request Example

```
import okx.MarketData as MarketData

flag = "0" # Production trading:0 , demo trading:1

marketDataAPI = MarketData.MarketAPI(flag=flag)

# Retrieve the latest block trading volume in the last 24 hours
result = marketDataAPI.get_block_ticker(
    instId="BTC-USDT"
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USD-SWAP</code>

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "instType": "SWAP",
      "instId": "LTC-USD-SWAP",
      "volCcy24h": "2222",
      "vol24h": "2222",
      "ts": "1597026383085"
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID
instType	String	Instrument type
volCcy24h	String	24h trading volume, with a unit of <code>currency</code> . If it is a <code>derivatives</code> contract, the value is the number of base currency. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in quote currency.

Parameter	Type	Description
vol24h	String	24h trading volume, with a unit of <code>contract</code> . If it is a <code>derivatives</code> contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in base currency.
ts	String	Block ticker data generation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

#### GET PUBLIC MULTI-LEG TRANSACTIONS OF BLOCK TRADES

Retrieves the executed block trades. The data will be updated 15 minutes after the block trade execution.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

**HTTP REQUEST**

`GET /api/v5/rfq/public-trades`

Request Example

```
import okx.BlockTrading as BlockTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

blockTradingAPI = BlockTrading.BlockTradingAPI(apikey, secretkey, passphrase, False, flag)

# Retrieves the executed block trades
result = blockTradingAPI.get_public_trades()
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
beginId	String	No	The starting <code>blockTdId</code> the request to begin with. Pagination of data to return records newer than the requested <code>blockTdId</code> , not including beginId.
endId	String	No	The last <code>blockTdId</code> the request to end with. Pagination of data to return records earlier than the requested <code>blockTdId</code> , not including endId.
limit	String	No	Number of results per request. The maximum is 100 which is also the default value.

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "blockTdId": "439161457415012352",
      "groupId": "",
      "legs": [
        {
          "instId": "BTC-USD-210826",
          "side": "sell",
          "sz": "100",
          "px": "11000",
          "tradeId": "439161457415012354"
        },
        {
          "instId": "BTC-USD-SWAP",
          "side": "sell",
          "sz": "100",
          "px": "11000",
          "tradeId": "439161457415012355"
        }
      ]
    }
  ]
}
```

```

    "px": "50",
    "tradeId": "439161457415012355"
},
{
    "instId": "BTC-USDT",
    "side": "buy",
    "sz": "0.1", //for public feed, spot "sz" is in baseccy
    "px": "10.1",
    "tradeId": "439161457415012356"
},
{
    "instId": "BTC-USD-210326-60000-C",
    "side": "buy",
    "sz": "200",
    "px": "0.008",
    "tradeId": "439161457415012357"
},
{
    "instId": "BTC-USD-220930-5000-P",
    "side": "sell",
    "sz": "200",
    "px": "0.008",
    "tradeId": "439161457415012360"
},
{
    "instId": "BTC-USD-220930-10000-C",
    "side": "sell",
    "sz": "200",
    "px": "0.008",
    "tradeId": "439161457415012361"
},
{
    "instId": "BTC-USD-220930-10000-P",
    "side": "sell",
    "sz": "200",
    "px": "0.008",
    "tradeId": "439161457415012362"
},
{
    "instId": "ETH-USD-220624-100100-C",
    "side": "sell",
    "sz": "100",
    "px": "0.008",
    "tradeId": "439161457415012363"
}
],
"strategy": "CALL_CALENDAR_SPREAD",
"cTime": "1650976251241"
}
]
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
code	String	The result code, 0 means success.
msg	String	The error message, not empty if the code is not 0.
data	Array of objects	Array of objects containing the results of the public block trade.
> strategy	String	Option strategy, e.g. CALL_CALENDAR_SPREAD
> cTime	String	The time the trade was executed. Unix timestamp in milliseconds.
> blockTId	String	Block trade ID.
> groupId	String	Group RFQ ID Only applicable to group RFQ, return "" for normal RFQ
> legs	Array of objects	Legs of trade

Parameter	Type	Description
>> instId	String	Instrument ID, e.g. BTC-USDT-SWAP
>> px	String	The price the leg executed
>> sz	String	Trade quantity For spot trading, the unit is base currency For <b>FUTURES</b> / <b>SWAP</b> / <b>OPTION</b> , the unit is contract.
>> side	String	The direction of the leg from the Takers perspective. Valid value can be buy or sell.
>> tradeld	String	Last traded ID.

### Group RFQ introduction

1. Add new response parameter groupId, facilitating clients to map subaccount execution to group RFQ. Only applicable to group RFQ, return "" for normal RFQ.
2. Data return by this endpoint should be at **\*\*parent RFQ level\*\*** regardless of the subaccounts allocation. blockTdId and tradeld will be empty.

## GET PUBLIC SINGLE-LEG TRANSACTIONS OF BLOCK TRADES

Retrieve the recent block trading transactions of an instrument. Descending order by tradeld. The data will be updated 15 minutes after the block trade execution.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

### HTTP REQUEST

GET /api/v5/public/block-trades

### Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <b>BTC-USDT</b>

### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "fillVol": "5",
      "fwdPx": "26857.86591585",
      "groupId": "",
      "idxPx": "26889.7",
      "instId": "BTC-USD-231013-22000-P",
      "markPx": "0.0000000000000001",
      "px": "0.0026",
      "side": "buy",
      "sz": "1",
      "tradeId": "632960608383700997",
      "ts": "1697181568974"
    }
  ]
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID
tradeld	String	Trade ID
px	String	Trade price
sz	String	Trade quantity For spot trading, the unit is base currency For <b>FUTURES</b> / <b>SWAP</b> / <b>OPTION</b> , the unit is contract.
side	String	Trade side <b>buy</b> <b>sell</b>
fillVol	String	Implied volatility Only applicable to <b>OPTION</b>
fwdPx	String	Forward price Only applicable to <b>OPTION</b>
idxPx	String	Index price Applicable to <b>FUTURES</b> , <b>SWAP</b> , <b>OPTION</b>
markPx	String	Mark price Applicable to <b>FUTURES</b> , <b>SWAP</b> , <b>OPTION</b>
groupId	String	Group RFQ ID Only applicable to group RFQ, return "" for normal RFQ
ts	String	Trade time, Unix timestamp format in milliseconds, e.g. <b>1597026383085</b> .

Up to 500 most recent historical public transaction data can be retrieved.

#### Group RFQ introduction

1. Add new response parameter groupId, facilitating clients to map subaccount execution to group RFQ. Only applicable to group RFQ, return "" for normal RFQ.
2. Data return by this endpoint should be at \*\*child RFQ execution level\*\* but split into a single leg. tradeld will be populated.

## WebSocket Private Channel

### RFQS CHANNEL

Retrieve the RFQs sent or received by the user. Data will be pushed whenever the user sends or receives an RFQ.

### URL PATH

/ws/v5/business (required login)

#### Request Example

```
import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
```

```

print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/business",
        useServerTime=False
    )
    await ws.start()
    args = [
        {
            "channel": "rfqs"
        }
    ]

    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>rfqs</code>

## Successful Response Example

```
{
    "id": "1512",
    "event": "subscribe",
    "arg": {
        "channel": "rfqs"
    },
    "connId": "a4d3ae55"
}
```

## Failure Response Example

```
{
    "id": "1512",
    "event": "error",
    "code": "60012",
    "msg": "Invalid request: {\\"op\\": \\"subscribe\\", \\"args\\": [{ \\"channel\\": \\"rfqs\\"}]}",
    "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Operation   
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name 
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

### Push Data Example

```
{
  "arg": {
    "channel": "rfqs",
    "uid": "77982378738415879"
  },
  "data": [
    {
      "cTime": "1611033737572",
      "uTime": "1611033737572",
      "traderCode": "DSK2",
      "rfqId": "22534",
      "clRfqId": "",
      "tag": "123456",
      "state": "active",
      "flowType": "",
      "validUntil": "1611033857557",
      "allowPartialExecution": false,
      "counterparties": [
        "DSK4",
        "DSK5"
      ],
      "legs": [
        {
          "instId": "BTCUSD-211208-36000-C",
          "tdMode": "cross",
          "ccy": "USDT",
          "sz": "25.0",
          "side": "buy",
          "posSide": "long",
          "tgtCcy": ""
        },
        {
          "instId": "ETHUSD-211208-45000-C",
          "tdMode": "cross",
          "ccy": "USDT",
          "sz": "25.0",
          "side": "sell",
          "posSide": "long",
          "tgtCcy": ""
        }
      ]
    }
  ]
}
```

### PUSH DATA PARAMETERS

Parameters	Types	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> uid	String	User Identifier
data	Array of objects	Subscribed data
> cTime	String	The timestamp the RFQ was created, Unix timestamp format in milliseconds.
> uTime	String	The timestamp the RFQ was updated latest, Unix timestamp format in milliseconds.
> state	String	The status of the RFQ. Valid values can be <code>active</code> , <code>canceled</code> , <code>filled</code> , <code>expired</code> , <code>traded_away</code> or <code>failed</code> . <code>traded_away</code> only applies to Maker.
> counterparties	Array of Strings	The list of counterparties traderCode the RFQ was broadcasted to.
> validUntil	String	The timestamp the RFQ expires. Unix timestamp format in milliseconds.
> clRfqId	String	Client-supplied RFQ ID. This attribute is treated as client sensitive information. It will not be exposed to the Maker. Return empty for Maker, eg. "".
> tag	String	RFQ tag. The block trade associated with the RFQ will have the same tag.
> flowType	String	Identify the type of the RFQ. Only applicable to Makers, return "" for Takers
> traderCode	String	A unique identifier of taker. Empty If anonymous mode is <code>True</code> .
> rfqId	String	RFQ ID
> allowPartialExecution	Boolean	Whether the RFQ can be partially filled provided that the shape of legs stays the same. Valid value is <code>true</code> or <code>false</code> . <code>false</code> by default.
> legs	Array of objects	An Array of objects containing each leg of the RFQ.
>> instId	String	Instrument ID, e.g. BTC-USDT-SWAP
>> tdMode	String	Trade mode Margin mode: <code>cross</code> <code>isolated</code> Non-Margin mode: <code>cash</code> . If not provided, tdMode will inherit default values set by the system shown below: Futures mode & SPOT: <code>cash</code> Buy options in Futures mode and Multi-currency Margin: <code>isolated</code> Other cases: <code>cross</code>
>> ccy	String	Margin currency. Only applicable to <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code> . The parameter will be ignored in other scenarios.
>> sz	String	Size of the leg.
>> side	String	The direction of the leg. Valid values can be buy or sell.
>> posSide	String	Position side. The default is <code>net</code> in the net mode. If not specified, return "", which is equivalent to net. It can only be <code>long</code> or <code>short</code> in the long/short mode. If not specified, return "", which corresponds to the direction that opens new positions for the trade (buy => long, sell => short). Only applicable to <code>FUTURES</code> / <code>SWAP</code> .

Parameters	Types	Description
>> tgtCcy	String	Defines the unit of the "sz" attribute. Only applicable to instType = SPOT. The valid enumerations are <code>base_ccy</code> and <code>quote_ccy</code> . When not specified this is equal to <code>base_ccy</code> by default.
>> tradeQuoteCcy	String	The quote currency used for trading. Only applicable to SPOT. The default value is the quote currency of the instId, for example: for <code>BTC-USD</code> , the default is <code>USD</code> .
> groupId	String	Group RFQ ID Only applicable to group RFQ, return "" for normal RFQ
> acctAlloc	Array of objects	Account level allocation of the RFQ This is only applicable to the taker.
>> acct	String	The name of the allocated account of the RFQ.
>> legs	Array of objects	The allocated legs of the account.
>>> instId	String	The Instrument ID of each leg. Example : "BTC-USDT-SWAP"
>>> sz	String	The allocated size of each leg.
>>> tdMode	String	Trade mode
>>> ccy	String	Margin currency
>>> posSide	String	Position side

state: pending\_fill is a kind of moment state, and this channel doesn't update it.

### Group RFQ introduction

1. allowPartialExecution field is always true for group RFQ for taker and maker.
2. Add a new response parameter acctAlloc with all account allocation the same as the initial request, but it is only applicable to takers.
3. Add a new response parameter groupId, applicable to both takers and makers.
4. For group RFQ state,
  1. if any allocated account is pending execution, then pending\_fill
  2. otherwise,
    1. if any allocated account is filled, then filled
    2. if all allocated accounts are failed, then failed

### QUOTES CHANNEL

Retrieve the Quotes sent or received by the user. Data will be pushed whenever the user sends or receives a Quote.

### URL PATH

/ws/v5/business (required login)

### Request Example

```
import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():
    pass
```

```

ws = WsPrivateAsync(
    apiKey = "YOUR_API_KEY",
    passphrase = "YOUR_PASSPHRASE",
    secretKey = "YOUR_SECRET_KEY",
    url = "wss://ws.okx.com:8443/ws/v5/business",
    useServerTime=False
)
await ws.start()
args = [
    {
        "channel": "quotes"
    }
]
await ws.subscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

await ws.unsubscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation subscribe unsubscribe
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name quotes

## Successful Response Example

```
{
    "id": "1512",
    "event": "subscribe",
    "arg": {
        "channel": "quotes"
    },
    "connId": "a4d3ae55"
}
```

## Failure Response Example

```
{
    "id": "1512",
    "event": "error",
    "code": "60012",
    "msg": "Invalid request: {\\"op\\": \\"subscribe\\\", \\"args\\\": [{ \\"channel\\\" : \\"quotes\\\" }]}",
    "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message

Parameter	Type	Required	Description
event	String	Yes	Operation   
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name 
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

### Push Data Example

```
{
  "arg": {
    "channel": "quotes",
    "uid": "77982378738415879"
  },
  "data": [
    {
      "validUntil": "1608997227854",
      "uTime": "1608267227834",
      "cTime": "1608267227834",
      "legs": [
        {
          "px": "0.0023",
          "sz": "25.0",
          "instId": "BTC-USD-220114-25000-C",
          "tdMode": "cross",
          "ccy": "USDT",
          "side": "sell",
          "posSide": "long",
          "tgtCcy": ""
        },
        {
          "px": "0.0045",
          "sz": "25",
          "instId": "BTC-USD-220114-35000-C",
          "tdMode": "cross",
          "ccy": "USDT",
          "side": "buy",
          "posSide": "long",
          "tgtCcy": ""
        }
      ],
      "quoteId": "25092",
      "rfqId": "18753",
      "tag": "123456",
      "traderCode": "SATS",
      "quoteSide": "sell",
      "state": "canceled",
      "reason": "mmp_canceled",
      "clQuoteId": ""
    }
  ]
}
```

### PUSH DATA PARAMETERS

Parameters	Types	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> uid	String	User Identifier
data	Array of objects	Subscribed data
> cTime	String	The timestamp the Quote was created, Unix timestamp format in milliseconds.
> uTime	String	The timestamp the Quote was updated latest, Unix timestamp format in milliseconds.
> state	String	The status of the quote. Valid values can be <code>active</code> <code>canceled</code> <code>filled</code> <code>expired</code> or <code>failed</code> .
> reason	String	Reasons of state. Valid values can be <code>mmp_canceled</code> .
> validUntil	String	The timestamp the Quote expires. Unix timestamp format in milliseconds.
> rfqid	String	RFQ ID.
> clRfqId	String	Client-supplied RFQ ID. This attribute is treated as client sensitive information. It will not be exposed to the Maker, just return empty string "" for Maker.
> quoteld	String	Quote ID
> clQuoteld	String	Client-supplied Quote ID. This attribute is treated as client sensitive information. It will not be exposed to the Taker, just return empty string "" for Taker.
> tag	String	Quote tag. The block trade associated with the Quote will have the same tag.
> traderCode	String	A unique identifier of maker. Empty If anonymous mode of Quote is <code>True</code> .
> quoteSide	String	Top level side of Quote. Its value can be buy or sell.
> legs	Array of objects	The legs of the Quote.
>> instId	String	The instrument name of quoted leg.
>> tdMode	String	<p>Trade mode            Margin mode: <code>cross</code> <code>isolated</code>            Non-Margin mode: <code>cash</code>.</p> <p>If not provided, tdMode will inherit default values set by the system shown below:            Futures mode &amp; SPOT: <code>cash</code>            Buy options in Futures mode and Multi-currency Margin: <code>isolated</code>            Other cases: <code>cross</code></p>
>> ccy	String	<p>Margin currency.            Only applicable to <code>cross</code> <code>MARGIN</code> orders in <code>Futures mode</code>. The parameter will be ignored in other scenarios.</p>
>> sz	String	The size of the quoted leg in contracts or spot.
>> px	String	The price of the leg.
>> side	String	The direction of the leg. Valid values can be buy or sell.
>> posSide	String	<p>Position side.            The default is <code>net</code> in the net mode. If not specified, return "", which is equivalent to net.            It can only be <code>long</code> or <code>short</code> in the long/short mode. If not specified, return "", which corresponds to the direction that opens new positions for the trade (buy =&gt; long, sell =&gt; short).            Only applicable to <code>FUTURES</code>/<code>SWAP</code>.</p>

Parameters	Types	Description
>> tgtCcy	String	Defines the unit of the "sz" attribute. Only applicable to instType = SPOT. The valid enumerations are <code>base_ccy</code> and <code>quote_ccy</code> . When not specified this is equal to <code>base_ccy</code> by default.
>> tradeQuoteCcy	String	The quote currency used for trading. Only applicable to SPOT. The default value is the quote currency of the instId, for example: for <code>BTC-USD</code> , the default is <code>USD</code> .

## STRUCTURE BLOCK TRADES CHANNEL

Retrieve user's block trades data. All the legs in the same block trade are included in the same update. Data will be pushed whenever there is a block trade that the user is a counterparty for.

### URL PATH

/ws/v5/business (required login)

#### Request Example

```
import asyncio

from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():

    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/business",
        useServerTime=False
    )
    await ws.start()
    args = [
        {
            "channel": "struc-block-trades"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels

Parameter	Type	Required	Description
-----------	------	----------	-------------

> channel	String	Yes	Channel name struc-block-trades
-----------	--------	-----	------------------------------------

### Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "struc-block-trades"
  },
  "connId": "a4d3ae55"
}
```

### Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \\"subscribe\\", \\"argss\\": [{ \\"channel\\": \\"struc-block-trades\\\"}]}",
  "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Operation subscribe unsubscribe error
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name struc-block-trades
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

### Push Data Example

```
{
  "arg": {
    "channel": "struc-block-trades",
    "uid": "77982378738415879"
  },
  "data": [
    {
      "cTime": "1608267227834",
      "rfqId": "18753",
      "clRfqId": "",
      "quoteId": "25092",
      "clQuoteId": "",
      "blockTdId": "180184",
      "tag": "123456",
      "tTraderCode": "ANAND",
      "mTraderCode": "WAGMI",
      "isSuccessful": true,
      "isPartial": false
    }
  ]
}
```

```

"errorCode": "",
"legs": [
  {
    "px": "0.0023",
    "sz": "25.0",
    "instId": "BTC-USD-20220630-60000-C",
    "side": "sell",
    "fee": "0.1001",
    "feeCcy": "BTC",
    "tradeId": "10211",
    "tgtCcy": ""
  },
  {
    "px": "0.0033",
    "sz": "25",
    "instId": "BTC-USD-20220630-50000-C",
    "side": "buy",
    "fee": "0.1001",
    "feeCcy": "BTC",
    "tradeId": "10212",
    "tgtCcy": ""
  }
]
}
]
}
}

```

#### PUSH DATA PARAMETERS

Parameters	Types	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> uid	String	User Identifier
data	Array of objects	Subscribed data
> cTime	String	The time the trade was executed. Unix timestamp in milliseconds.
> rfqid	String	RFQ ID.
> clRfqId	String	Client-supplied RFQ ID. This attribute is treated as client sensitive information. It will not be exposed to the Maker, just return empty string "" for Maker.
> quoteld	String	Quote ID.
> clQuoteld	String	Client-supplied Quote ID. This attribute is treated as client sensitive information. It will not be exposed to the Taker, just return empty string "" for Taker.
> blockTld	String	Block trade ID.
> tag	String	Trade tag. The block trade will have the tag of the RFQ or Quote it corresponds to.
> tTraderCode	String	A unique identifier of the Taker. Empty If anonymous mode of RFQ is <a href="#">True</a> .
> mTraderCode	String	A unique identifier of the Maker. Empty If anonymous mode of Quote is <a href="#">True</a> .
> isSuccessful	Boolean	Whether the trade is filled successfully
> errorCode	String	Error code for unsuccessful trades. It is "" for successful trade.

Parameters	Types	Description
> legs	Array of objects	Legs of trade
>> instId	String	Instrument ID, e.g. BTC-USDT-SWAP
>> px	String	The price the leg executed
>> sz	String	Size of the leg.
>> side	String	The direction of the leg. Valid value can be buy or sell.
>> tgtCcy	String	Defines the unit of the "sz" attribute. Only applicable to instType = SPOT. The valid enumerations are <code>base_ccy</code> and <code>quote_ccy</code> . When not specified this is equal to <code>base_ccy</code> by default.
>> fee	String	Fee. Negative number represents the user transaction fee charged by the platform. Positive fee represents rebate.
>> feeCcy	String	Fee currency
>> tradeld	String	Last traded ID.
> acctAlloc	Array of objects	Applicable to both taker, maker
>> blockTId	String	Block trade ID
>> errorCode	String	Error code for unsuccessful trades. It is "0" for successful trade.
>> acct	String	The name of the allocated account of the RFQOnly applicable to taker, return "" to makers
>> legs	Array of objects	The allocated legs of the account.
>>> instId	String	The Instrument ID of each leg. Example : "BTC-USDT-SWAP"
>>> sz	String	Filled size
>>> tradeld	String	Trade ID
>>> fee	String	Fee
>>> feeCcy	String	Fee currency

#### Group RFQ introduction

1. This endpoint is at parent RFQ level and contains account allocation. For parent RFQ, we should return the actual executed size, i.e. failed execution size should not be included in the parent RFQ level.
2. For account allocation, we should include both filled and failed child RFQ but add an errorCode to indicate whether a child RFQ is filled.
3. Trade results will only be returned to group RFQ creator. Allocated subaccounts and MSAs will not see trade results. Allocated accounts are expected to get these trades through trading bills.
4. Trades data will only be returned after all child RFQs are executed.
5. For parent RFQ isSuccessful field,
  1. it will return true if any child RFQs are filled
  2. otherwise, if all child RFQ fails, it will return false
6. Parent RFQ blockTId or legs tradeld will be empty. However, account allocation breakdown will be offered and tradeld will be attached.

# WebSocket Public Channel

## PUBLIC STRUCTURE BLOCK TRADES CHANNEL

Retrieve the recent block trades data in OKX. All the legs in the same block trade are included in the same update. The data will be pushed 15 minutes after the block trade execution.

### URL PATH

/ws/v5/business

#### Request Example

```
import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/business")
    await ws.start()
    args = [
        {
            "channel": "public-struc-block-trades"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>public-struc-block-trades</code>

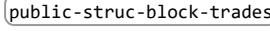
#### Successful Response Example

```
{
    "id": "1512",
    "event": "subscribe",
    "arg": {
        "channel": "public-struc-block-trades"
    },
    "connId": "a4d3ae55"
}
```

#### Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \"subscribe\", \\"argss\\": [{ \\"channel\\": \"public-struc-block-trades\"}]}",
  "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Operation   
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name 
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

## Push Data Example

```
{
  "arg": {
    "channel": "public-struc-block-trades"
  },
  "data": [
    {
      "cTime": "1608267227834",
      "blockTdId": "1802896",
      "groupId": "",
      "legs": [
        {
          "px": "0.323",
          "sz": "25.0",
          "instId": "BTC-USD-20220114-13250-C",
          "side": "sell",
          "tradeId": "15102"
        },
        {
          "px": "0.666",
          "sz": "25",
          "instId": "BTC-USD-20220114-21125-C",
          "side": "buy",
          "tradeId": "15103"
        }
      ]
    }
  ]
}
```

## PUSH DATA PARAMETERS

Parameters	Types	Description
arg	Object	Successfully subscribed channel

Parameters	Types	Description
> channel	String	Channel name
data	Array of objects	Subscribed data
> cTime	String	The time the trade was executed. Unix timestamp in milliseconds.
> blockTdId	String	Block trade ID.
> groupIdx	String	Group RFQ ID Only applicable to group RFQ, return "" for normal RFQ
> legs	Array of objects	Legs of trade
>> instId	String	Instrument ID, e.g. BTC-USDT-SWAP
>> px	String	The price the leg executed
>> sz	String	Trade quantity For spot trading, the unit is base currency For <b>FUTURES</b> / <b>SWAP</b> / <b>OPTION</b> , the unit is contract.
>> side	String	The direction of the leg from the Takers perspective. Valid value can be <b>buy</b> or <b>sell</b> .
>> tradeId	String	Last traded ID.

### Group RFQ introduction

1. Add new response parameter groupIdx, facilitating clients to map subaccount execution to group RFQ. Only applicable to group RFQ, return "" for normal RFQ.
2. Data return by this endpoint should be at \*\*parent RFQ level\*\* regardless of the subaccounts allocation. blockTdId and tradeId will be empty.

## PUBLIC BLOCK TRADES CHANNEL

Retrieve the recent block trades data by individual legs. Each leg in a block trade is pushed in a separate update. The data will be pushed 15 minutes after the block trade execution.

### URL PATH

/ws/v5/business

### Request Example

```
import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/business")
    await ws.start()
    args = [
        {
            "channel": "public-block-trades",
            "instId": "BTC-USDT-SWAP"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)
```

```
asyncio.run(main())
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation subscribe unsubscribe
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name public-block-trades
> instId	String	Yes	Instrument ID, e.g. BTC-USDT-SWAP.

### Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "public-block-trades",
    "instId": "BTC-USDT-SWAP",
    "connId": "a4d3ae55"
  }
}
```

### Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \\"subscribe\\", \\"args\\": [{ \\"channel\\": \\"public-block-trades\\\" }]}",
  "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Operation subscribe unsubscribe error
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name public-block-trades
> instId	String	Yes	Instrument ID, e.g. BTC-USDT-SWAP.
code	String	No	Error code
msg	String	No	Error message

Parameter	Type	Required	Description
connId	String	Yes	WebSocket connection ID

### Push Data Example

```
{
  "arg": {
    "channel": "public-block-trades",
    "instId": "BTC-USD-231020-5000-P"
  },
  "data": [
    {
      "fillVol": "5",
      "fwdPx": "26808.16",
      "groupId": "",
      "idxPx": "27222.5",
      "instId": "BTC-USD-231020-5000-P",
      "markPx": "0.0022406326071111",
      "px": "0.0048",
      "side": "buy",
      "sz": "1",
      "tradeId": "633971452580106242",
      "ts": "1697422572972"
    }
  ]
}
```

### PUSH DATA PARAMETERS

Parameters	Types	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> instId	String	Instrument ID, e.g. BTC-USDT-SWAP.
data	Array of objects	Information of the public trade object.
> instId	String	Instrument ID, e.g. BTC-USDT-SWAP.
> tradeId	String	Trade ID, generated by counter.
> px	String	The price the leg executed.
> sz	String	Trade quantity For spot trading, the unit is base currency For <b>FUTURES</b> / <b>SWAP</b> / <b>OPTION</b> , the unit is contract.
> side	String	Trade direction, buy, sell, from taker perspective.
> fillVol	String	Implied volatility Only applicable to <b>OPTION</b>
> fwdPx	String	Forward price Only applicable to options
> idxPx	String	Index price Applicable to <b>FUTURES</b> , <b>SWAP</b> , <b>OPTION</b>
> markPx	String	Mark price Applicable to <b>FUTURES</b> , <b>SWAP</b> , <b>OPTION</b>
> groupId	String	Group RFQ ID Only applicable to group RFQ, return "" for normal RFQ

Parameters	Types	Description
> ts	String	Filled time, Unix timestamp format in milliseconds, e.g. 1597026383085.

### Group RFQ introduction

1. Add new response parameter groupId, facilitating clients to map subaccount execution to group RFQ. Only applicable to group RFQ, return "" for normal RFQ.
2. Data return by this endpoint should be at \*\*child RFQ execution level\*\* but split into a single leg. tradelId will be populated.

### BLOCK TICKERS CHANNEL

Retrieve the latest block trading volume in the last 24 hours.

The data will be pushed when triggered by transaction execution event. In addition, it will also be pushed in 5 minutes interval according to subscription granularity.

### URL PATH

/ws/v5/business

### Request Example

```
import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/business")
    await ws.start()
    args = [{
        "channel": "block-tickers",
        "instId": "BTC-USDT"
    }]

    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>block-tickers</code>
> instId	String	Yes	Instrument ID e.g. BTC-USDT-SWAP

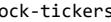
### Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "block-tickers",
    "instId": "LTC-USD-200327"
  },
  "connId": "a4d3ae55"
}
```

### Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\"op\": \"subscribe\", \"args\": [{ \"channel\" : \"block-tickers\", \"instId\" : \"LTC-USD-200327\"}]}",
  "connId": "a4d3ae55"
}
```

### RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event   
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name 
> instId	String	Yes	Instrument ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

### Push Data Example

```
{
  "arg": {
    "channel": "block-tickers"
  },
  "data": [
    {
      "instType": "SWAP",
      "instId": "LTC-USD-SWAP",
      "volCcy24h": "0",
      "vol24h": "0",
      "ts": "1597026383085"
    }
  ]
}
```

### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel

Parameter	Type	Description
> channel	String	Channel name
> instId	String	Instrument ID
data	Array of objects	Subscribed data
> instId	String	Instrument ID
> instType	String	Instrument type
> volCcy24h	String	24h trading volume, with a unit of <code>currency</code> . If it is a <code>derivatives</code> contract, the value is the number of base currency. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in quote currency.
> vol24h	String	24h trading volume, with a unit of <code>contract</code> . If it is a <code>derivatives</code> contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in base currency.
> ts	String	Block ticker data generation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

# Spread Trading

👉 The Spread Orderbook product enables users to **post or consume liquidity** on spreads for **large sizes** that are guaranteed **atomic execution**. Benefits include simplified futures rolls, funding arbitrage, yield enhancement, and speculation on basis and term structures.

## Introduction

### BASIC CONCEPTS

1. **Spread** - Entering a trade where the trader is long one instrument and short an offsetting quantity of a related instrument, forming a trade with two risk offsetting legs.
2. **Order-book** - A collection of offers to trade an instrument or basket. Each offer contains a defined instrument or group of instruments, relevant quantity, and the price at which the offerer is willing to transact. Takers can then immediately consume these offers up to the full amount of quantity listed at the offered price. The pending order limit of spread trading is 500 across all spreads.

### HIGH LEVEL WORKFLOW

Nitro Spreads is centered around the familiar concept of a Central Limit Order Book (**CLOB**).

- Spreads consist of instruments sourced from OKX where they are cleared and settled.
- Anyone can act as a "Taker," who consumes an existing resting order, or a "Maker," whose order is consumed.
- Trades take place when orders are crossed. Trades are then sent for clearing and settlement on OKX.

At a high level, the Nitro Spreads workflow is as follows:

1. *Maker* rests a Limit Order upon a Spread's Order Book.
2. *Taker* consumes a resting Order via a Limit Order.
3. The crossed orders are sent for clearing and settlement.
4. The *Taker* and *Maker* receive confirmation of the success or rejection of the Trade.
5. All users are notified of successfully settled & cleared Trades, minus the counterparties or sides (`buy` / `sell`) involved.

Key aspects of Nitro Spreads:

- All Spreads have **publicly accessible** Central Limit Order Books (**CLOB**).
- The availability of trading Spreads is determined by OKX. Typically, these Spreads encompass all possible combinations of delta one derivatives (Expiry Futures and Perpetual Futures) and SPOT within a specific instrument family (e.g. "BTC/USDT" or "ETH/USDC").

- **Partial fills** and multiple orders can be consumed as part of a single trade.
- **Counterparties** are **NOT** selected. All Spread Order Books can be engaged by anyone, effectively trading against the broader market.
- Anonymity is maintained throughout the process, with all orders and trades conducted on an **anonymous basis**.
- Users have the flexibility to place multiple orders on both the bid and ask sides of the Order Book, allowing for a **ladder-style** configuration.

# Comprehensive API Workflow

Notifications regarding Orders and Trades will be received by both the Taker and the Maker through the WebSocket Notification channels.

A user assumes the role of a *Maker* when their Order is executed upon by another Order. A user becomes a *Taker* when they submit an Order that crosses an existing Order in the Order Book.

## OBTAINING AVAILABLE SPREADS

To retrieve all available Spreads for trading on OKX, make a request to the `GET /api/v5/sprd/spreads` endpoint.

## RETRIEVING YOUR ORDERS

To retrieve orders on OKX, make a request to the `GET /api/v5/sprd/order` endpoint.

## RETRIEVING YOUR TRADES

To retrieve trades on OKX, make a request to the `GET /api/v5/sprd/trades` endpoint.

## SUBMITTING AN ORDER

To submit an order to a Spread's Order Book, make a request to the `POST /api/v5/sprd/order` endpoint.

## SPREAD STATES

There are three different states during a Spread's life cycle: `live`, `suspend`, and `expired` as detailed below:

1. `live`: Spreads that are actively traded on Nitro Spreads
2. `suspend`: Spreads in which at least one of the legs is suspended and the other one is active or suspended on the OKX orderbook exchange; or spreads in which the underlying instruments are still live on the OKX orderbook exchange, but removed from Nitro Spreads
3. `expired`: Spreads in which at least one of the underlying instruments is expired on the OKX orderbook exchange

Please refer to the following table for all possible scenarios given the state of the underlying instruments and the resulting state of the spread on Nitro Spreads (except for the case that the spread is delisted on Nitro Spreads):

Instrument A	Instrument B	Spread State
Live	Live	Live
Suspend	Live	Suspend
Live	Suspend	Suspend
Suspend	Suspend	Suspend
Expired	Live	Expired
Live	Expired	Expired
Suspend	Expired	Expired
Expired	Suspend	Expired
Expired	Expired	Expired

## TRADE LIFECYCLE

In order for a trade to take place, two orders must be crossed within a Spread's Order Book.

Obtain information about the state of an Order and determine if it has reached its final state by monitoring the `sprd-orders` WebSocket channel. The `state` key in the channel indicates the current state of the Order. If the state is `live` or `partially_filled`, it means that the Order still has available size (`sz`) that the creator or another user can take action on. On the other hand, if the state is `canceled` or `filled`, the Order no longer has any available actions that the creator or any other user can take action on.

It is important to closely track the values of the following attributes: `sz`(size), `pendingFillSz` (pending fill size), `canceledSz` (canceled size), and `accFillSz`(accumulated fill size). These attributes provide crucial information regarding the status and progression of the Order.

## ORDER STATE

Track the state of an order by subscribing to the `sprd-orders` WebSocket channel.

1. Upon submitting an order, whether as a Maker or Taker, an order update message is sent via the orders WebSocket channel. The message will indicate the order's `state` == `live`.
2. Order matching and trade settlement are asynchronous processes. When the order is matched but not settled, system pushes `pendingSettleSz > 0` and `fillSz == ""`
3. If the order is partially filled, an order update message is sent with `state` == `partially_filled`.
4. In the event that the order is completely filled, an order update message is sent with the `state` == `filled`.
5. If the order is not fully filled but has reached its final state, an order update message is sent with the `state` == `canceled`.
6. If a certain part of an order is rejected, an order update message is sent with updated `canceledSz` and `pendingFillSz`, and `code` and `msg` corresponding to the error.

## TRADE STATE

Track the state of a trade by subscribing to the `sprd-trades` WebSocket channel.

1. After an executed trade undergoes clearing and settlement on OKX, it reaches finality.
2. For successfully cleared trades, a WebSocket message is sent with the `state` denoted as `filled`.
3. In the case of an unsuccessful trade clearing, a trade update message is sent with the `state` reflected as `rejected`.
4. If the trade state is `rejected`, the trade update message will also include the error `code` and a corresponding error message (`msg`) that explains the reason for the rejection.

## ALL TRADES

All users have the ability to receive updates on all trades that take place through the OKX Nitro Spreads product.

It's important to note that OKX Nitro Spreads does not disclose information about the counterparties involved in the trades or the individual `side` (`buy` or `sell`) of the composite legs that were traded.

1. By subscribing to the `sprd-public-trades` WebSocket channel, WebSocket messages are sent exclusively for trades that have been successfully cleared and settled.

# REST API

## PLACE ORDER

Place a new order

### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: TRADE

### HTTP REQUEST

`POST /api/v5/sprd/order`

Request Example

```
import okx.SpreadTrading as SpreadTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1
```

```

spreadAPI = SpreadTrading.SpreadTradingAPI(apikey, secretkey, passphrase, False, flag)

# place order
result = spreadAPI.place_order(sprdId='BTC-USDT_BTC-USDT-SWAP',
                                clOrdId='b16', side='buy', ordType='limit',
                                px='2', sz='2')
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
sprdId	String	Yes	spread ID, e.g. BTC-USDT_BTC-USDT-SWAP
clOrdId	String	No	Client Order ID as assigned by the client A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
tag	String	No	Order tag A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters.
side	String	Yes	Order side, <code>buy</code> <code>sell</code>
ordType	String	Yes	Order type <code>market</code> : Market order <code>limit</code> : Limit order <code>post_only</code> : Post-only order <code>ioc</code> : Immediate-or-cancel order
sz	String	Yes	Quantity to buy or sell. The unit is USD for inverse spreads, and the corresponding baseCcy for linear and hybrid spreads.
px	String	Yes	Order price. Only applicable to <code>limit</code> , <code>post_only</code> , <code>ioc</code>

## Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "clOrdId": "b15",
      "ordId": "312269865356374016",
      "tag": "",
      "sCode": "0",
      "sMsg": ""
    }
  ]
}
```

## RESPONSE EXAMPLE

Parameter	Type	Description
ordId	String	Order ID
clOrdId	String	Client Order ID as assigned by the client
tag	String	Order tag
sCode	String	The code of the event execution result, 0 means success.
sMsg	String	Rejection or success message of event execution.

### clOrdId

clOrdId is a user-defined unique ID used to identify the order. It will be included in the response parameters if you have specified during order

submission, and can be used as a request parameter to the endpoints to query, cancel and amend orders. `clOrdId` must be unique among the `clOrdIds` of all pending orders.

#### ordType

Order type. When creating a new order, you must specify the order type. The order type you specify will affect: 1) what order parameters are required, and 2) how the matching system executes your order. The following are valid order types:

`limit`: Limit order, which requires specified `sz` and `px`.

`post_only`: Post-only order, which the order can only provide liquidity to the market and be a maker. If the order would have executed on placement, it will be canceled instead. `ioc`: Immediate-or-cancel order

#### sz

The `sz` unit for inverse spreads is USD in Nitro Spread, as opposed to contract in OKX orderbook.

### CANCEL ORDER

Cancel an incomplete order.

#### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: USER ID

#### PERMISSION: TRADE

#### HTTP REQUEST

`POST /api/v5/sprd/cancel-order`

#### Request Example

```
import okx.SpreadTrading as SpreadTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

spreadAPI = SpreadTrading.SpreadTradingAPI(apikey, secretkey, passphrase, False, flag)

# cancel order
result = spreadAPI.cancel_order(ordId='1905309079888199680')
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
<code>ordId</code>	String	Conditional	Order ID Either <code>ordId</code> or <code>clOrdId</code> is required. If both are passed, <code>ordId</code> will be used.
<code>clOrdId</code>	String	Conditional	Client Order ID as assigned by the client

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "clOrdId": "oktswap6",
      "ordId": "12345689",
      "sCode": "0",
      "sMsg": ""
    }
  ]
}
```

]

}

## RESPONSE EXAMPLE

Parameter	Type	Description
ordId	String	Order ID
clOrdId	String	Client Order ID as assigned by the client
sCode	String	The code of the event execution result, 0 means success.
sMsg	String	Rejection message if the request is unsuccessful.

Cancel order returns with sCode equal to 0. It is not strictly considered that the order has been canceled. It only means that your cancellation request has been accepted by the system server. The result of the cancellation is subject to the state pushed by the order channel or the get order state.

## CANCEL ALL ORDERS

Cancel all pending orders.

### RATE LIMIT: 10 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: TRADE

### HTTP REQUEST

POST /api/v5/sprd/mass-cancel

Request Example

```
import okx.SpreadTrading as SpreadTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

spreadAPI = SpreadTrading.SpreadTradingAPI(apikey, secretkey, passphrase, False, flag)

# cancel all
result = spreadAPI.cancel_all_orders(sprdId="BTC-USDT-BTC-USDT-SWAP")
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
sprdId	String	No	spread ID

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "result": true
    }
  ]
}
```

## RESPONSE EXAMPLE

Parameter	Type	Description
result	Boolean	Result of the request <code>true</code> , <code>false</code>

Getting a response with result=true means your request has been successfully received and will be processed. The result of the cancellation is subject to the state pushed by the order channel or the get order state.

## AMEND ORDER

Amend an incomplete order.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

## HTTP REQUEST

`POST /api/v5/sprd/amend-order`

Request Example

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
ordId	String	Conditional	Order ID Either <code>ordId</code> or <code>clOrdId</code> is required. If both are passed, <code>ordId</code> will be used.
clOrdId	String	Conditional	Client Order ID as assigned by the client
reqId	String	No	Client Request ID as assigned by the client for order amendment A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. The response will include the corresponding <code>reqId</code> to help you identify the request if you provide it in the request.
newSz	String	Conditional	New quantity after amendment Either <code>newSz</code> or <code>newPx</code> is required. When amending a partially-filled order, the <code>newSz</code> should include the amount that has been filled.
newPx	String	Conditional	New price after amendment Either <code>newSz</code> or <code>newPx</code> is required.

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "clOrdId": "",
      "ordId": "12344",
      "reqId": "b12344",
      "sCode": "0",
      "sMsg": ""
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ordId	String	Order ID
clOrdId	String	Client Order ID as assigned by the client.
reqId	String	Client Request ID as assigned by the client for order amendment.
sCode	String	The code of the event execution result, 0 means success.
sMsg	String	Rejection message if the request is unsuccessful.

#### newSz

If the new quantity of the order is less than or equal to the (accFillSz + canceledSz + pendingSettleSz), after pendingSettleSz is settled, the order status will be transitioned into filled (if canceledSz = 0), or canceled (if canceledSz > 0).

#### The amend order returns sCode equal to 0

It is not strictly considered that the order has been amended. It only means that your amend order request has been accepted by the system server. The result of the amend is subject to the status pushed by the order channel or the order status query.

### GET ORDER DETAILS

Retrieve order details.

#### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: USER ID

#### PERMISSION: READ

#### HTTP REQUEST

GET /api/v5/sprd/order

#### Request Example

```
import okx.SpreadTrading as SpreadTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

spreadAPI = SpreadTrading.SpreadTradingAPI(apikey, secretkey, passphrase, False, flag)

# get order details
result = spreadAPI.get_order_details(ordId='190530907988199680')
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
ordId	String	Conditional	Order ID Either <code>ordId</code> or <code>clOrdId</code> is required, if both are passed, <code>ordId</code> will be used
clOrdId	String	Conditional	Client Order ID as assigned by the client. The latest order will be returned.

#### Response Example

```
{
  "code": "0",
```

```

"msg": "",
"data": [
{
  "instId": "BTC-USD-200329",
  "ordId": "312269865356374016",
  "clOrdId": "b1",
  "tag": "",
  "px": "999",
  "sz": "3",
  "ordType": "limit",
  "side": "buy",
  "fillSz": "0",
  "fillPx": "",
  "tradeId": "",
  "accFillSz": "0",
  "pendingFillSz": "2",
  "pendingSettleSz": "1",
  "canceledSz": "1",
  "state": "live",
  "avgPx": "0",
  "cancelSource": "",
  "uTime": "1597026383085",
  "cTime": "1597026383085"
}
]
}

```

#### RESPONSE EXAMPLE

Parameter	Type	Description
sprId	String	spread ID
ordId	String	Order ID
clOrdId	String	Client Order ID as assigned by the client
tag	String	Order tag
px	String	Price
sz	String	Quantity to buy or sell
ordType	String	Order type <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">market</div> : Market order <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">limit</div> : Limit order <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">post_only</div> : Post-only order <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">ioc</div> : Immediate-or-cancel order
side	String	Order side
fillSz	String	Last fill quantity
fillPx	String	Last fill price
tradeId	String	Last trade ID
accFillSz	String	Accumulated fill quantity
pendingFillSz	String	Live quantity
pendingSettleSz	String	Quantity that's pending settlement
canceledSz	String	Quantity canceled due order cancellations or trade rejections
avgPx	String	Average filled price. If none is filled, it will return "0".
state	String	State <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">canceled</div>

Parameter	Type	Description
		<input type="checkbox"/> live <input type="checkbox"/> partially_filled <input type="checkbox"/> filled
cancelSource	String	Source of the order cancellation. Valid values and the corresponding meanings are: <input type="checkbox"/> 0: Order canceled by system <input type="checkbox"/> 1: Order canceled by user <input type="checkbox"/> 14: Order canceled: IOC order was partially canceled due to incompletely filled <input type="checkbox"/> 15: Order canceled: The order price is beyond the limit <input type="checkbox"/> 20: Cancel all after triggered <input type="checkbox"/> 31: The post-only order will take liquidity in maker orders <input type="checkbox"/> 32: Self trade prevention <input type="checkbox"/> 34: Order failed to settle due to insufficient margin <input type="checkbox"/> 35: Order cancellation due to insufficient margin from another order <input type="checkbox"/> 44: Your order was canceled because your available balance of this crypto was insufficient for auto conversion. Auto conversion was triggered when the total collateralized liabilities for this crypto reached the platform's risk control limit.
uTime	String	Update time, Unix timestamp format in milliseconds, e.g. <input type="text" value="1597026383085"/>
cTime	String	Creation time, Unix timestamp format in milliseconds, e.g. <input type="text" value="1597026383085"/>

Order sizes equation: pendingFillSz + canceledSz + accFillSz = sz

## GET ACTIVE ORDERS

Retrieve all incomplete orders under the current account.

**RATE LIMIT: 10 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

### HTTP REQUEST

`GET /api/v5/sprd/orders-pending`

Request Example

```

import okx.SpreadTrading as SpreadTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

spreadAPI = SpreadTrading.SpreadTradingAPI(apikey, secretkey, passphrase, False, flag)

# get active orders
result = spreadAPI.get_active_orders()
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
sprdId	String	No	spread ID, e.g.
ordType	String	No	Order type <input type="checkbox"/> market: Market order <input type="checkbox"/> limit: Limit order

Parameter	Type	Required	Description
			<p><code>post_only</code>: Post-only order</p> <p><code>ioc</code>: Immediate-or-cancel order</p>
state	String	No	<p>State</p> <p><code>live</code></p> <p><code>partially_filled</code></p>
beginId	String	No	Start order ID the request to begin with. Pagination of data to return records newer than the requested order Id, not including beginId
endId	String	No	End order ID the request to end with. Pagination of data to return records earlier than the requested order Id, not including endId
limit	String	No	Number of results per request. The maximum is 100. The default is 100

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "sprdId": "BTC-USDT_BTC-UST-SWAP",
      "ordId": "312269865356374016",
      "clOrdId": "b1",
      "tag": "",
      "px": "999",
      "sz": "3",
      "ordType": "limit",
      "side": "buy",
      "fillSz": "0",
      "fillPx": "",
      "tradeId": "",
      "accFillSz": "0",
      "pendingFillSz": "2",
      "pendingSettleSz": "1",
      "canceledSz": "1",
      "state": "live",
      "avgPx": "0",
      "cancelSource": "",
      "uTime": "1597026383085",
      "cTime": "1597026383085"
    }
  ]
}
```

#### RESPONSE EXAMPLE

Parameter	Type	Description
sprdId	String	spread ID
ordId	String	Order ID
clOrdId	String	Client Order ID as assigned by the client
tag	String	Order tag
px	String	Price
sz	String	Quantity to buy or sell

Parameter	Type	Description
ordType	String	Order type <code>market</code> : Market order <code>limit</code> : Limit order <code>post_only</code> : Post-only order <code>ioc</code> : Immediate-or-cancel order
side	String	Order side
fillSz	String	Last fill quantity
fillPx	String	Last fill price
tradeId	String	Last trade ID
accFillSz	String	Accumulated fill quantity
pendingFillSz	String	Quantity still remaining to be filled
pendingSettleSz	String	Quantity that's pending settlement
canceledSz	String	Quantity canceled due order cancellations or trade rejections
avgPx	String	Average filled price. If none is filled, it will return "0".
state	String	State <code>live</code> <code>partially_filled</code>
cancelSource	String	Source of the order cancellation. Valid values and the corresponding meanings are: <code>0</code> : Order canceled by system <code>1</code> : Order canceled by user <code>14</code> : Order canceled: IOC order was partially canceled due to incompletely filled <code>15</code> : Order canceled: The order price is beyond the limit <code>20</code> : Cancel all after triggered <code>31</code> : The post-only order will take liquidity in maker orders <code>32</code> : Self trade prevention <code>34</code> : Order failed to settle due to insufficient margin <code>35</code> : Order cancellation due to insufficient margin from another order <code>44</code> : Your order was canceled because your available balance of this crypto was insufficient for auto conversion. Auto conversion was triggered when the total collateralized liabilities for this crypto reached the platform's risk control limit.
uTime	String	Update time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
cTime	String	Creation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

#### GET ORDERS (LAST 21 DAYS)

Retrieve the completed order data for the last 21 days, and the incomplete orders (filledSz =0 & state = canceled) that have been canceled are only reserved for 2 hours. Results are returned in counter chronological order of orders creation.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/sprd/orders-history`

Request Example

```
import okx.SpreadTrading as SpreadTrading
```

```

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

spreadAPI = SpreadTrading.SpreadTradingAPI(apikey, secretkey, passphrase, False, flag)

# get orders history
result = spreadAPI.get_orders()
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
sprdId	String	No	spread ID, e.g.
ordType	String	No	Order type <input type="button" value="market"/> : Market order <input type="button" value="limit"/> : limit order <input type="button" value="post_only"/> : Post-only order <input type="button" value="ioc"/> : Immediate-or-cancel order
state	String	No	State <input type="button" value="canceled"/> <input type="button" value="filled"/>
beginId	String	No	Start order ID the request to begin with. Pagination of data to return records newer than the requested order Id, not including beginId
endId	String	No	End order ID the request to end with. Pagination of data to return records earlier than the requested order Id, not including endId
begin	String	No	Filter with a begin timestamp. Unix timestamp format in milliseconds, e.g. <input type="button" value="1597026383085"/> . Date older than 7 days will be truncated.
end	String	No	Filter with an end timestamp. Unix timestamp format in milliseconds, e.g. <input type="button" value="1597026383085"/>
limit	String	No	Number of results per request. The maximum is 100. The default is 100

## Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "sprdId": "BTC-USDT_BTC-UST-SWAP",
      "ordId": "312269865356374016",
      "clOrdId": "b1",
      "tag": "",
      "px": "999",
      "sz": "3",
      "ordType": "limit",
      "side": "buy",
      "fillSz": "0",
      "fillPx": "",
      "tradeId": "",
      "accFillSz": "0",
      "pendingFillSz": "2",
      "pendingSettleSz": "1",
      "canceledSz": "1",
      "state": "live",
      "avgPx": "0",
      "cancelSource": "",
      "uTime": "1597026383085",
      "cTime": "1597026383085"
    }
  ]
}
```

```

}
]
}
}
```

## RESPONSE EXAMPLE

Parameter	Type	Description
sprId	String	spread ID
ordId	String	Order ID
clOrdId	String	Client Order ID as assigned by the client
tag	String	Order tag
px	String	Price
sz	String	Quantity to buy or sell
ordType	String	Order type <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px;">market</span> : Market order <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px;">limit</span> : limit order <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px;">post_only</span> : Post-only order <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px;">ioc</span> : Immediate-or-cancel order
side	String	Order side
fillSz	String	Last fill quantity
fillPx	String	Last fill price
tradeId	String	Last trade ID
accFillSz	String	Accumulated fill quantity
pendingFillSz	String	Quantity still remaining to be filled, including pendingSettleSz
pendingSettleSz	String	Quantity that's pending settlement
canceledSz	String	Quantity canceled due to order cancellations or trade rejections
avgPx	String	Average filled price. If none is filled, it will return "0".
state	String	State <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px;">canceled</span> <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px;">filled</span>
cancelSource	String	Source of the order cancellation. Valid values and the corresponding meanings are: <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px;">0</span> : Order canceled by system <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px;">1</span> : Order canceled by user <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px;">14</span> : Order canceled: IOC order was partially canceled due to incompletely filled <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px;">15</span> : Order canceled: The order price is beyond the limit <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px;">20</span> : Cancel all after triggered <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px;">31</span> : The post-only order will take liquidity in maker orders <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px;">32</span> : Self trade prevention <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px;">34</span> : Order failed to settle due to insufficient margin <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px;">35</span> : Order cancellation due to insufficient margin from another order <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px;">44</span> : Your order was canceled because your available balance of this crypto was insufficient for auto conversion. Auto conversion was triggered when the total collateralized liabilities for this crypto reached the platform's risk control limit.
uTime	String	Update time, Unix timestamp format in milliseconds, e.g. <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px;">1597026383085</span>
cTime	String	Creation time, Unix timestamp format in milliseconds, e.g. <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px;">1597026383085</span>

## GET ORDERS HISTORY (LAST 3 MONTHS)

Retrieve the completed order data for the last 3 months, including those placed 3 months ago but completed in the last 3 months. Results are returned in counter chronological order.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/sprd/orders-history-archive`

Request Example

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
sprdId	String	No	spread ID, e.g.
ordType	String	No	Order type <code>market</code> : Market order <code>limit</code> : limit order <code>post_only</code> : Post-only order <code>ioc</code> : Immediate-or-cancel order
state	String	No	State <code>canceled</code> <code>filled</code>
instType	String	No	Instrument type <code>SPOT</code> <code>FUTURES</code> <code>SWAP</code> Any orders with spreads containing the specified instrument type in any legs will be returned
instFamily	String	No	Instrument family, e.g. BTC-USDT. Any orders with spreads containing the specified instrument family in any legs will be returned
beginId	String	No	Start order ID the request to begin with. Pagination of data to return records newer than the requested order Id, not including beginId
endId	String	No	End order ID the request to end with. Pagination of data to return records earlier than the requested order Id, not including endId
begin	String	No	Filter with a begin timestamp. Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
end	String	No	Filter with an end timestamp. Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. The maximum is 100. The default is 100

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "sprdId": "BTC-USDT_BTC-UST-SWAP",
      "ordId": "312269865356374016",
      "c1OrdId": "b1",
      "tag": "",
      "px": "999",
      "sz": "3",
      "ordType": "limit",
      "side": "buy",
      "fillSz": "0",
      "status": "filled",
      "time": "2020-09-15T14:00:00Z"
    }
  ]
}
```

```

"fillPx": "",
"tradeId": "",
"accFillSz": "0",
"pendingFillSz": "2",
"pendingSettleSz": "1",
"canceledSz": "1",
"state": "canceled",
"avgPx": "0",
"cancelSource": "",
"uTime": "1597026383085",
"cTime": "1597026383085"
}
]
}

```

#### RESPONSE EXAMPLE

Parameter	Type	Description
sprId	String	spread ID
ordId	String	Order ID
clOrdId	String	Client Order ID as assigned by the client
tag	String	Order tag
px	String	Price
sz	String	Quantity to buy or sell
ordType	String	Order type <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; display: inline-block;">market</span> : Market order <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; display: inline-block;">limit</span> : limit order <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; display: inline-block;">post_only</span> : Post-only order <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; display: inline-block;">ioc</span> : Immediate-or-cancel order
side	String	Order side
fillSz	String	Last fill quantity
fillPx	String	Last fill price
tradeId	String	Last trade ID
accFillSz	String	Accumulated fill quantity
pendingFillSz	String	Quantity still remaining to be filled, including pendingSettleSz
pendingSettleSz	String	Quantity that's pending settlement
canceledSz	String	Quantity canceled due to order cancellations or trade rejections
avgPx	String	Average filled price. If none is filled, it will return "0".
state	String	State <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; display: inline-block;">canceled</span> <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; display: inline-block;">filled</span>
cancelSource	String	Source of the order cancellation. Valid values and the corresponding meanings are: <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; display: inline-block;">0</span> : Order canceled by system <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; display: inline-block;">1</span> : Order canceled by user <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; display: inline-block;">14</span> : Order canceled: IOC order was partially canceled due to incompletely filled <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; display: inline-block;">15</span> : Order canceled: The order price is beyond the limit <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; display: inline-block;">20</span> : Cancel all after triggered <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; display: inline-block;">31</span> : The post-only order will take liquidity in maker orders <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 5px; display: inline-block;">32</span> : Self trade prevention

Parameter	Type	Description
		<p>34: Order failed to settle due to insufficient margin</p> <p>35: Order cancellation due to insufficient margin from another order</p> <p>44: Your order was canceled because your available balance of this crypto was insufficient for auto conversion. Auto conversion was triggered when the total collateralized liabilities for this crypto reached the platform's risk control limit.</p>
uTime	String	Update time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
cTime	String	Creation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

#### GET TRADES (LAST 7 DAYS)

Retrieve historical transaction details **for the last 7 days**. Results are returned in counter chronological order.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

#### HTTP REQUEST

`GET /api/v5/sprd/trades`

#### Request Example

```
import okx.SpreadTrading as SpreadTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

spreadAPI = SpreadTrading.SpreadTradingAPI(apikey, secretkey, passphrase, False, flag)

# get private trades
result = spreadAPI.get_trades()
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
sprdId	String	No	spread ID, e.g.
tradeld	String	No	Trade ID
ordId	String	No	Order ID
beginId	String	No	Start trade ID the request to begin with. Pagination of data to return records newer than the requested tradeld, not including beginId
endId	String	No	End trade ID the request to end with. Pagination of data to return records earlier than the requested tradeld, not including endId
begin	String	No	Filter with a begin timestamp. Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
end	String	No	Filter with an end timestamp. Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. The maximum is 100. The default is 100

#### Response Example

```
{
  "code": "0",
```

```

"msg": "",
"data": [
  {
    "sprdId": "BTC-USDT-SWAP_BTC-USDT-200329",
    "tradeId": "123",
    "ordId": "123445",
    "clOrdId": "b16",
    "tag": "",
    "fillPx": "999",
    "fillSz": "3",
    "state": "filled",
    "side": "buy",
    "execType": "M",
    "ts": "1597026383085",
    "legs": [
      {
        "instId": "BTC-USDT-SWAP",
        "px": "20000",
        "sz": "3",
        "szCont": "0.03",
        "side": "buy",
        "fillPnl": "",
        "fee": "",
        "feeCcy": "",
        "tradeId": "1232342342"
      },
      {
        "instId": "BTC-USDT-200329",
        "px": "21000",
        "sz": "3",
        "szCont": "0.03",
        "side": "sell",
        "fillPnl": "",
        "fee": "",
        "feeCcy": "",
        "tradeId": "5345646634"
      }
    ],
    "code": "",
    "msg": ""
  }
]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
sprdId	String	spread ID
tradeld	String	Trade ID
ordId	String	Order ID
clOrdId	String	Client Order ID as assigned by the client
tag	String	Order tag
fillPx	String	Filled price
fillSz	String	Filled quantity
side	String	Order side, <code>buy</code> <code>sell</code>
state	String	Trade state. Valid values are <code>filled</code> and <code>rejected</code>
execType	String	Liquidity taker or maker, <code>T</code> : taker <code>M</code> : maker
ts	String	Data generation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code> .

Parameter	Type	Description
legs	Array of objects	Legs of trade
> instId	String	Instrument ID, e.g. BTC-USDT-SWAP
> px	String	The price the leg executed
> sz	String	The size of each leg
> szCont	String	Filled amount of the contract Only applicable to contracts, return "" for spot
> side	String	The direction of the leg. Valid value can be <code>buy</code> or <code>sell</code> .
> fillPnl	String	Last filled profit and loss, applicable to orders which have a trade and aim to close position. It always is 0 in other conditions
> fee	String	Fee. Negative number represents the user transaction fee charged by the platform. Positive number represents rebate.
> feeCcy	String	Fee currency
> tradeId	String	Traded ID in the OKX orderbook.
code	String	Error Code, the default is 0
msg	String	Error Message, the default is ""

#### GET SPREADS (PUBLIC)

Retrieve all available spreads based on the request parameters.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/sprd/spreads`

Request Example

```
import okx.SpreadTrading as SpreadTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

spreadAPI = SpreadTrading.SpreadTradingAPI(apikey, secretkey, passphrase, False, flag)

# get spreads
result = spreadAPI.get_spreads()
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
baseCcy	string	No	Currency instrument is based in, e.g. BTC, ETH
instId	String	No	The instrument ID to be included in the spread.

Parameter	Type	Required	Description
sprdId	String	No	The spread ID
state	string	No	Spreads which are available to trade, suspended or expired. Valid values include <code>live</code> , <code>suspend</code> and <code>expired</code> .

### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "sprdId": "ETH-USD-SWAP_ETH-USD-231229",
      "sprdType": "inverse",
      "state": "live",
      "baseCcy": "ETH",
      "szCcy": "USD",
      "quoteCcy": "USD",
      "tickSz": "0.01",
      "minSz": "10",
      "lotSz": "10",
      "listTime": "1686903000159",
      "legs": [
        {
          "instId": "ETH-USD-SWAP",
          "side": "sell"
        },
        {
          "instId": "ETH-USD-231229",
          "side": "buy"
        }
      ],
      "expTime": "170383680000",
      "uTime": "1691376905595"
    },
    {
      "sprdId": "BTC-USDT_BTC-USDT-SWAP",
      "sprdType": "linear",
      "state": "live",
      "baseCcy": "BTC",
      "szCcy": "BTC",
      "quoteCcy": "USDT",
      "tickSz": "0.0001",
      "minSz": "0.001",
      "lotSz": "1",
      "listTime": "1597026383085",
      "expTime": "1597029999085",
      "uTime": "1597028888085",
      "legs": [
        {
          "instId": "BTC-USDT",
          "side": "sell"
        },
        {
          "instId": "BTC-USDT-SWAP",
          "side": "buy"
        }
      ]
    },
    {
      "sprdId": "BTC-USDT_BTC-USDT-230317",
      "sprdType": "linear",
      "state": "live",
      "baseCcy": "BTC",
      "szCcy": "BTC",
      "quoteCcy": "USDT",
      "tickSz": "0.0001",
      "minSz": "0.001",
      "lotSz": "1",
      "listTime": "1597026383085",
      "expTime": "1597029999085",
      "uTime": "1597028888085",
      "legs": [
        {
          "instId": "BTC-USDT",
          "side": "sell"
        }
      ]
    }
  ]
}
```

```

    "instId": "BTC-USDT-230317",
    "side": "buy"
  }
]
}
}
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
sprdId	String	spread ID
sprdType	String	spread Type. Valid values are <code>linear</code> , <code>inverse</code> , <code>hybrid</code>
state	String	Current state of the spread. Valid values include <code>live</code> , <code>expired</code> , <code>suspend</code> .
baseCcy	String	Currency instrument is based in. Valid values include BTC, ETH
szCcy	String	The currency the spread order size is submitted to the underlying venue in, e.g. USD, BTC, ETH.
quoteCcy	String	The currency the spread is priced in, e.g. USDT, USD
tickSz	String	Tick size, e.g. 0.0001 in the quoteCcy of the spread.
minSz	String	Minimum order size in the szCcy of the spread.
lotSz	String	The minimum order size increment the spread can be traded in the szCcy of the spread.
listTime	String	The timestamp the spread was created. Unix timestamp format in milliseconds, , e.g. <code>1597026383085</code>
expTime	String	Expiry time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
uTime	String	The timestamp the spread was last updated. Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
legs	array of objects	
> instId	String	Instrument ID, e.g. BTC-USD-SWAP
> side	String	The direction of the leg of the spread. Valid Values include <code>buy</code> and <code>sell</code> .

## GET ORDER BOOK (PUBLIC)

Retrieve the order book of the spread.

### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: IP

### PERMISSION: READ

### HTTP REQUEST

```
GET /api/v5/sprd/books
```

### Request Example

```

import okx.SpreadTrading as SpreadTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

spreadAPI = SpreadTrading.SpreadTradingAPI(apikey, secretkey, passphrase, False, flag)

```

```
# get order book
result = spreadAPI.get_order_book(sprdId="BTC-USDT_BTC-USDT-SWAP", sz=20)
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
sprdId	String	Yes	spread ID, e.g. BTC-USDT_BTC-USDT-SWAP
sz	String	No	Order book depth per side. Maximum value is 400. Default value is 5.

## Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "asks": [
        [
          "41006.8", // price
          "0.60038921", // quantity
          "1" // number of orders at the price
        ]
      ],
      "bids": [
        [
          "41006.3",
          "0.30178218",
          "2"
        ]
      ],
      "ts": "1629966436396"
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
asks	Array of arrays	Order book on sell side
bids	Array of arrays	Order book on buy side
ts	String	Order book generation time

An example of the array of asks and bids values: ["411.8", "10", "4"]

- "411.8" is the depth price
- "10" is the quantity at the price (Unit: szCcy)
- "4" is the number of orders at the price.

## GET TICKER (PUBLIC)

Retrieve the latest price snapshot, best bid/ask price and quantity.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

## HTTP REQUEST

```
GET /api/v5/market/sprd-ticker
```

## Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
sprdId	String	Yes	spread ID, e.g. BTC-USDT_BTC-USDT-SWAP

## Response Example

```
{  
  "code": "0",  
  "msg": "",  
  "data": [  
    {  
      "sprdId": "BTC-USDT_BTC-USDT-SWAP",  
      "last": "14.5",  
      "lastSz": "0.5",  
      "askPx": "8.5",  
      "askSz": "12.0",  
      "bidPx": "0.5",  
      "bidSz": "12.0",  
      "open24h": "4",  
      "high24h": "14.5",  
      "low24h": "-2.2",  
      "vol24h": "6.67",  
      "ts": "1715331406485"  
    }  
  ]  
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
sprdId	String	spread ID
last	String	Last traded price
lastSz	String	Last traded size
askPx	String	Best ask price
askSz	String	Best ask size
bidPx	String	Best bid price
bidSz	String	Best bid size
open24h	String	Open price in the past 24 hours
high24h	String	Highest price in the past 24 hours
low24h	String	Lowest price in the past 24 hours
vol24h	String	24h trading volume The unit is USD for inverse spreads, and the corresponding baseCcy for linear and hybrid spreads.
ts	String	Ticker data generation time, Unix timestamp format in milliseconds, e.g. 1597026383085.

### GET PUBLIC TRADES (PUBLIC)

Retrieve the recent transactions of an instrument (at most 500 records per request). Results are returned in counter chronological order.

#### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: IP

#### PERMISSION: READ

## HTTP REQUEST

GET /api/v5/sprd/public-trades

### Request Example

```
import okx.SpreadTrading as SpreadTrading

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading:0 , demo trading:1

spreadAPI = SpreadTrading.SpreadTradingAPI(apikey, secretkey, passphrase, False, flag)

# get public trades
result = spreadAPI.get_public_trades(sprdId='ETH-USDT-SWAP_ETH-USDT-230929')
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
sprdId	String	No	Spread ID, e.g. BTC-USDT_BTC-USDT-SWAP

### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "sprdId": "BTC-USDT_BTC-USDC-SWAP",
      "side": "sell",
      "sz": "0.1",
      "px": "964.1",
      "tradeId": "242720719",
      "ts": "1654161641568"
    }
  ]
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
sprdId	String	spread ID
tradId	String	Trade ID
px	String	Trade price
sz	String	Trade quantity
side	String	Trade side of the taker. <input type="button" value="buy"/> <input type="button" value="sell"/>
ts	String	Trade time, Unix timestamp format in milliseconds, e.g. <input type="text" value="1597026383085"/> .

### GET CANDLESTICKS

Retrieve the candlestick charts. This endpoint can retrieve the latest 1,440 data entries. Charts are returned in groups based on the requested bar.

**RATE LIMIT: 40 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

## HTTP REQUEST

```
GET /api/v5/market/sprd-candles
```

### Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
sprdId	String	Yes	Spread ID
bar	String	No	Bar size, the default is 1m, e.g. [1m/3m/5m/15m/30m/1H/2H/4H] UTC+8 opening price k-line:[6H/12H/1D/2D/3D/1W/1M/3M] UTC+0 opening price k-line:[/6Hutc/12Hutc/1Dutc/2Dutc/3Dutc/1Wutc/1Mutc/3Mutc]
after	String	No	Pagination of data to return records earlier than the requested ts
before	String	No	Pagination of data to return records newer than the requested ts. The latest data will be returned when using before individually
limit	String	No	Number of results per request. The maximum is 300. The default is 100.

### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    [
      "1597026383085",
      "3.721",
      "3.743",
      "3.677",
      "3.708",
      "8422410",
      "0"
    ],
    [
      "1597026383085",
      "3.731",
      "3.799",
      "3.494",
      "3.72",
      "24912403",
      "1"
    ]
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Opening time of the candlestick, Unix timestamp format in milliseconds, e.g. 1597026383085
o	String	Open price
h	String	Highest price
l	String	Lowest price
c	String	Close price
vol	String	Trading volume

Parameter	Type	Description
confirm	String	The state of candlesticks. ① represents that it is uncompleted ② represents that it is completed.

The first candlestick data may be incomplete, and should not be polled repeatedly.

The data returned will be arranged in an array like this: [ts,o,h,l,c,vol,confirm].

#### GET CANDLESTICKS HISTORY

Retrieve history candlestick charts from recent years.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

**HTTP REQUEST**

`GET /api/v5/market/sprd-history-candles`

Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
sprdId	String	Yes	Spread ID
after	String	No	Pagination of data to return records earlier than the requested ts
before	String	No	Pagination of data to return records newer than the requested ts. The latest data will be returned when using before individually
bar	String	No	Bar size, the default is 1m, e.g. [1m/3m/5m/15m/30m/1H/2H/4H] UTC+8 opening price k-line:[6H/12H/1D/2D/3D/1W/1M/3M] UTC+0 opening price k-line:[6Hutc/12Hutc/1Dutc/2Dutc/3Dutc/1Wutc/1Mutc/3Mutc]
limit	String	No	Number of results per request. The maximum is 100. The default is 100.

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    [
      "1597026383085",
      "3.721",
      "3.743",
      "3.677",
      "3.708",
      "8422410",
      "1"
    ],
    [
      "1597026383085",
      "3.731",
      "3.799",
      "3.494",
      "3.72",
      "24912403",
      "1"
    ]
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Opening time of the candlestick, Unix timestamp format in milliseconds, e.g. 1597026383085
o	String	Open price
h	String	Highest price
l	String	Lowest price
c	String	Close price
vol	String	Trading volume
confirm	String	The state of candlesticks. ① represents that it is uncompleted ② represents that it is completed.

The data returned will be arranged in an array like this: [ts,o,h,l,c,vol,confirm]

## CANCEL ALL AFTER

Cancel all pending orders after the countdown timeout. Only applicable to spread trading.

### RATE LIMIT: 1 REQUEST PER SECOND

### RATE LIMIT RULE: USER ID

### PERMISSION: TRADE

### HTTP REQUEST

POST /api/v5/sprd/cancel-all-after

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
timeOut	String	Yes	The countdown for order cancellation, with second as the unit. Range of value can be 0, [10, 120]. Setting timeOut to 0 disables Cancel All After.

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "triggerTime": "1587971460",
      "ts": "1587971400"
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
triggerTime	String	The time the cancellation is triggered. triggerTime=0 means Cancel All After is disabled.
ts	String	The time the request is received.

Users are recommended to send a request to the exchange every second. When the cancel all after is triggered, the trading engine will cancel orders on behalf of the client one by one and this operation may take up to a few seconds. This feature is intended as a protection mechanism for clients only and clients should not use this feature as part of their trading strategies.

## WebSocket Trade API

### WS / PLACE ORDER

You can place an order only if you have sufficient funds.

#### URL PATH

/ws/v5/business (required login)

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

Rate limit is shared with the Nitro Spread 'Place order' REST API endpoints

#### Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	Yes	Unique identifier of the message provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation sprd-order
args	Array of objects	Yes	Request parameters
> sprdId	String	Yes	spread ID, e.g. BTC-USDT_BTC-USD-SWAP
> clOrdId	String	No	Client Order ID as assigned by the client A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
> tag	String	No	Order tag A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters.
> side	String	Yes	Order side buy sell
> ordType	String	Yes	Order type: market: Market order limit: Limit order post_only: Post-only order ioc: Immediate-or-cancel order
> sz	String	Yes	Quantity to buy or sell
> px	String	Yes	Order price. Only applicable to limit, post_only, ioc order.

#### SUCCESSFUL RESPONSE EXAMPLE

```
{
  "id": "1512",
  "op": "sprd-order",
  "data": [
    {
      "clOrdId": "",
      "ordId": "12345689",
      "tag": "",
      "sCode": "0",
      "sMsg": ""
    }
  ],
  "code": "0",
  "msg": ""
}
```

### Failure Response Example

```
{
  "id": "1512",
  "op": "sprd-order",
  "data": [
    {
      "clOrdId": "",
      "ordId": "",
      "tag": "",
      "sCode": "5XXX",
      "sMsg": "not exist"
    }
  ],
  "code": "1",
  "msg": ""
}
```

### Response Example When Format Error

```
{
  "id": "1512",
  "op": "sprd-order",
  "data": [],
  "code": "60013",
  "msg": "Invalid args"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
id	String	Unique identifier of the message
op	String	Operation
code	String	Error Code
msg	String	Error message
data	Array of objects	Data
> ordId	String	Order ID
> clOrdId	String	Client Order ID as assigned by the client
> tag	String	Order tag
> sCode	String	Order status code, 0 means success
> sMsg	String	Rejection or success message of event execution.

## clOrdId

clOrdId is a user-defined unique ID used to identify the order. It will be included in the response parameters if you have specified during order submission, and can be used as a request parameter to the endpoints to query, cancel and amend orders.

clOrdId must be unique among the clOrdIds of all pending orders.

## WS / AMEND ORDER

Amend an incomplete order.

### URL PATH

/ws/v5/business (required login)

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

Rate limit is shared with the `Amend order` REST API endpoints

### Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	Yes	Unique identifier of the messageProvided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation sprd-amend-order
args	Array of objects	Yes	Request Parameters
> ordId	String	Conditional	Order ID Either <code>ordId</code> or <code>clOrdId</code> is required, if both are passed, <code>ordId</code> will be used.
> clOrdId	String	Conditional	Client Order ID as assigned by the client
> reqId	String	No	Client Request ID as assigned by the client for order amendment A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
> newSz	String	Conditional	New quantity after amendment. Either <code>newSz</code> or <code>newPx</code> is required. When amending a partially-filled order, the newSz should include the amount that has been filled and failed.
> newPx	String	Conditional	New price after amendment.

### Successful Response Example

```
{  
  "id": "1512",  
  "op": "sprd-amend-order",  
  "data": [  
    {  
      "clOrdId": "",  
      "ordId": "2510789768709120",  
      "reqId": "b12344",  
      "sCode": "0",  
      "sMsg": ""  
    }  
,  
    {"code": "0",  
     "msg": ""  
  ]  
}
```

## Failure Response Example

```
{  
  "id": "1512",  
  "op": "sprd-amend-order",  
  "data": [  
    {  
      "clOrdId": "",  
      "ordId": "2510789768709120",  
      "reqId": "b12344",  
      "sCode": "5XXXX",  
      "sMsg": "order not exist"  
    }  
,  
  ],  
  "code": "1",  
  "msg": ""  
}
```

## Response Example When Format Error

```
{  
  "id": "1512",  
  "op": "sprd-amend-order",  
  "data": [],  
  "code": "60013",  
  "msg": "Invalid args"  
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
id	String	Unique identifier of the message
op	String	Operation
code	String	Error Code
msg	String	Error message
data	Array of objects	Data
> ordId	String	Order ID
> clOrdId	String	Client Order ID as assigned by the client
> reqId	String	Client Request ID as assigned by the client for order amendment
> sCode	String	Order status code, 0 means success
> sMsg	String	Order status message

### newSz

If the new quantity of the order is less than or equal to the (accFillSz + canceledSz + pendingSettleSz), after pendingSettleSz is settled, the order status will be transitioned into filled (if canceledSz = 0), or canceled (if canceledSz > 0).

### The amend order returns sCode equal to 0

It is not strictly considered that the order has been amended. It only means that your amend order request has been accepted by the system

## WS / CANCEL ORDER

Cancel an incomplete order

### URL PATH

/ws/v5/business (required login)

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

Rate limit is shared with the Nitro Spread 'Cancel order' REST API endpoints

### Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	Yes	Unique identifier of the message provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation sprd-cancel-order
args	Array of objects	Yes	Request Parameters
> ordId	String	Conditional	Order ID Either ordId or clOrdId is required, if both are passed, ordId will be used
> clOrdId	String	Conditional	Client Order ID as assigned by the client A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.

### Successful Response Example

```
{
  "id": "1514",
  "op": "sprd-cancel-order",
  "data": [
    {
      "clOrdId": "",
      "ordId": "2510789768709120",
      "sCode": "0",
      "sMsg": ""
    }
  ],
  "code": "0",
  "msg": ""
}
```

### Failure Response Example

```
{
  "id": "1514",
  "op": "sprd-cancel-order",
  "data": [
    {
      "clOrdId": "",
      "ordId": "2510789768709120",
      "sCode": "5XXX",
      "sMsg": "Order not exist"
    }
  ],
}
```

```

"code": "1",
"msg": ""
}

```

## Response Example When Format Error

```

{
  "id": "1514",
  "op": "sprd-cancel-order",
  "data": [],
  "code": "60013",
  "msg": "Invalid args"
}

```

### RESPONSE PARAMETERS

Parameter	Type	Description
id	String	Unique identifier of the message
op	String	Operation
code	String	Error Code
msg	String	Error message
data	Array of objects	Data
> ordId	String	Order ID
> clOrdId	String	Client Order ID as assigned by the client
> sCode	String	Order status code, 0 means success
> sMsg	String	Order status message

Cancel order returns with sCode equal to 0. It is not strictly considered that the order has been canceled. It only means that your cancellation request has been accepted by the system server. The result of the cancellation is subject to the state pushed by the sprd-orders channel or the get order state.

### WS / CANCEL ALL ORDERS

#### URL PATH

/ws/v5/business (required login)

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	Yes	Unique identifier of the message provided by client. It will be returned in response message to identify the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation sprd-mass-cancel
args	Array of objects	Yes	Request parameters

Parameter	Type	Required	Description
> sprdId	String	No	spread ID

#### SUCCESSFUL RESPONSE EXAMPLE

```
{
  "id": "1512",
  "op": "sprd-mass-cancel",
  "data": [
    {
      "result": true
    }
  ],
  "code": "0",
  "msg": ""
}
```

#### Response Example When Format Error

```
{
  "id": "1512",
  "op": "sprd-mass-cancel",
  "data": [],
  "code": "60013",
  "msg": "Invalid args"
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
id	String	Unique identifier of the message
op	String	Operation
code	String	Error Code
msg	String	Error message
data	Array of objects	Data
> result	Boolean	Result of the request [true], [false]

## WebSocket Private Channel

- Production Trading URL: <wss://ws.okx.com:8443/ws/v5/business>
- Demo Trading URL: <wss://wspap.okx.com:8443/ws/v5/business>

#### ORDER CHANNEL

Retrieve order information from the `sprd-order` Websocket channel. Data will not be pushed when first subscribed. Data will only be pushed when triggered by events such as placing/canceling order.

#### URL PATH

/ws/v5/business (required login)

Request Example : single

```
import asyncio
from okx.websocket.WsPrivateAsync import WsPrivateAsync
```

```

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/business",
        useServerTime=False
    )
    await ws.start()
    args = [
        {
            "channel": "sprd-orders",
            "sprdId": "BTC-USDT_BTC-USDT-SWAP"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())

```

### Request Example:

```

import asyncio
from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/business",
        useServerTime=False
    )
    await ws.start()
    args = [
        {
            "channel": "sprd-orders",
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())

```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>

Parameter	Type	Required	Description
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name sprd-orders
> sprdId	String	No	Spread ID

Successful Response Example : single

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "sprd-orders",
    "sprdId": "BTC-USDT_BTC-UST-SWAP"
  },
  "connId": "a4d3ae55"
}
```

Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "sprd-orders"
  },
  "connId": "a4d3ae55"
}
```

Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\"op\": \"subscribe\", \"argss\": [{ \"channel\": \"sprd-orders\", \"instType\": \"FUTURES\"}]}",
  "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Required	Type	Description
id	String	No	Unique identifier of the message
event	Yes	String	Event subscribe unsubscribe error
arg	No	Object	Subscribed channel
> channel	Yes	String	Channel name
> sprdId	No	String	Spread ID
code	No	String	Error code
msg	No	String	Error message
connId	String	Yes	WebSocket connection ID

```
{
  "arg": {
    "channel": "sprd-orders",
    "sprdId": "BTC-USDT_BTC-USDT-SWAP",
    "uid": "614488474791936"
  },
  "data": [
    {
      "sprdId": "BTC-USDT_BTC-USDT-SWAP",
      "ordId": "312269865356374016",
      "clOrdId": "b1",
      "tag": "",
      "px": "999",
      "sz": "3",
      "ordType": "limit",
      "side": "buy",
      "fillSz": "0",
      "fillPx": "",
      "tradeId": "",
      "accFillSz": "0",
      "pendingFillSz": "2",
      "pendingSettleSz": "1",
      "canceledSz": "1",
      "state": "live",
      "avgPx": "0",
      "cancelSource": "",
      "uTime": "1597026383085",
      "cTime": "1597026383085",
      "code": "0",
      "msg": "",
      "reqId": "",
      "amendResult": ""
    }
  ]
}
```

## PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> uid	String	User Identifier
> sprdId	String	spread ID
data	Array of objects	Subscribed data
> sprdId	String	spread ID, e.g.
> ordId	String	Order ID
> clOrdId	String	Client Order ID as assigned by the client
> tag	String	Order tag
> px	String	Order price
> sz	String	The original order quantity, in the unit of szCcy
> ordType	String	Order type <div style="display: flex; justify-content: space-between; align-items: flex-start;"> <span>market</span>: Market order  <span>limit</span>: limit order  <span>post_only</span>: Post-only order  <span>ioc</span>: Immediate-or-cancel order       </div>

Parameter	Type	Description
> side	String	Order side, buy sell
> fillSz	String	Last trade quantity, only applicable to order updates representing successful settlement
> fillPx	String	Last trade price, only applicable to order updates representing successful settlement
> tradeId	String	Last trade ID
> accFillSz	String	Accumulated fill quantity
> pendingFillSz	String	Quantity still remaining to be filled
> pendingSettleSz	String	Quantity that's pending settlement
> canceledSz	String	Quantity canceled due order cancellations or trade rejections
> avgPx	String	Average filled price. If none is filled, it will return "0".
> state	String	Order state: <input type="button" value="canceled"/> <input type="button" value="live"/> <input type="button" value="partially_filled"/> <input type="button" value="filled"/>
> cancelSource	String	Source of the order cancellation. Valid values and the corresponding meanings are: <input type="button" value="0: Order canceled by system"/> <input type="button" value="1: Order canceled by user"/> <input type="button" value="14: Order canceled: IOC order was partially canceled due to incompletely filled"/> <input type="button" value="15: Order canceled: The order price is beyond the limit"/> <input type="button" value="20: Cancel all after triggered"/> <input type="button" value="31: The post-only order will take liquidity in maker orders"/> <input type="button" value="32: Self trade prevention"/> <input type="button" value="34: Order failed to settle due to insufficient margin"/> <input type="button" value="35: Order cancellation due to insufficient margin from another order"/> <input type="button" value="44: Your order was canceled because your available balance of this crypto was insufficient for auto conversion. Auto conversion was triggered when the total collateralized liabilities for this crypto reached the platform's risk control limit."/>
> uTime	String	Update time, Unix timestamp format in milliseconds, e.g. 1597026383085
> cTime	String	Creation time, Unix timestamp format in milliseconds, e.g. 1597026383085
> code	String	Error Code, the default is 0
> msg	String	Error Message, the default is ""
> reqId	String	Client Request ID as assigned by the client for order amendment. "" will be returned if there is no order amendment.
> amendResult	String	The result of amending the order <input type="button" value="-1: failure"/> <input type="button" value="0: success"/> "" will be returned if there is no order amendment.

## TRADES CHANNEL

All updates relating to User's Trades are sent through the `spred-trades` WebSocket Notifications channel.

This is a private channel and consumable solely by the authenticated user.

Updates received through the `spred-trades` WebSocket Notification channel can include Trades being `filled` or `rejected`.

You may receive multiple notifications if an Order of yours interacts with more than one other Order.

/ws/v5/business (required login)

## Request Example : single

```

import asyncio
from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/business",
        useServerTime=False
    )
    await ws.start()
    args = [
        {
            "channel": "sprd-trades",
            "sprdId": "BTC-USDT-BTC-USDT-SWAP"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())

```

## Request Example:

```

import asyncio
from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/business",
        useServerTime=False
    )
    await ws.start()
    args = [
        {
            "channel": "sprd-trades",
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation subscribe unsubscribe
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name sprd-trades
> sprdId	String	No	Spread ID

#### RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event subscribe unsubscribe error
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> sprdId	String	No	Spread ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

Push Data Example

#### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> uid	String	User Identifier
> sprdId	String	spread ID
data	Array of objects	Subscribed data
> sprdId	String	spread ID
> tradId	String	Trade ID
> ordId	String	Order ID
> clOrdId	String	Client Order ID as assigned by the client

Parameter	Type	Description
> tag	String	Order tag
> fillPx	String	Last filled price
> fillSz	String	Last filled quantity
> side	String	Order side, buy sell
> state	String	Trade state. Valid values are filled and rejected
> execType	String	Liquidity taker or maker T: taker M: maker
> ts	String	Data generation time, Unix timestamp format in milliseconds, e.g. 1597026383085.
> legs	Array of objects	Legs of trade
>> instId	String	Instrument ID, e.g. BTC-USDT-SWAP
>> px	String	The price the leg executed
>> sz	String	Size of the leg in contracts or spot.
>> szCont	String	Filled amount of the contract Only applicable to contracts, return "" for spot
>> side	String	The direction of the leg. Valid value can be <code>buy</code> or <code>sell</code> .
>> fillPnl	String	Last filled profit and loss, applicable to orders which have a trade and aim to close position. It always is 0 in other conditions
>> fee	String	Fee. Negative number represents the user transaction fee charged by the platform. Positive number represents rebate.
>> feeCcy	String	Fee currency
>> tradId	String	Traded ID in the OKX orderbook.
> code	String	Error Code, the default is 0
> msg	String	Error Message, the default is ""

## WebSocket Public Channel

- Production Trading URL: `wss://ws.okx.com:8443/ws/v5/business`
- Demo Trading URL: `wss://wspap.okx.com:8443/ws/v5/business`

### ORDER BOOK CHANNEL

Retrieve order book data. Available channels:

- `sprd-bbo-tbt`: 1 depth level snapshot will be pushed in the initial push. Snapshot data will be pushed every 10 ms when there are changes in the 1 depth level snapshot.
- `sprd-books5`: 5 depth levels snapshot will be pushed in the initial push. Snapshot data will be pushed every 100 ms when there are changes in the 5 depth levels snapshot.
- `sprd-books-12-tbt`: 400 depth levels will be pushed in the initial full snapshot. Incremental data will be pushed every 10 ms for the changes in the order book during that period of time.
- The push sequence for order book channels within the same connection and trading symbols is fixed as: `sprd-bbo-tbt` -> `sprd-books-12-tbt` -> `sprd-books5`.

/ws/v5/business

## Request Example: sprd-books5

```

import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/business")
    await ws.start()
    args = [
        {
            "channel": "sprd-books5",
            "sprdId": "BTC-USDT_BTC-USDT-SWAP"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())

```

## Request Example: sprd-books-l2-tbt

```

import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/business")
    await ws.start()
    args = [
        {
            "channel": "sprd-books-l2-tbt",
            "sprdId": "BTC-USDT_BTC-USDT-SWAP"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels <code>sprd-bbo-tbt</code>

Parameter	Type	Required	Description
			<div style="display: flex; justify-content: space-around;"> <span>sprd-books5</span> <span>sprd-books-12-tbt</span> </div>
> channel	String	Yes	Channel name
> sprdId	String	Yes	spread ID

Successful Response Example: sprd-books5

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "sprd-books5",
    "sprdId": "BTC-USDT_BTC-USDT-SWAP"
  },
  "connId": "a4d3ae55"
}
```

Successful Response Example: sprd-books-l2-tbt

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "sprd-books-12-tbt",
    "sprdId": "BTC-USDT_BTC-USDT-SWAP"
  },
  "connId": "214fdd24"
}
```

Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\"op\": \"subscribe\", \"args\":[{ \"channel\" : \"sprd-books5\", \"sprdId\" : \"BTC-USDT_BTC-USDT-191227\"}]}",
  "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event <div style="display: flex; justify-content: space-around;"> <span>subscribe</span> <span>unsubscribe</span> <span>error</span> </div>
arg	Object	No	Subscribed channels <div style="display: flex; justify-content: space-around;"> <span>sprd-bbo-tbt</span> <span>sprd-books5</span> <span>sprd-books-12-tbt</span> </div>
> channel	String	Yes	Channel name
> sprdId	String	Yes	spread ID
msg	String	No	Error message
code	String	No	Error code

Parameter	Type	Required	Description
connId	String	Yes	WebSocket connection ID

#### Push Data Example: sprd-books5

```
{
  "arg": {
    "channel": "sprd-books5",
    "sprdId": "BTC-USDT_BTC-USDT-SWAP"
  },
  "data": [
    {
      "asks": [
        ["111.06", "55154", "2"],
        ["111.07", "53276", "2"],
        ["111.08", "72435", "2"],
        ["111.09", "70312", "2"],
        ["111.1", "67272", "2"]]
      ],
      "bids": [
        ["111.05", "57745", "2"],
        ["111.04", "57109", "2"],
        ["111.03", "69563", "2"],
        ["111.02", "71248", "2"],
        ["111.01", "65090", "2"]]
      ],
      "ts": "1670324386802",
      "seqId": 1724294007352168320
    }
  ]
}
```

#### Push Data Example: sprd-books-l2-tbt

```
{
  "arg": {
    "channel": "sprd-books-l2-tbt",
    "sprdId": "BTC-USDT_BTC-USDT-SWAP"
  },
  "action": "snapshot",
  "data": [
    {
      "asks": [
        ["1.9", "1.1", "3"],
        ["2.5", "0.9", "1"],
        ["3.2", "4.921", "1"],
        ["4.8", "0.165", "1"],
        ["5.2", "4.921", "1"]
        .....
      ],
      "bids": [
        ["1.8", "0.165", "1"],
        ["0.6", "0.2", "2"],
        ["0", "23.49", "1"],
        ["-0.1", "1", "1"],
        ["-0.6", "1", "1"],
        ["-3.9", "4.921", "1"]
        .....
      ],
      "ts": "1724391380926",
      "checksum": -1285595583,
      "prevSeqId": -1,
      "seqId": 1724294007352168320
    }
  ]
}
```

#### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel

Parameter	Type	Description
> channel	String	Channel name
> sprdId	String	spread ID
action	String	Push data action, incremental data or full snapshot. snapshot: full update: incremental
data	Array of objects	Subscribed data
> asks	Array of strings	Order book on sell side
> bids	Array of strings	Order book on buy side
> ts	String	Order book generation time, Unix timestamp format in milliseconds, e.g. 1597026383085
> checksum	Integer	Checksum, implementation details below. Only applicable to <a href="#">sprd-books-12-tbt</a> .
> prevSeqId	Integer	Sequence ID of the last sent message. Only applicable to <a href="#">sprd-books-12-tbt</a> .
> seqId	Integer	Sequence ID of the current message, implementation details below.

An example of the array of asks and bids values: ["411.8", "10", "4"]

- "411.8" is the depth price
- "10" is the quantity at the price (Unit: szCcy)
- "4" is the number of orders at the price.

## SEQUENCE ID

[seqId](#) is the sequence ID of the market data published. The set of sequence ID received by users is the same if users are connecting to the same channel through multiple websocket connections. Each [sprdId](#) has an unique set of sequence ID. Users can use [prevSeqId](#) and [seqId](#) to build the message sequencing for incremental order book updates. Generally the value of seqId is larger than prevSeqId. The [prevSeqId](#) in the new message matches with [seqId](#) of the previous message. The smallest possible sequence ID value is 0, except in snapshot messages where the prevSeqId is always -1.

Exceptions:

1. If there are no updates to the depth for an extended period, OKX will send a message with `'asks': [], 'bids': []` to inform users that the connection is still active. [seqId](#) is the same as the last sent message and [prevSeqId](#) equals to [seqId](#). 2. The sequence number may be reset due to maintenance, and in this case, users will receive an incremental message with [seqId](#) smaller than [prevSeqId](#). However, subsequent messages will follow the regular sequencing rule.

## EXAMPLE

1. Snapshot message: prevSeqId = -1, seqId = 10
2. Incremental message 1 (normal update): prevSeqId = 10, seqId = 15
3. Incremental message 2 (no update): prevSeqId = 15, seqId = 15
4. Incremental message 3 (sequence reset): prevSeqId = 15, seqId = 3
5. Incremental message 4 (normal update): prevSeqId = 3, seqId = 5

## CHECKSUM

This mechanism can assist users in checking the accuracy of depth data.

## MERGING INCREMENTAL DATA INTO FULL DATA

After subscribing to the incremental load push (such as [books](#) 400 levels) of Order Book Channel, users first receive the initial full load of market depth. After the incremental load is subsequently received, update the local full load.

1. If there is the same price, compare the size. If the size is 0, delete this depth data. If the size changes, replace the original data.
2. If there is no same price, sort by price (bid in descending order, ask in ascending order), and insert the depth information into the full load.

## CALCULATE CHECKSUM

Use the first 25 bids and asks in the full load to form a string (where a colon connects the price and size in an ask or a bid), and then calculate the CRC32 value (32-bit signed integer).

## PUBLIC TRADES CHANNEL

Retrieve the recent trades data from `sprd-public-trades`. Data will be pushed whenever there is a trade. Every update contains only one trade.

### URL PATH

/ws/v5/business

#### Request Example

```
import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/business")
    await ws.start()
    args = [
        {
            "channel": "sprd-public-trades",
            "sprdId": "BTC-USDT_BTC-USDT-SWAP"
        }
    ]

    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>sprd-public-trades</code>
> sprdId	String	Yes	spread ID

#### Successful Response Example

#### Failure Response Example

### RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event <code>subscribe</code>

Parameter	Type	Required	Description
			<a href="#">unsubscribe</a> <a href="#">error</a>
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> sprdId	String	Yes	spread ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

### Push Data Example

#### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> sprdId	String	spread ID
data	Array of objects	Subscribed data
> sprdId	String	spread ID, e.g.
> tradId	String	Trade ID
> px	String	Trade price
sz	String	Trade quantity For spot trading, the unit is base currency For <a href="#">FUTURES</a> / <a href="#">SWAP</a> / <a href="#">OPTION</a> , the unit is contract.
> side	String	Trade direction, buy, sell
> ts	String	Filled time, Unix timestamp format in milliseconds, e.g. 1597026383085

#### TICKERS CHANNEL

Retrieve the last traded price, bid price, ask price. The fastest rate is 1 update/100ms. There will be no update if the event is not triggered. The events which can trigger update: trade, the change on best ask/bid price

#### URL PATH

/ws/v5/business

### Request Example

```
import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/business")
    await ws.start()
    args = [
        {
            "channel": "sprd-tickers",
        }
    ]
    ws.subscribe(args)
    await ws.close()
```

```

    "sprdId": "BTC-USDT_BTC-USDT-SWAP"
}

]

await ws.subscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

await ws.unsubscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation subscribe unsubscribe
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name sprd-tickers
> sprdId	String	Yes	spread ID

Successful Response Example

Failure Response Example

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event subscribe unsubscribe error
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> sprdId	String	Yes	spread ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

Push Data Example

## PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel

Parameter	Type	Description
> channel	String	Channel name
> sprdId	String	spread ID
data	Array of objects	Subscribed data
> sprdId	String	spread ID
> last	String	Last traded price
> lastSz	String	Last traded size
> askPx	String	Best ask price
> askSz	String	Best ask size
> bidPx	String	Best bid price
> bidSz	String	Best bid size
> open24h	String	Open price in the past 24 hours
> high24h	String	Highest price in the past 24 hours
> low24h	String	Lowest price in the past 24 hours
> vol24h	String	24h trading volume, with a unit of base currency or USD
> ts	String	Ticker data generation time, Unix timestamp format in milliseconds, e.g. 1597026383085

### vol24h

For Spot vs USDT-margined contracts spread and USDT-margined contracts spread, the volume is with the unit of base currency; for Crypto-margined contracts spread, the volume is with the unit of USD.

## CANDLESTICKS CHANNEL

Retrieve the candlesticks data of an instrument. The push frequency is the fastest interval 1 second push the data.

### URL PATH

/ws/v5/business

### Request Example

```

import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/business")
    await ws.start()
    args = [
        {
            "channel": "sprd-candleID",
            "sprdId": "BTC-USDT_BTC-USDT-SWAP"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

```

```
asyncio.run(main())
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation, subscribe unsubscribe
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <div style="display: flex; justify-content: space-around; gap: 10px;"> <span>sprd-candle3M</span> <span>sprd-candle1M</span>   <span>sprd-candle1W</span>   <span>sprd-candle1D</span> <span>sprd-candle2D</span> <span>sprd-candle3D</span> <span>sprd-candle5D</span>   <span>sprd-candle12H</span> <span>sprd-candle6H</span> <span>sprd-candle4H</span> <span>sprd-candle2H</span> <span>sprd-candle1H</span>   <span>sprd-candle30m</span> <span>sprd-candle15m</span> <span>sprd-candle5m</span> <span>sprd-candle3m</span> <span>sprd-candle1m</span>   <span>sprd-candle3Mutc</span> <span>sprd-candle1Mutc</span> <span>sprd-candle1Wutc</span> <span>sprd-candle1Dutc</span> <span>sprd-candle2Dutc</span> <span>sprd-candle3Dutc</span>   <span>c</span> <span>sprd-candle5Dutc</span> <span>sprd-candle12Hutc</span> <span>sprd-candle6Hutc</span> </div>
> sprdId	String	Yes	Spread ID

### Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "sprd-candle1D",
    "sprdId": "BTC-USDT_BTC-USDT-SWAP"
  },
  "connId": "a4d3ae55"
}
```

### Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \"subscribe\", \\"args\\\": [{ \\"channel\\\" : \\"sprd-candle1D\\\", \\"instId\\\" : \\"BTC-USD-191227\\\" }]}",
  "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event, subscribe unsubscribe error
arg	Object	No	Subscribed channel
channel	String	yes	channel name
sprdId	String	Yes	Spread ID

Parameter	Type	Required	Description
code	String	No	Error code
msg	String	No	Error message

### Push Data Example

```
{
  "arg": {
    "channel": "sprd-candle1D",
    "sprdId": "BTC-USDT_BTC-USD-SWAP"
  },
  "data": [
    [
      "1597026383085",
      "8533.02",
      "8553.74",
      "8527.17",
      "8548.26",
      "45247",
      "0"
    ]
  ]
}
```

### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> sprdId	String	Spread ID
data	Array of Arrays	Subscribed data
> ts	String	Opening time of the candlestick, Unix timestamp format in milliseconds, e.g. 1597026383085
> o	String	Open price
> h	String	highest price
> l	String	Lowest price
> c	String	Close price
> vol	String	Trading volume, in szCcy
> confirm	String	The state of candlesticks.0 represents that it is uncompleted, 1 represents that it is completed.

The data returned will be arranged in an array like this: [ts,o,h,l,c,vol,confirm]

# Public Data

The API endpoints of [Public Data](#) do not require authentication.

# REST API

## GET INSTRUMENTS

Retrieve a list of instruments with open contracts for OKX. Retrieve available instruments info of current account, please refer to Get instruments.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP + INSTRUMENT TYPE**

### HTTP REQUEST

```
GET /api/v5/public/instruments
```

#### Request Example

```
import okx.PublicData as PublicData

flag = "0" # Production trading: 0, Demo trading: 1

publicDataAPI = PublicData.PublicAPI(flag=flag)

# Retrieve a list of instruments with open contracts
result = publicDataAPI.get_instruments(
    instType="SPOT"
)
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	Yes	Instrument type <code>SPOT</code> : Spot <code>MARGIN</code> : Margin <code>SWAP</code> : Perpetual Futures <code>FUTURES</code> : Expiry Futures <code>OPTION</code> : Option
instFamily	String	Conditional	Instrument family Only applicable to <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code> . If <code>instType</code> is <code>OPTION</code> , <code>instFamily</code> is required.
instId	String	No	Instrument ID

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "alias": "",
      "auctionEndTime": "",
      "baseCcy": "BTC",
      "category": "1",
      "ctMult": "",
      "ctType": "",
      "ctVal": "",
      "ctValCcy": "",
      "contTdSwTime": "1704876947000",
      "expTime": "",
      "futureSettlement": false,
      "instFamily": "",
      "instId": "BTC-USDT",
      "instType": "SPOT",
      "lever": "10",
      "listTime": "1606468572000",
      "lotSz": "0.00000001",
      "maxIcebergSz": "999999999.0000000000000000",
      "maxLmtAmt": "1000000",
      "minSz": "0.00000001",
      "orderSz": "0.00000001",
      "symbol": "BTC-USDT"
    }
  ]
}
```

```

    "maxLmtSz": "999999999",
    "maxMktAmt": "1000000",
    "maxMktSz": "",
    "maxStopSz": "",
    "maxTriggerSz": "999999999.0000000000000000",
    "maxTwapSz": "999999999.0000000000000000",
    "minSz": "0.0001",
    "optType": "",
    "openType": "call_auction",
    "preMktSwTime": "",
    "quoteCcy": "USDT",
    "tradeQuoteCcyList": [
        "USDT"
    ],
    "settleCcy": "",
    "state": "live",
    "ruleType": "normal",
    "stk": "",
    "tickSz": "0.1",
    "uly": "",
    "instIdCode": 100000000
}
]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type
instId	String	Instrument ID, e.g. <code>BTC-USD-SWAP</code>
uly	String	Underlying, e.g. <code>BTC-USD</code> Only applicable to <code>MARGIN/FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>
instFamily	String	Instrument family, e.g. <code>BTC-USD</code> Only applicable to <code>MARGIN/FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>
category	String	Currency category. Note: this parameter is already deprecated
baseCcy	String	Base currency, e.g. <code>BTC</code> in <code>BTC-USDT</code> Only applicable to <code>SPOT</code> / <code>MARGIN</code>
quoteCcy	String	Quote currency, e.g. <code>USDT</code> in <code>BTC-USDT</code> Only applicable to <code>SPOT</code> / <code>MARGIN</code>
settleCcy	String	Settlement and margin currency, e.g. <code>BTC</code> Only applicable to <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>
ctVal	String	Contract value Only applicable to <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>
ctMult	String	Contract multiplier Only applicable to <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>
ctValCcy	String	Contract value currency Only applicable to <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>
optType	String	Option type, <code>C</code> : Call <code>P</code> : put Only applicable to <code>OPTION</code>
stk	String	Strike price Only applicable to <code>OPTION</code>
listTime	String	Listing time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

Parameter	Type	Description
auctionEndTime	String	<p>The end time of call auction, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code></p> <p>Only applicable to <code>SPOT</code> that are listed through call auctions, return "" in other cases (deprecated, use <code>contTdSwTime</code>)</p>
contTdSwTime	String	<p>Continuous trading switch time. The switch time from call auction, prequote to continuous trading, Unix timestamp format in milliseconds. e.g. <code>1597026383085</code>.</p> <p>Only applicable to <code>SPOT</code>/<code>MARGIN</code> that are listed through call auction or prequote, return "" in other cases.</p>
preMktSwTime	String	<p>The time premarket swap switched to normal swap, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>.</p> <p>Only applicable premarket <code>SWAP</code></p>
openType	String	<p>Open type</p> <ul style="list-style-type: none"> <li><code>fix_price</code>: fix price opening</li> <li><code>pre_quote</code>: pre-quote</li> <li><code>call_auction</code>: call auction</li> </ul> <p>Only applicable to <code>SPOT</code>/<code>MARGIN</code>, return "" for all other business lines</p>
expTime	String	<p>Expiry time</p> <p>Applicable to <code>SPOT</code>/<code>MARGIN</code>/<code>FUTURES</code>/<code>SWAP</code>/<code>OPTION</code>. For <code>FUTURES</code>/<code>OPTION</code>, it is natural delivery/exercise time. It is the instrument offline time when there is <code>SPOT/MARGIN/FUTURES/SWAP</code> manual offline. Update once change.</p>
lever	String	<p>Max Leverage,</p> <p>Not applicable to <code>SPOT</code>, <code>OPTION</code></p>
tickSz	String	<p>Tick size, e.g. <code>0.0001</code></p> <p>For Option, it is minimum tickSz among tick band, please use "Get option tick bands" if you want get option tickBands.</p>
lotSz	String	<p>Lot size</p> <p>If it is a derivatives contract, the value is the number of contracts.</p> <p>If it is <code>SPOT</code>/<code>MARGIN</code>, the value is the quantity in <code>base currency</code>.</p>
minSz	String	<p>Minimum order size</p> <p>If it is a derivatives contract, the value is the number of contracts.</p> <p>If it is <code>SPOT</code>/<code>MARGIN</code>, the value is the quantity in <code>base currency</code>.</p>
ctType	String	<p>Contract type</p> <ul style="list-style-type: none"> <li><code>linear</code>: linear contract</li> <li><code>inverse</code>: inverse contract</li> </ul> <p>Only applicable to <code>FUTURES</code>/<code>SWAP</code></p>
alias	String	<p>Alias</p> <ul style="list-style-type: none"> <li><code>this_week</code></li> <li><code>next_week</code></li> <li><code>this_month</code></li> <li><code>next_month</code></li> <li><code>quarter</code></li> <li><code>next_quarter</code></li> <li><code>third_quarter</code></li> </ul> <p>Only applicable to <code>FUTURES</code></p> <p><b>Not recommended for use, users are encouraged to rely on the expTime field to determine the delivery time of the contract</b></p>
state	String	<p>Instrument status</p> <ul style="list-style-type: none"> <li><code>live</code></li> <li><code>suspend</code></li> <li><code>preopen</code>, e.g. There will be <code>preopen</code> before the Futures and Options new contracts state is live.</li> <li><code>test</code>: Test pairs, can't be traded</li> </ul>
ruleType	String	<p>Trading rule types</p> <ul style="list-style-type: none"> <li><code>normal</code>: normal trading</li> <li><code>pre_market</code>: pre-market trading</li> </ul>

Parameter	Type	Description
maxLmtSz	String	The maximum order quantity of a single limit order. If it is a derivatives contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in <code>base currency</code> .
maxMktSz	String	The maximum order quantity of a single market order. If it is a derivatives contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in <code>USDT</code> .
maxLmtAmt	String	Max USD amount for a single limit order
maxMktAmt	String	Max USD amount for a single market order Only applicable to <code>SPOT</code> / <code>MARGIN</code>
maxTwapSz	String	The maximum order quantity of a single TWAP order. If it is a derivatives contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in <code>base currency</code> . The minimum order quantity of a single TWAP order is <code>minSz*2</code>
maxIcebergSz	String	The maximum order quantity of a single iceBerg order. If it is a derivatives contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in <code>base currency</code> .
maxTriggerSz	String	The maximum order quantity of a single trigger order. If it is a derivatives contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in <code>base currency</code> .
maxStopSz	String	The maximum order quantity of a single stop market order. If it is a derivatives contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in <code>USDT</code> .
futureSettlement	Boolean	Whether daily settlement for expiry feature is enabled Applicable to <code>FUTURES</code> <code>cross</code>
tradeQuoteCcyList	Array of strings	List of quote currencies available for trading, e.g. <code>["USD", "USDC"]</code> .
instIdCode	Integer	Instrument ID code. For simple binary encoding, you must use <code>instIdCode</code> instead of <code>instId</code> . For the same <code>instId</code> , its value may be different between production and demo trading.

When a new contract is going to be listed, the instrument data of the new contract will be available with status `preopen`. When a product is going to be delisted (e.g. when a `FUTURES` contract is settled or `OPTION` contract is exercised), the instrument will not be available

#### listTime and contTdSwTime

For spot symbols listed through a call auction or pre-open, `listTime` represents the start time of the auction or pre-open, and `contTdSwTime` indicates the end of the auction or pre-open and the start of continuous trading. For other scenarios, `listTime` will mark the beginning of continuous trading, and `contTdSwTime` will return an empty value `""`.

#### state

The state will always change from `'preopen'` to `'live'` when the `listTime` is reached.

When a product is going to be delisted (e.g. when a `FUTURES` contract is settled or `OPTION` contract is exercised), the instrument will not be available.

Instruments REST endpoints and WebSocket channel will update `'expTime'` once the delisting announcement is published.

Instruments REST endpoint and WebSocket channel will update `'listTime'` once the listing announcement is published:

1. For `'SPOT/MARGIN/SWAP'`, this event is only applicable to `'instType'`, `'instId'`, `'listTime'`, `'state'`.

2. For `FUTURES`, this event is only applicable to `instType`, `instFamily`, `listTime`, `state`.

3. Other fields will be "" temporarily, but they will be updated at least 5 minutes in advance of the `listTime`, then the WebSocket subscription using related `instId`/`instFamily` can be available.

#### GET ESTIMATED DELIVERY/EXERCISE PRICE

Retrieve the estimated delivery price which will only have a return value one hour before the delivery/exercise.

#### RATE LIMIT: 10 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: IP + INSTRUMENT ID

#### HTTP REQUEST

```
GET /api/v5/public/estimated-price
```

#### Request Example

```
import okx.PublicData as PublicData

flag = "0" # Production trading: 0, Demo trading: 1

publicDataAPI = PublicData.PublicAPI(flag=flag)

# Retrieve estimated delivery/exercise price
result = publicDataAPI.get_estimated_price(
    instId = "BTC-USD-200214",
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USD-200214</code> only applicable to <code>FUTURES</code> / <code>OPTION</code>

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "instType": "FUTURES",
      "instId": "BTC-USDT-201227",
      "settlePx": "200",
      "ts": "1597026383085"
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type <code>FUTURES</code> <code>OPTION</code>
instId	String	Instrument ID, e.g. <code>BTC-USD-200214</code>
settlePx	String	Estimated delivery/exercise price
ts	String	Data return time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

#### GET DELIVERY/EXERCISE HISTORY

Retrieve delivery records of Futures and exercise records of Options in the last 3 months.

**RATE LIMIT: 40 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP + (INSTRUMENT TYPE + INSTFAMILY)**

**HTTP REQUEST**

GET /api/v5/public/delivery-exercise-history

Request Example

```
import okx.PublicData as PublicData

flag = "0" # Production trading: 0, Demo trading: 1

publicDataAPI = PublicData.PublicAPI(flag=flag)

# Retrieve delivery records of Futures and exercise records of Options in the last 3 months
result = publicDataAPI.get_delivery_exercise_history(
    instType="FUTURES",
    instFamily="BTC-USD"
)
print(result)
```

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
instType	String	Yes	Instrument type <a href="#">FUTURES</a> <a href="#">OPTION</a>
instFamily	String	Yes	Instrument family, only applicable to <a href="#">FUTURES</a> / <a href="#">OPTION</a>
after	String	No	Pagination of data to return records earlier than the requested <a href="#">ts</a>
before	String	No	Pagination of data to return records newer than the requested <a href="#">ts</a>
limit	String	No	Number of results per request. The maximum is <a href="#">100</a> ; The default is <a href="#">100</a>

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ts": "1597026383085",
      "details": [
        {
          "type": "delivery",
          "insId": "BTC-USD-190927",
          "px": "0.016"
        }
      ]
    },
    {
      "ts": "1597026383085",
      "details": [
        {
          "insId": "BTC-USD-200529-6000-C",
          "type": "exercised",
          "px": "0.016"
        },
        {
          "insId": "BTC-USD-200529-8000-C",
          "type": "exercised",
          "px": "0.016"
        }
      ]
    }
  ]
}
```

```
]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Delivery/exercise time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
details	Array of objects	Delivery/exercise details
> instId	String	Delivery/exercise contract ID
> px	String	Delivery/exercise price
> type	String	Type <code>delivery</code> <code>exercised</code> <code>expired_otm</code> : Out of the money

## GET ESTIMATED FUTURE SETTLEMENT PRICE

Retrieve the estimated settlement price which will only have a return value one hour before the settlement.

**RATE LIMIT: 10 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP + INSTRUMENT ID**

## HTTP REQUEST

```
GET /api/v5/public/estimated-settlement-info
```

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>XRP-USDT-250307</code> only applicable to <code>FUTURES</code>

Response Example

```
{
  "code": "0",
  "data": [
    {
      "estSettlePx": "2.5666068562369959",
      "instId": "XRP-USDT-250307",
      "nextSettleTime": "1741248000000",
      "ts": "1741246429748"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID, e.g. <code>XRP-USDT-250307</code>
nextSettleTime	String	Next settlement time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
estSettlePx	String	Estimated settlement price
ts	String	Data return time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## GET FUTURES SETTLEMENT HISTORY

Retrieve settlement records of futures in the last 3 months.

**RATE LIMIT: 40 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP + (INSTRUMENT FAMILY)**

### HTTP REQUEST

```
GET /api/v5/public/settlement-history
```

Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instFamily	String	Yes	Instrument family
after	String	No	Pagination of data to return records earlier than (not include) the requested <code>ts</code>
before	String	No	Pagination of data to return records newer than (not include) the requested <code>ts</code>
limit	String	No	Number of results per request. The maximum is <code>100</code> . The default is <code>100</code>

Response Example

```
{
  "code": "0",
  "data": [
    {
      "details": [
        {
          "instId": "XRP-USDT-250307",
          "settlePx": "2.5192078615298715"
        }
      ],
      "ts": "1741161600000"
    },
    {
      "details": [
        {
          "instId": "XRP-USDT-250307",
          "settlePx": "2.5551316341327384"
        }
      ],
      "ts": "1741075200000"
    }
  ],
  "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Settlement time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
details	Array of objects	Settlement info
> instId	String	Instrument ID
> settlePx	String	Settlement price

## GET FUNDING RATE

Retrieve funding rate.

**RATE LIMIT: 10 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP + INSTRUMENT ID**

## HTTP REQUEST

GET /api/v5/public/funding-rate

### Request Example

```
import okx.PublicData as PublicData

flag = "0" # Production trading: 0, Demo trading: 1

publicDataAPI = PublicData.PublicAPI(flag=flag)

# Retrieve funding rate
result = publicDataAPI.get_funding_rate(
    instId="BTC-USD-SWAP",
)
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USD-SWAP</code> or <code>ANY</code> to return the funding rate info of all swap symbols only applicable to <code>SWAP</code>

### Response Example

```
{
  "code": "0",
  "data": [
    {
      "formulaType": "noRate",
      "fundingRate": "0.0000182221218054",
      "fundingTime": "1743609600000",
      "impactValue": "",
      "instId": "BTC-USDT-SWAP",
      "instType": "SWAP",
      "interestRate": "",
      "maxFundingRate": "0.00375",
      "method": "current_period",
      "minFundingRate": "-0.00375",
      "nextFundingRate": "",
      "nextFundingTime": "1743638400000",
      "premium": "0.0000910113652644",
      "settFundingRate": "0.0000145824401745",
      "settState": "settled",
      "ts": "1743588686291"
    }
  ],
  "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type <code>SWAP</code>
instId	String	Instrument ID, e.g. <code>BTC-USD-SWAP</code> or <code>ANY</code>
method	String	Funding rate mechanism <code>current_period</code> <code>next_period</code> (no longer supported)
formulaType	String	Formula type <code>noRate</code> : old funding rate formula <code>withRate</code> : new funding rate formula
fundingRate	String	Current funding rate

Parameter	Type	Description
nextFundingRate	String	Forecasted funding rate for the next period The nextFundingRate will be "" if the method is <code>current_period</code> (no longer supported)
fundingTime	String	Settlement time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
nextFundingTime	String	Forecasted funding time for the next period , Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
minFundingRate	String	The lower limit of the funding rate
maxFundingRate	String	The upper limit of the funding rate
interestRate	String	Interest rate
impactValue	String	Depth weighted amount (in the unit of quote currency)
settState	String	Settlement state of funding rate <code>processing</code> <code>settled</code>
settFundingRate	String	If settState = <code>processing</code> , it is the funding rate that is being used for current settlement cycle. If settState = <code>settled</code> , it is the funding rate that is being used for previous settlement cycle
premium	String	Premium index formula: $[\text{Max}(0, \text{Impact bid price} - \text{Index price}) - \text{Max}(0, \text{Index price} - \text{Impact ask price})] / \text{Index price}$
ts	String	Data return time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

For some altcoins perpetual swaps with significant fluctuations in funding rates, OKX will closely monitor market changes. When necessary, the funding rate collection frequency, currently set at 8 hours, may be adjusted to higher frequencies such as 6 hours, 4 hours, 2 hours, or 1 hour. Thus, users should focus on the difference between `fundingTime` and `nextFundingTime` fields to determine the funding fee interval of a contract.

## GET FUNDING RATE HISTORY

Retrieve funding rate history. This endpoint can return data up to three months.

### RATE LIMIT: 10 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: IP + INSTRUMENT ID

### HTTP REQUEST

`GET /api/v5/public/funding-rate-history`

#### Request Example

```
import okx.PublicData as PublicData

flag = "0" # Production trading: 0, Demo trading: 1

publicDataAPI = PublicData.PublicAPI(flag=flag)

# Retrieve funding rate history
result = publicDataAPI.funding_rate_history(
    instId="BTC-USD-SWAP",
)
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USD-SWAP</code> only applicable to <code>SWAP</code>
before	String	No	Pagination of data to return records newer than the requested <code>fundingTime</code>
after	String	No	Pagination of data to return records earlier than the requested <code>fundingTime</code>
limit	String	No	Number of results per request. The maximum is <code>400</code> ; The default is <code>400</code>

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "formulaType": "noRate",
      "fundingRate": "0.0000746604960499",
      "fundingTime": "1703059200000",
      "instId": "BTC-USD-SWAP",
      "instType": "SWAP",
      "method": "next_period",
      "realizedRate": "0.0000746572360545"
    },
    {
      "formulaType": "noRate",
      "fundingRate": "0.000227985782722",
      "fundingTime": "1703030400000",
      "instId": "BTC-USD-SWAP",
      "instType": "SWAP",
      "method": "next_period",
      "realizedRate": "0.0002279755647389"
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type <code>SWAP</code>
instId	String	Instrument ID, e.g. <code>BTC-USD-SWAP</code>
formulaType	String	Formula type <code>noRate</code> : old funding rate formula <code>withRate</code> : new funding rate formula
fundingRate	String	Predicted funding rate
realizedRate	String	Actual funding rate
fundingTime	String	Settlement time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
method	String	Funding rate mechanism <code>current_period</code> <code>next_period</code>

For some altcoins perpetual swaps with significant fluctuations in funding rates, OKX will closely monitor market changes. When necessary, the funding rate collection frequency, currently set at 8 hours, may be adjusted to higher frequencies such as 6 hours, 4 hours, 2 hours, or 1 hour.

Thus, users should focus on the difference between `fundingTime` and `nextFundingTime` fields to determine the funding fee interval of a contract.

## GET OPEN INTEREST

Retrieve the total open interest for contracts on OKX.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP + INSTRUMENT ID**

### HTTP REQUEST

```
GET /api/v5/public/open-interest
```

Request Example

```
import okx.PublicData as PublicData

flag = "0" # Production trading: 0, Demo trading: 1

publicDataAPI = PublicData.PublicAPI(flag=flag)

# Retrieve the total open interest for contracts on OKX
result = publicDataAPI.get_open_interest(
    instType="SWAP",
)
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	Yes	Instrument type <b>SWAP</b> <b>FUTURES</b> <b>OPTION</b>
instFamily	String	Conditional	Instrument family Applicable to <b>FUTURES</b> / <b>SWAP</b> / <b>OPTION</b> If instType is <b>OPTION</b> , instFamily is required.
instId	String	No	Instrument ID, e.g. <b>BTC-USDT-SWAP</b> Applicable to <b>FUTURES</b> / <b>SWAP</b> / <b>OPTION</b>

Response Example

```
{
    "code": "0",
    "msg": "",
    "data": [
        {
            "instType": "SWAP",
            "instId": "BTC-USDT-SWAP",
            "oi": "5000",
            "oiCcy": "555.55",
            "oiUsd": "50000",
            "ts": "1597026383085"
        }
    ]
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type

Parameter	Type	Description
instId	String	Instrument ID
oi	String	Open interest in number of contracts
oiCcy	String	Open interest in number of coin
oiUsd	String	Open interest in number of USD
ts	String	Data return time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

#### GET LIMIT PRICE

Retrieve the highest buy limit and lowest sell limit of the instrument.

#### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: IP

#### HTTP REQUEST

`GET /api/v5/public/price-limit`

#### Request Example

```
import okx.PublicData as PublicData

flag = "0" # Production trading: 0, Demo trading: 1

publicDataAPI = PublicData.PublicAPI(flag=flag)

# Retrieve the highest buy limit and lowest sell limit of the instrument
result = publicDataAPI.get_price_limit(
    instId="BTC-USD-SWAP",
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT-SWAP</code>

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "instType": "SWAP",
      "instId": "BTC-USDT-SWAP",
      "buyLmt": "17057.9",
      "sellLmt": "16388.9",
      "ts": "1597026383085",
      "enabled": true
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type
instId	String	Instrument ID, e.g. <code>BTC-USDT-SWAP</code>

Parameter	Type	Description
buyLmt	String	Highest buy limit Return "" when enabled is false
sellLmt	String	Lowest sell limit Return "" when enabled is false
ts	String	Data return time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
enabled	Boolean	Whether price limit is effective <code>true</code> : the price limit is effective <code>false</code> : the price limit is not effective

## GET OPTION MARKET DATA

Retrieve option market data.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP + INSTFAMILY**

### HTTP REQUEST

`GET /api/v5/public/opt-summary`

#### Request Example

```
import okx.PublicData as PublicData

flag = "0" # Production trading: 0, Demo trading: 1

publicDataAPI = PublicData.PublicAPI(flag=flag)

# Retrieve option market data
result = publicDataAPI.get_opt_summary(
    instFamily="BTC-USD",
)
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instFamily	String	Yes	Instrument family, only applicable to <code>OPTION</code>
expTime	String	No	Contract expiry date, the format is "YYMMDD", e.g. "200527"

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "askVol": "3.7207056835937498",
      "bidVol": "0",
      "delta": "0.8310206676289528",
      "deltaBS": "0.9857332101544538",
      "fwdPx": "39016.8143629068452065",
      "gamma": "-1.1965483553276135",
      "gammaBS": "0.000011933182397798109",
      "instId": "BTC-USD-220309-33000-C",
      "instType": "OPTION",
      "lever": "0",
      "markVol": "1.5551965233045728",
      "realVol": "0",
      "volLv": "0",
      "theta": "-0.0014131955002093717",
      "thetaBS": "-66.03526900575946",
    }
  ]
}
```

```

    "ts": "1646733631242",
    "uly": "BTC-USD",
    "vega": "0.00018173851073258973",
    "vegaBS": "0.7089307622132419"
},
{
    "askVol": "1.7968814062499998",
    "bidVol": "0",
    "delta": "-0.014668822072611904",
    "deltaBS": "-0.01426678984554619",
    "fwdPx": "39016.8143629068452065",
    "gamma": "0.49483062407551576",
    "gammaBS": "0.00011933182397798109",
    "instId": "BTC-USD-220309-33000-P",
    "instType": "OPTION",
    "lever": "0",
    "markVol": "1.5551965233045728",
    "realVol": "0",
    "volLv": "0",
    "theta": "-0.0014131955002093717",
    "thetaBS": "-54.93377294845015",
    "ts": "1646733631242",
    "uly": "BTC-USD",
    "vega": "0.00018173851073258973",
    "vegaBS": "0.7089307622132419"
}
]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type OPTION
instId	String	Instrument ID, e.g. BTC-USD-200103-5500-C
uly	String	Underlying
delta	String	Sensitivity of option price to uly price
gamma	String	The delta is sensitivity to uly price
vega	String	Sensitivity of option price to implied volatility
theta	String	Sensitivity of option price to remaining maturity
deltaBS	String	Sensitivity of option price to uly price in BS mode
gammaBS	String	The delta is sensitivity to uly price in BS mode
vegaBS	String	Sensitivity of option price to implied volatility in BS mode
thetaBS	String	Sensitivity of option price to remaining maturity in BS mode
lever	String	Leverage
markVol	String	Mark volatility
bidVol	String	Bid volatility
askVol	String	Ask volatility
realVol	String	Realized volatility (not currently used)
volLv	String	Implied volatility of at-the-money options
fwdPx	String	Forward price

Parameter	Type	Description
ts	String	Data update time, Unix timestamp format in milliseconds, e.g. 1597026383085

#### GET DISCOUNT RATE AND INTEREST-FREE QUOTA

Retrieve discount rate level and interest-free quota.

**RATE LIMIT: 2 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

#### HTTP REQUEST

GET /api/v5/public/discount-rate-interest-free-quota

#### Request Example

```
import okx.PublicData as PublicData

flag = "0" # Production trading: 0, Demo trading: 1

publicDataAPI = PublicData.PublicAPI(flag=flag)

# Retrieve discount rate level and interest-free quota
result = publicDataAPI.discount_interest_free_quota()
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	No	Currency
discountLv	String	No	Discount level (Deprecated)

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "amt": "0",
      "ccy": "BTC",
      "collateralRestrict": false,
      "details": [
        {
          "discountRate": "0.98",
          "liqPenaltyRate": "0.02",
          "maxAmt": "20",
          "minAmt": "0",
          "tier": "1",
          "disCcyEq": "1000"
        },
        {
          "discountRate": "0.9775",
          "liqPenaltyRate": "0.0225",
          "maxAmt": "25",
          "minAmt": "20",
          "tier": "2",
          "disCcyEq": "2000"
        }
      ],
      "discountLv": "1",
      "minDiscountRate": "0"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency
colRes	String	<p>Platform level collateral restriction status</p> <p>0: The restriction is not enabled.</p> <p>1: The restriction is not enabled. But the crypto is close to the platform's collateral limit.</p> <p>2: The restriction is enabled. This crypto can't be used as margin for your new orders. This may result in failed orders. But it will still be included in the account's adjusted equity and doesn't impact margin ratio.</p> <p>Refer to Introduction to the platform collateralized borrowing limit for more details.</p>
collateralRestrict	Boolean	<p><code>true</code></p> <p><code>false</code> (deprecated, use colRes instead)</p>
amt	String	Interest-free quota
discountLv	String	Discount rate level. (Deprecated)
minDiscountRate	String	Minimum discount rate when it exceeds the maximum amount of the last tier.
details	Array of objects	New discount details.
> discountRate	String	Discount rate
> maxAmt	String	<p>Tier - upper bound.</p> <p>The unit is the currency like BTC. "" means positive infinity</p>
> minAmt	String	<p>Tier - lower bound.</p> <p>The unit is the currency like BTC. The minimum is 0</p>
> tier	String	Tiers
> liqPenaltyRate	String	Liquidation penalty rate
> disCcyEq	String	Discount equity in currency for quick calculation if your equity is the <code>maxAmt</code>

## GET SYSTEM TIME

Retrieve API server time.

## RATE LIMIT: 10 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: IP

### HTTP REQUEST

`GET /api/v5/public/time`

#### Request Example

```
import okx.PublicData as PublicData

flag = "0" # Production trading: 0, Demo trading: 1

publicDataAPI = PublicData.PublicAPI(flag=flag)

# Retrieve API server time
result = publicDataAPI.get_system_time()
print(result)
```

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
```

```
{
  "ts": "1597026383085"
}
]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	System time, Unix timestamp format in milliseconds, e.g. 1597026383085

## GET MARK PRICE

Retrieve mark price.

We set the mark price based on the SPOT index and at a reasonable basis to prevent individual users from manipulating the market and causing the contract price to fluctuate.

**RATE LIMIT: 10 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP + INSTRUMENT ID**

## HTTP REQUEST

```
GET /api/v5/public/mark-price
```

### Request Example

```
import okx.PublicData as PublicData

flag = "0" # Production trading: 0, Demo trading: 1

publicDataAPI = PublicData.PublicAPI(flag=flag)

# Retrieve mark price
result = publicDataAPI.get_mark_price(
    instType="SWAP",
)
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	Yes	Instrument type <input type="button" value="MARGIN"/> <input type="button" value="SWAP"/> <input type="button" value="FUTURES"/> <input type="button" value="OPTION"/>
instFamily	String	No	Instrument family Applicable to <input type="button" value="FUTURES"/> / <input type="button" value="SWAP"/> / <input type="button" value="OPTION"/>
instId	String	No	Instrument ID, e.g. <input type="button" value="BTC-USD-SWAP"/>

### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "instType": "SWAP",
      "instId": "BTC-USDT-SWAP",
      "markPx": "200",
      "ts": "1597026383085"
    }
  ]
}
```

] }

## RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type <code>MARGIN</code> <code>SWAP</code> <code>FUTURES</code> <code>OPTION</code>
instId	String	Instrument ID, e.g. <code>BTC-USD-200214</code>
markPx	String	Mark price
ts	String	Data return time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## GET POSITION TIERS

Retrieve position tiers information, maximum leverage depends on your borrowings and Maintenance margin ratio.

**RATE LIMIT: 10 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

### HTTP REQUEST

`GET /api/v5/public/position-tiers`

#### Request Example

```
import okx.PublicData as PublicData

flag = "0" # Production trading: 0, Demo trading: 1

publicDataAPI = PublicData.PublicAPI(flag=flag)

# Retrieve position tiers information
result = publicDataAPI.get_position_tiers(
    instType="SWAP",
    tdMode="cross",
    uly="BTC-USD"
)
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	Yes	Instrument type <code>MARGIN</code> <code>SWAP</code> <code>FUTURES</code> <code>OPTION</code>
tdMode	String	Yes	Trade mode Margin mode <code>cross</code> <code>isolated</code>
instFamily	String	Conditional	Single instrument family or multiple instrument families (no more than 5) separated with comma. If instType is <code>SWAP/FUTURES/OPTION</code> , <code>instFamily</code> is required.
instId	String	Conditional	Single instrument or multiple instruments (no more than 5) separated with comma. Either instId or ccy is required, if both are passed, instId will be used, ignore when instType is one of <code>SWAP</code> , <code>FUTURES</code> , <code>OPTION</code>

Parameter	Type	Required	Description
ccy	String	Conditional	Margin currency Only applicable to cross MARGIN. It will return borrowing amount for <a href="#">Multi-currency margin</a> and <a href="#">Portfolio margin</a> when <a href="#">ccy</a> takes effect.
tier	String	No	Tiers

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "baseMaxLoan": "50",
      "imr": "0.1",
      "instId": "BTC-USDT",
      "maxLever": "10",
      "maxSz": "50",
      "minSz": "0",
      "mmr": "0.03",
      "optMgnFactor": "0",
      "quoteMaxLoan": "500000",
      "tier": "1",
      "uly": "",
      "instFamily": ""
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
uly	String	Underlying Applicable to <a href="#">FUTURES</a> / <a href="#">SWAP</a> / <a href="#">OPTION</a>
instFamily	String	Instrument family Applicable to <a href="#">FUTURES</a> / <a href="#">SWAP</a> / <a href="#">OPTION</a>
instId	String	Instrument ID
tier	String	Tiers
minSz	String	The minimum borrowing amount or position of this gear is only applicable to margin/options/perpetual/delivery, the minimum position is 0 by default It will return the minimum borrowing amount when <a href="#">ccy</a> takes effect.
maxSz	String	The maximum borrowing amount or number of positions held in this position is only applicable to margin/options/perpetual/delivery It will return the maximum borrowing amount when <a href="#">ccy</a> takes effect.
mmr	String	Position maintenance margin requirement rate
imr	String	Initial margin requirement rate
maxLever	String	Maximum available leverage
optMgnFactor	String	Option Margin Coefficient (only applicable to options)
quoteMaxLoan	String	Quote currency borrowing amount (only applicable to leverage and the case when <a href="#">instId</a> takes effect)
baseMaxLoan	String	Base currency borrowing amount (only applicable to leverage and the case when <a href="#">instId</a> takes effect)

#### GET INTEREST RATE AND LOAN QUOTA

**RATE LIMIT: 2 REQUESTS PER 2 SECONDS****RATE LIMIT RULE: IP****HTTP REQUEST**

```
GET /api/v5/public/interest-rate-loan-quota
```

**Request Example**

```
import okx.PublicData as PublicData

flag = "0" # Production trading: 0, Demo trading: 1

publicDataAPI = PublicData.PublicAPI(flag=flag)

# Retrieve interest rate and loan quota
result = publicDataAPI.get_interest_rate_loan_quota()
print(result)
```

**Response Example**

```
{
  "code": "0",
  "data": [
    {
      "configCcyList": [
        {
          "ccy": "USDT",
          "rate": "0.00043728",
        }
      ],
      "basic": [
        {
          "ccy": "USDT",
          "quota": "500000",
          "rate": "0.00043728"
        },
        {
          "ccy": "BTC",
          "quota": "10",
          "rate": "0.00019992"
        }
      ],
      "vip": [
        {
          "irDiscount": "",
          "loanQuotaCoef": "6",
          "level": "VIP1"
        },
        {
          "irDiscount": "",
          "loanQuotaCoef": "7",
          "level": "VIP2"
        }
      ],
      "config": [
        {
          "ccy": "USDT",
          "stgyType": "0", // normal
          "quota": "xxxxxx",
          "level": "VIP 8"
        },
        .....
        {
          "ccy": "USDT",
          "stgyType": "1", // delta neutral
          "quota": "xxxxxx",
          "level": "VIP 1"
        },
        .....
      ],
      "regular": [
        .....
      ],
    }
  ]
}
```

```

        "irDiscount": "",
        "loanQuotaCoef": "1",
        "level": "Lv1"
    },
    {
        "irDiscount": "",
        "loanQuotaCoef": "2",
        "level": "Lv1"
    }
]
},
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
basic	Array of objects	Basic interest rate
> ccy	String	Currency
> rate	String	Daily borrowing rate
> quota	String	Max borrow
vip	Array of objects	Interest info for vip users
> level	String	VIP Level, e.g. <code>VIP1</code>
> loanQuotaCoef	String	Loan quota coefficient. Loan quota = <code>quota</code> * <code>level</code>
> irDiscount	String	<code>Interest rate discount</code> (Deprecated)
regular	Array of objects	Interest info for regular users
> level	String	Regular user Level, e.g. <code>Lv1</code>
> loanQuotaCoef	String	Loan quota coefficient. Loan quota = <code>quota</code> * <code>level</code>
> irDiscount	String	<code>Interest rate discount</code> (Deprecated)
configCcyList	Array of strings	Currencies that have loan quota configured using customized absolute value. Users should refer to config to get the loan quota of a currency which is listed in configCcyList, instead of getting it from basic/vip/regular.
> ccy	String	Currency
> rate	String	Daily rate
config	Array of objects	The currency details of loan quota configured using customized absolute value
> ccy	String	Currency

Parameter	Type	Description
> stgyType	String	<p>Strategy type  <input type="radio"/> 0: general strategy  <input checked="" type="radio"/> 1: delta neutral strategy</p> <p>If only <input type="radio"/> 0 is returned for a currency, it means the loan quota is shared between accounts in general strategy and accounts in delta neutral strategy; if both <input checked="" type="radio"/> 1 are returned for a currency, it means accounts in delta neutral strategy have separate loan quotas.</p>
> quota	String	Loan quota in absolute value
> level	String	VIP level

## GET UNDERLYING

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

### HTTP REQUEST

GET /api/v5/public/underlying

#### Request Example

```
import okx.PublicData as PublicData

flag = "0" # Production trading: 0, Demo trading: 1

publicDataAPI = PublicData.PublicAPI(flag=flag)

# Get underlying
result = publicDataAPI.get_underlying(
    instType="FUTURES"
)
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	Yes	<p>Instrument type  <input type="radio"/> SWAP  <input checked="" type="radio"/> FUTURES  <input type="radio"/> OPTION</p>

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    [
      "LTC-USDT",
      "BTC-USDT",
      "ETC-USDT"
    ]
  ]
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
uly	Array	Underlying

## GET SECURITY FUND

**RATE LIMIT: 10 REQUESTS PER 2 SECONDS****RATE LIMIT RULE: IP****HTTP REQUEST****GET /api/v5/public/insurance-fund****Request Example**

```
import okx.PublicData as PublicData

flag = "0" # Production trading: 0, Demo trading: 1

publicDataAPI = PublicData.PublicAPI(flag=flag)

# Get security fund balance information
result = publicDataAPI.get_insurance_fund(
    instType="SWAP",
    ccy="BTC-USD"
)
print(result)
```

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
instType	String	Yes	Instrument type <input type="button" value="MARGIN"/> <input type="button" value="SWAP"/> <input type="button" value="FUTURES"/> <input type="button" value="OPTION"/>
type	String	No	Type <input type="button" value="regular_update"/> <input type="button" value="liquidation_balance_deposit"/> <input type="button" value="bankruptcy_loss"/> <input type="button" value="platform_revenue"/> <input type="button" value="ad1"/> : ADL historical data The default is <input type="button" value="all type"/>
instFamily	String	Conditional	Instrument family Required for <input type="button" value="FUTURES"/> <input type="button" value="SWAP"/> <input type="button" value="OPTION"/>
ccy	String	Conditional	Currency, only applicable to <input type="button" value="MARGIN"/>
before	String	No	Pagination of data to return records newer than the requested <input type="button" value="ts"/>
after	String	No	Pagination of data to return records earlier than the requested <input type="button" value="ts"/>
limit	String	No	Number of results per request. The maximum is <input type="button" value="100"/> ; The default is <input type="button" value="100"/>

**Response Example**

```
{
  "code": "0",
  "data": [
    {
      "details": [
        {
          "adlType": "",
          "amt": "",
          "balance": "1343.1308",
          "ccy": "ETH",
          "maxBal": "",
          "maxBalTs": ""
        }
      ]
    }
  ]
}
```

```

        "ts": "1704883083000",
        "type": "regular_update"
    }
],
"instFamily": "ETH-USD",
"instType": "OPTION",
"total": "1369179138.7489"
}
],
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
total	String	The total balance of security fund, in <a href="#">USD</a>
instFamily	String	Instrument family Applicable to <a href="#">FUTURES</a> / <a href="#">SWAP</a> / <a href="#">OPTION</a>
instType	String	Instrument type
details	Array of objects	security fund data
> balance	String	The balance of security fund
> amt	String	The change in the balance of security fund Applicable when type is <a href="#">liquidation_balance_deposit</a> , <a href="#">bankruptcy_loss</a> or <a href="#">platform_revenue</a>
> ccy	String	The currency of security fund
> type	String	The type of security fund
> maxBal	String	Maximum security fund balance in the past eight hours Only applicable when type is <a href="#">ad1</a>
> maxBalTs	String	Timestamp when security fund balance reached maximum in the past eight hours, Unix timestamp format in milliseconds, e.g. <a href="#">1597026383085</a> Only applicable when type is <a href="#">ad1</a>
> decRate	String	Real-time security fund decline rate (compare balance and maxBal) Only applicable when type is <a href="#">ad1</a> (Deprecated)
> adlType	String	ADL related events <a href="#">rate_adl_start</a> : ADL begins due to high security fund decline rate <a href="#">bal_adl_start</a> : ADL begins due to security fund balance falling <a href="#">pos_adl_start</a> : ADL begins due to the volume of liquidation orders falls to a certain level (only applicable to premarket symbols) <a href="#">adl_end</a> : ADL ends Only applicable when type is <a href="#">ad1</a>
> ts	String	The update timestamp of security fund. Unix timestamp format in milliseconds, e.g. <a href="#">1597026383085</a>

The enumeration value `regular\_update` of type field is used to present up-to-minute security fund change. The amt field will be used to present the difference of security fund balance when the type field is `liquidation\_balance\_deposit`, `bankruptcy\_loss` or `platform\_revenue`, which is generated once per day around 08:00 am (UTC). When type is `regular\_update`, the amt field will be returned as `""`.

## UNIT CONVERT

Convert the crypto value to the number of contracts, or vice versa

**RATE LIMIT: 10 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

## HTTP REQUEST

GET /api/v5/public/convert-contract-coin

### Request Example

```
import okx.PublicData as PublicData

flag = "0" # Production trading: 0, Demo trading: 1

publicDataAPI = PublicData.PublicAPI(flag=flag)

# Convert the crypto value to the number of contracts, or vice versa
result = publicDataAPI.get_convert_contract_coin(
    instId="BTC-USD-SWAP",
    px="35000",
    sz="0.888"
)
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
type	String	No	Convert type ①: Convert currency to contract ②: Convert contract to currency The default is ①
instId	String	Yes	Instrument ID only applicable to <b>FUTURES</b> / <b>SWAP</b> / <b>OPTION</b>
sz	String	Yes	Quantity to buy or sell It is quantity of currency while converting currency to contract; It is quantity of contract while converting contract to currency.
px	String	Conditional	Order price For crypto-margined contracts, it is necessary while converting. For USDT-margined contracts, it is necessary while converting between usdt and contract. It is optional while converting between coin and contract. For OPTION, it is optional.
unit	String	No	The unit of currency <b>coin</b> <b>usds</b> : USDT/USDC The default is <b>coin</b> , only applicable to USD◎-margined contracts from <b>FUTURES</b> / <b>SWAP</b>
opType	String	No	Order type <b>open</b> : round down sz when opening positions <b>close</b> : round sz to the nearest when closing positions The default is <b>close</b> Applicable to <b>FUTURES</b> <b>SWAP</b>

### Response Example

```
{
    "code": "0",
    "data": [
        {
            "instId": "BTC-USD-SWAP",
            "px": "35000",
            "sz": "311",
            "type": "1",
            "unit": "coin"
        }
    ],
}
```

```

"msg": "",
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
type	String	Convert type ①: Convert currency to contract ②: Convert contract to currency
instId	String	Instrument ID
px	String	Order price
sz	String	Quantity to buy or sell It is quantity of contract while converting currency to contract It is quantity of currency while contract to currency.
unit	String	The unit of currency coin usds: USDT/USDC

## GET OPTION TICK BANDS

Get option tick bands information

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

### HTTP REQUEST

```
GET /api/v5/public/instrument-tick-bands
```

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instType	String	Yes	Instrument type OPTION
instFamily	String	No	Instrument family Only applicable to OPTION

Response Example

```

{
  "code": "0",
  "msg": "",
  "data": [
    {
      "instType": "OPTION",
      "instFamily": "BTC-USD",
      "tickBand": [
        {
          "minPx": "0",
          "maxPx": "100",
          "tickSz": "0.1"
        },
        {
          "minPx": "100",
          "maxPx": "10000",
          "tickSz": "1"
        }
      ]
    },
    {
      "instType": "OPTION"
    }
  ]
}

```

```
        "instFamily": "ETH-USD",
        "tickBand": [
            {
                "minPx": "0",
                "maxPx": "100",
                "tickSz": "0.1"
            },
            {
                "minPx": "100",
                "maxPx": "10000",
                "tickSz": "1"
            }
        ]
    }
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
instType	String	Instrument type
instFamily	String	Instrument family
tickBand	Array of objects	Tick size band
> minPx	String	Minimum price while placing an order
> maxPx	String	Maximum price while placing an order
> tickSz	String	Tick size, e.g. 0.0001

**GET PREMIUM HISTORY**

It will return premium data in the past 6 months.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

#### RATE LIMIT RULE: IP

## HTTP REQUEST

#### GET /api/v1/public/a/premium\_history

### Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USDT-SWAP</code> Applicable to <code>SWAP</code>
after	String	No	Pagination of data to return records earlier than the requested ts(not included)
before	String	No	Pagination of data to return records newer than the requested ts(not included)
limit	String	No	Number of results per request. The maximum is <code>100</code> . The default is <code>100</code> .

## Response Example

```
{  
  "code": "0",  
  "data": [  
    {
```

```
        "instId": "BTC-USDT-SWAP",
        "premium": "0.0000578896878167",
        "ts": "1713925924000"
    }
],
"msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Instrument ID, e.g. <code>BTC-USDT-SWAP</code>
premium	String	Premium index formula: $[\text{Max}(0, \text{Impact bid price} - \text{Index price}) - \text{Max}(0, \text{Index price} - \text{Impact ask price})] / \text{Index price}$
ts	String	Data generation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## GET INDEX TICKERS

Retrieve index tickers.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

## RATE LIMIT RULE: IP

## HTTP REQUEST

GET /api/v5/market/index-tickers

## Request Example

```
import okx.MarketData as MarketData

flag = "0" # Production trading:0 , demo trading:1

marketDataAPI = MarketData.MarketAPI(flag=flag)

# Retrieve index tickers
result = marketDataAPI.get_index_tickers(
    instId="BTC-USDT"
)
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
quoteCcy	String	Conditional	Quote currency Currently there is only an index with <code>USD/USDT/BTC/USDC</code> as the quote currency.
instId	String	Conditional	Index, e.g. <code>BTC-USD</code> Either <code>quoteCcy</code> or <code>instId</code> is required. Same as <code>uly</code> .

## Response Example

```
        "ts": "1649419644492"
```

```
}
```

```
]
```

```
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Index
idxPx	String	Latest index price
high24h	String	Highest price in the past 24 hours
low24h	String	Lowest price in the past 24 hours
open24h	String	Open price in the past 24 hours
sodUtc0	String	Open price in the UTC 0
sodUtc8	String	Open price in the UTC 8
ts	String	Index price update time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## GET INDEX CANDLESTICKS

Retrieve the candlestick charts of the index. This endpoint can retrieve the latest 1,440 data entries. Charts are returned in groups based on the requested bar.

### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: IP

### HTTP REQUEST

```
GET /api/v5/market/index-candles
```

#### Request Example

```
import okx.MarketData as MarketData

flag = "0" # Production trading:0 , demo trading:1

marketDataAPI = MarketData.MarketAPI(flag=flag)

# Retrieve the candlestick charts of the index
result = marketDataAPI.get_index_candlesticks(
    instId="BTC-USD"
)
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Index, e.g. <code>BTC-USD</code> Same as <code>uly</code> .
after	String	No	Pagination of data to return records earlier than the requested <code>ts</code>
before	String	No	Pagination of data to return records newer than the requested <code>ts</code> . The latest data will be returned when using <code>before</code> individually
bar	String	No	Bar size, the default is <code>1m</code> e.g. <code>[1m/3m/5m/15m/30m/1H/2H/4H]</code> UTC+8 opening price k-line: <code>[6H/12H/1D/1W/1M/3M]</code> UTC+0 opening price k-line: <code>[6Hutc/12Hutc/1Dutc/1Wutc/1Mutc/3Mutc]</code>

Parameter	Type	Required	Description
limit	String	No	Number of results per request. The maximum is <code>100</code> . The default is <code>100</code>

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    [
      "1597026383085",
      "3.721",
      "3.743",
      "3.677",
      "3.708",
      "0"
    ],
    [
      "1597026383085",
      "3.731",
      "3.799",
      "3.494",
      "3.72",
      "1"
    ]
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Opening time of the candlestick, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
o	String	Open price
h	String	highest price
l	String	Lowest price
c	String	Close price
confirm	String	The state of candlesticks. <code>0</code> represents that it is uncompleted, <code>1</code> represents that it is completed.

The candlestick data may be incomplete, and should not be polled repeatedly.

The data returned will be arranged in an array like this: [ts,o,h,l,c,confirm].

#### GET INDEX CANDLESTICKS HISTORY

Retrieve the candlestick charts of the index from recent years.

#### RATE LIMIT: 10 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: IP

#### HTTP REQUEST

```
GET /api/v5/market/history-index-candles
```

#### Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Index, e.g. <code>BTC-USD</code> Same as <code>uly</code> .
after	String	No	Pagination of data to return records earlier than the requested <code>ts</code>
before	String	No	Pagination of data to return records newer than the requested <code>ts</code> . The latest data will be returned when using <code>before</code> individually
bar	String	No	Bar size, the default is <code>1m</code> e.g. [1m/3m/5m/15m/30m/1H/2H/4H] UTC+8 opening price k-line: [6H/12H/1D/1W/1M] UTC+0 opening price k-line: [/6Hutc/12Hutc/1Dutc/1Wutc/1Mutc]
limit	String	No	Number of results per request. The maximum is <code>100</code> ; The default is <code>100</code>

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    [
      "1597026383085",
      "3.721",
      "3.743",
      "3.677",
      "3.708",
      "1"
    ],
    [
      "1597026383085",
      "3.731",
      "3.799",
      "3.494",
      "3.72",
      "1"
    ]
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Opening time of the candlestick, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
o	String	Open price
h	String	highest price
l	String	Lowest price
c	String	Close price
confirm	String	The state of candlesticks. ① represents that it is uncompleted, ② represents that it is completed.

The data returned will be arranged in an array like this: [ts,o,h,l,c,confirm].

Retrieve the candlestick charts of mark price. This endpoint can retrieve the latest 1,440 data entries. Charts are returned in groups based on the requested bar.

#### RATE LIMIT: 20 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: IP

#### HTTP REQUEST

```
GET /api/v5/market/mark-price-candles
```

#### Request Example

```
import okx.MarketData as MarketData

flag = "0" # Production trading:0 , demo trading:1

marketDataAPI = MarketData.MarketAPI(flag=flag)

# Retrieve the candlestick charts of mark price
result = marketDataAPI.get_mark_price_candlesticks(
    instId="BTC-USD-SWAP"
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USD-SWAP</code>
after	String	No	Pagination of data to return records earlier than the requested <code>ts</code>
before	String	No	Pagination of data to return records newer than the requested <code>ts</code> . The latest data will be returned when using <code>before</code> individually
bar	String	No	Bar size, the default is <code>1m</code> e.g. [1m/3m/5m/15m/30m/1H/2H/4H] UTC+8 opening price k-line: [6H/12H/1D/1W/1M/3M] UTC+0 opening price k-line: [6Hutc/12Hutc/1Dutc/1Wutc/1Mutc/3Mutc]
limit	String	No	Number of results per request. The maximum is <code>100</code> ; The default is <code>100</code>

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    [
      "1597026383085",
      "3.721",
      "3.743",
      "3.677",
      "3.708",
      "0"
    ],
    [
      "1597026383085",
      "3.731",
      "3.799",
      "3.494",
      "3.72",
      "1"
    ]
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Opening time of the candlestick, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
o	String	Open price
h	String	Highest price
l	String	Lowest price
c	String	Close price
confirm	String	The state of candlesticks. 0 represents that it is uncompleted, 1 represents that it is completed.

The candlestick data may be incomplete, and should not be polled repeatedly.

The data returned will be arranged in an array like this: [ts,o,h,l,c,confirm]

#### GET MARK PRICE CANDLESTICKS HISTORY

Retrieve the candlestick charts of mark price from recent years.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

#### HTTP REQUEST

`GET /api/v5/market/history-mark-price-candles`

Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	String	Yes	Instrument ID, e.g. <code>BTC-USD-SWAP</code>
after	String	No	Pagination of data to return records earlier than the requested <code>ts</code>
before	String	No	Pagination of data to return records newer than the requested <code>ts</code> . The latest data will be returned when using <code>before</code> individually
bar	String	No	Bar size, the default is <code>1m</code> e.g. [1m/3m/5m/15m/30m/1H/2H/4H] UTC+8 opening price k-line: [6H/12H/1D/1W/1M] UTC+0 opening price k-line: [6Hutc/12Hutc/1Dutc/1Wutc/1Mutc]
limit	String	No	Number of results per request. The maximum is <code>100</code> ; The default is <code>100</code>

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    [
      "1597026383085",
      "3.721",
      "3.743",
      "3.677",
      "3.708",
      "1"
    ]
  ]
}
```

```
],
[
  "1597026383085",
  "3.731",
  "3.799",
  "3.494",
  "3.72",
  "1"
]
]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Opening time of the candlestick, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
o	String	Open price
h	String	highest price
l	String	Lowest price
c	String	Close price
confirm	String	The state of candlesticks. 0 represents that it is uncompleted, 1 represents that it is completed.

The data returned will be arranged in an array like this: [ts,o,h,l,c,confirm]

## GET EXCHANGE RATE

This interface provides the average exchange rate data for 2 weeks

### RATE LIMIT: 1 REQUEST PER 2 SECONDS

### RATE LIMIT RULE: IP

### HTTP REQUEST

`GET /api/v5/market/exchange-rate`

#### Request Example

```
import okx.MarketData as MarketData

flag = "0" # Production trading:0 , demo trading:1

marketDataAPI = MarketData.MarketAPI(flag=flag)

# Retrieve average exchange rate data for 2 weeks
result = marketDataAPI.get_exchange_rate()
print(result)
```

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "usdCny": "7.162"
    }
  ]
}
```

]

}

## RESPONSE PARAMETERS

Parameter	Type	Description
usdCny	String	Exchange rate

## GET INDEX COMPONENTS

Get the index component information data on the market

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

### HTTP REQUEST

GET /api/v5/market/index-components

#### Request Example

```
import okx.MarketData as MarketData

flag = "0" # Production trading:0 , demo trading:1

marketDataAPI = MarketData.MarketAPI(flag=flag)

# Get the index component information data on the market
result = marketDataAPI.get_index_components(
    index="BTC-USD"
)
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
index	String	Yes	index, e.g <code>BTC-USDT</code> Same as <code>uly</code> .

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": {
    "components": [
      {
        "symbol": "BTC/USDT",
        "symPx": "52733.2",
        "wgt": "0.25",
        "cnvPx": "52733.2",
        "exch": "OKX"
      },
      {
        "symbol": "BTC/USDT",
        "symPx": "52739.87000000",
        "wgt": "0.25",
        "cnvPx": "52739.87000000",
        "exch": "Binance"
      },
      {
        "symbol": "BTC/USDT",
        "symPx": "52729.1",
        "wgt": "0.25",
        "cnvPx": "52729.1",
        "exch": "Huobi"
      },
      {
        "symbol": "BTC/USDT",
        "symPx": "52733.2",
        "wgt": "0.25",
        "cnvPx": "52733.2",
        "exch": "OKX"
      }
    ]
  }
}
```

```
        "symPx": "52739.47929397",
        "wgt": "0.25",
        "cnvPx": "52739.47929397",
        "exch": "Poloniex"
    },
],
"last": "52735.4123234925",
"index": "BTC-USDT",
"ts": "1630985335599"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
index	String	Index
last	String	Latest Index Price
ts	String	Data generation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
components	Array of objects	Components
> exch	String	Name of Exchange
> symbol	String	Name of Exchange Trading Pairs
> symPx	String	Price of Exchange Trading Pairs
> wgt	String	Weights
> cnvPx	String	Price converted to index

#### GET ECONOMIC CALENDAR DATA

Authentication is required for this endpoint. This endpoint is only supported in production environment.

Get the macro-economic calendar data within 3 months. Historical data from 3 months ago is only available to users with trading fee tier VIP1 and above.

**RATE LIMIT: 1 REQUEST PER 5 SECONDS**

#### RATE LIMIT RULE: IP

## HTTP REQUEST

GET /api/v5/public/economic-calendar

### Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
			dives, mali, malta, mauritania, mauritius, mexico, micronesia, moldova, monaco, mongolia, montenegro, morocco, mozambique, myanmar, namibia, nepal, netherlands, new_caledonia, new_zealand, nicaragua, niger, nigeria, north_korea, northern_mariana_islands, norway, opec, oman, pakistan, palau, palestine, panama, papua_new_guinea, paraguay, peru, philippines, poland, portugal, puerto_rico, qatar, russia, republic_of_the_congo, romania, russia, rwanda, slovakia, samoa, san_marino, sao_tome_and_principe, saudi_arabia, senegal, serbia, seychelles, sierra_leone, singapore, slovakia, slovenia, solomon_islands, somalia, south_africa, south_korea, south_sudan, spain, sri_lanka, st_kitts_and_nevis, st_lucia, sudan, suriname, swaziland, sweden, switzerland, syria, taiwan, tajikistan, tanzania, thailand, togo, tonga, trinidad_and_tobago, tunisia, turkey, turkmenistan, uganda, ukraine, united_arab_emirates, united_kingdom, united_states, uruguay, uzbekistan, vanuatu, venezuela, vietnam, world, yemen, zambia, zimbabwe
importance	string	No	<p>Level of importance</p> <p><input type="radio"/> 1: low</p> <p><input type="radio"/> 2: medium</p> <p><input type="radio"/> 3: high</p>
before	String	No	Pagination of data to return records newer than the requested ts based on the date parameter. Unix timestamp format in milliseconds.
after	String	No	Pagination of data to return records earlier than the requested ts based on the date parameter. Unix timestamp format in milliseconds. The default is the timestamp of the request moment.
limit	String	No	Number of results per request. The maximum is 100. The default is 100.

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "actual": "7.8%",
      "calendarId": "330631",
      "category": "Harmonised Inflation Rate YoY",
      "ccy": "",
      "date": "170012160000",
      "dateSpan": "0",
      "event": "Harmonised Inflation Rate YoY",
      "forecast": "7.8%",
      "importance": "1",
      "prevInitial": "",
      "previous": "9%",
      "refDate": "169871040000",
      "region": "Slovakia",
      "uTime": "1700121605007",
      "unit": "%"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
calendarId	string	Calendar ID
date	string	Estimated release time of the value of actual field, millisecond format of Unix timestamp, e.g. 1597026383085
region	string	Country, region or entity
category	string	Category name
event	string	Event name

Parameter	Type	Description
refDate	string	Date for which the datapoint refers to
actual	string	The actual value of this event
previous	string	Latest actual value of the previous period The value will be revised if revision is applicable
forecast	string	Average forecast among a representative group of economists
dateSpan	string	0: The time of the event is known 1: we only know the date of the event, the exact time of the event is unknown.
importance	string	Level of importance 1: low 2: medium 3: high
uTime	string	Update time of this record, millisecond format of Unix timestamp, e.g. 1597026383085
prevInitial	string	The initial value of the previous period Only applicable when revision happens
ccy	string	Currency of the data
unit	string	Unit of the data

## GET HISTORICAL MARKET DATA

### Data availability

Historical data backfill is currently in progress. Data availability may vary by module, instrument, and time period. The dataset will be continuously expanded to provide more comprehensive historical coverage.

### Legacy data format notice

For module 1 (trade history), some old historical files may contain column headers with both Chinese characters along with English column names. All the Chinese characters will be removed once the data backfill is done. Please account for this when parsing the data.

### Data release schedule

Most data for modules 1, 2, 3 is typically available on T+2; order book data is typically available on T+3.

Retrieve historical market data for OKX.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

**HTTP REQUEST**

GET /api/v5/public/market-data-history

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
module	String	Yes	Data module type 1: Trade history 2: 1-minute candlestick 3: Funding rate

Parameter	Type	Required	Description
			<p>5: 5000-level orderbook (from Nov 1, 2025)  6: 50-level orderbook</p>
instType	String	Yes	<p>Instrument type</p> <p>SPOT</p> <p>FUTURES</p> <p>SWAP</p> <p>OPTION</p>
instIdList	String	Conditional	<p>List of instrument IDs, e.g. <code>BTC-USDT</code>, or <code>ANY</code> for all instruments (<code>ANY</code> is only supported for module = 1, 2, 3 &amp; <code>dateAggrType</code> = <code>daily</code>)</p> <p>Multiple instrument IDs should be separated by commas, e.g. <code>BTC-USDT,ETH-USDT</code></p> <p>Maximum length = 10</p> <p>Only applicable when <code>instType</code> = <code>SPOT</code></p>
instFamilyList	String	Conditional	<p>List of instrument families, e.g. <code>BTC-USDT</code>, or <code>ANY</code> for all instruments (<code>ANY</code> is only supported for module = 1, 2, 3 &amp; <code>dateAggrType</code> = <code>daily</code>)</p> <p>Multiple instrument families should be separated by commas, e.g. <code>BTC-USDT,ETH-USDT</code></p> <p>Maximum length = 10 (= 1 when module = 6 &amp; <code>instType</code> = <code>OPTION</code>)</p> <p>Only applicable when <code>instType</code> ≠ <code>SPOT</code></p>
dateAggrType	String	Yes	<p>Date aggregation type</p> <p><code>daily</code> (not supported for module = 3 &amp; <code>instFamilyList</code> ≠ <code>ANY</code>)</p> <p><code>monthly</code> (not supported for module = 6)</p>
begin	String	Yes	<p>Begin timestamp. Unix timestamp format in milliseconds (inclusive)</p> <p>Maximum range: 20 days for daily, 20 months for monthly</p>
end	String	Yes	<p>End timestamp. Unix timestamp format in milliseconds (inclusive)</p> <p>When module = 6 &amp; <code>instType</code> = <code>OPTION</code>, only returns data for the day specified by <code>end</code></p>

### Response Example

```
{
  "code": "0",
  "data": [
    {
      "dateAggrType": "daily",
      "details": [
        {
          "dateRangeEnd": "1756656000000",
          "dateRangeStart": "1756569600000",
          "groupDetails": [
            {
              "dateTs": "1756656000000",
              "filename": "BTC-USDT-SWAP-trades-2025-09-01.zip",
              "sizeMB": "10.82",
              "url": "https://static.okx.com/cdn/okex/traderecords/trades/daily/20250901/BTC-USDT-SWAP-trades-2025-09-01.zip"
            },
            {
              "dateTs": "1756569600000",
              "filename": "BTC-USDT-SWAP-trades-2025-08-31.zip",
              "sizeMB": "4.82",
              "url": "https://static.okx.com/cdn/okex/traderecords/trades/daily/20250831/BTC-USDT-SWAP-trades-2025-08-31.zip"
            }
          ],
          "groupSizeMB": "15.64",
          "instFamily": "BTC-USDT",
          "instId": "",
          "instType": "SWAP"
        }
      ],
      "totalSizeMB": "15.64",
      "ts": "1756882260390"
    ],
    "msg": ""
  }
}
```

### Response Example when no data files are available

```
{
  "code": "0",
  "data": [
    {
      "dateAggrType": "monthly",
      "details": [],
      "totalSizeMB": "0",
      "ts": "1756889595507"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Response timestamp, Unix timestamp format in milliseconds
totalSizeMB	String	Total size of all data files in MB
dateAggrType	String	Date aggregation type <input checked="" type="button"/> daily <input type="button"/> monthly
details	Array	
> instId	String	Instrument ID
> instFamily	String	Instrument family
> dateRangeStart	String	Data range start date, Unix timestamp format in milliseconds (inclusive)
> dateRangeEnd	String	Data range end date, Unix timestamp format in milliseconds (inclusive)
> groupSizeMB	String	Data group size in MB
> groupDetails	Array	
>> filename	String	Data file name, e.g. <input type="button"/> BTC-USDT-SWAP-trades-2025-05-15.zip
>> dataTs	String	Data date timestamp, Unix timestamp format in milliseconds
>> sizeMB	String	File size in MB
>> url	String	Download URL

### Data query rules

- Only the date portion (yyyy-mm-dd) of timestamps is used; time components are ignored
- Both begin and end timestamps are inclusive
- Data is returned in reverse chronological order (closer to end first)
- If the query exceeds record limits, data closest to the end timestamp is returned
- Exception:** When module = 6 & instType = OPTION, only data for the day specified by the end is returned

### Timezone specifications for timestamp parsing

When converting Unix timestamps to dates, the following timezone conventions are applied to all timestamp fields (begin, end, dateRangeStart, dateRangeEnd, dataTs):

- Orderbook data** (modules 5, 6): UTC+0
- All other data modules** (modules 1, 2, 3): UTC+8

# WebSocket

## INSTRUMENTS CHANNEL

The instruments will be pushed if there is any change to the instrument's state (such as delivery of FUTURES, exercise of OPTION, listing of new contracts / trading pairs, trading suspension, etc.).

(The full instrument list is not pushed since December 28, 2022, you can click [here](#) to view details)

## URL PATH

/ws/v5/public

### Request Example

```
import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/public")
    await ws.start()
    args = [
        {
            "channel": "instruments",
            "instType": "SPOT"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>instruments</code>
> instType	String	Yes	Instrument type <code>SPOT</code> <code>MARGIN</code> <code>SWAP</code> <code>FUTURES</code> <code>OPTION</code>

### Successful Response Example

```
{  
    "id": "1512",  
    "event": "subscribe",  
    "channel": "instruments",  
    "instType": "SPOT",  
    "args": [{"channel": "instruments", "instType": "SPOT"}],  
    "time": "2022-12-28T10:00:00Z",  
    "type": "subscribe",  
    "status": "ok",  
    "code": 200, "msg": ""}
```

```

"arg": {
  "channel": "instruments",
  "instType": "SPOT"
},
"connId": "a4d3ae55"
}

```

## Failure Response Example

```

{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \"subscribe\", \\"args\\\": [{ \\"channel\\\" : \\"instruments\\\", \\"instType\\\" : \\"FUTURES\\\" }]}",
  "connId": "a4d3ae55"
}

```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event   
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> instType	String	Yes	Instrument type     
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

## Push Data Example

```

{
  "arg": {
    "channel": "instruments",
    "instType": "SPOT"
  },
  "data": [
    {
      "alias": "",
      "auctionEndTime": "",
      "baseCcy": "BTC",
      "category": "1",
      "cttMult": "",
      "cttType": "",
      "cttVal": "",
      "cttValcny": "",
      "contTdsTime": "1704876947000",
      "expTime": "",
      "futureSettlement": false,
      "instFamily": "",
      "instId": "BTC-USDT",
      "instType": "SPOT",
      "lever": "10",
      "marginType": "SPOT"
    }
  ]
}

```

```

"listTime": "1606468572000",
"lotSz": "0.0000001",
"maxIcebergSz": "999999999.0000000000000000",
"maxLmtAmt": "100000",
"maxLmtSz": "999999999",
"maxMktAmt": "100000",
"maxMktSz": "",
"maxStopSz": "",
"maxTriggerSz": "999999999.0000000000000000",
"maxTwapSz": "999999999.0000000000000000",
"minSz": "0.00001",
"optType": "",
"openType": "call_auction",
"preMktSwTime": "",
"quoteCcy": "USDT",
"settleCcy": "",
"state": "live",
"ruleType": "normal",
"stk": "",
"tickSz": "0.1",
"uly": "",
"instIdCode": 100000000
}
]
}

```

#### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Subscribed channel
> channel	String	Channel name
> instType	String	Instrument type
data	Array of objects	Subscribed data
> instType	String	Instrument type
> instId	String	Instrument ID, e.g. <span style="border: 1px solid black; padding: 2px;">BTC-USDT</span>
> uly	String	Underlying, e.g. <span style="border: 1px solid black; padding: 2px;">BTC-USD</span> Only applicable to <span style="border: 1px solid black; padding: 2px;">FUTURES</span> / <span style="border: 1px solid black; padding: 2px;">SWAP</span> / <span style="border: 1px solid black; padding: 2px;">OPTION</span>
> instFamily	String	Instrument family, e.g. <span style="border: 1px solid black; padding: 2px;">BTC-USD</span> Only applicable to <span style="border: 1px solid black; padding: 2px;">FUTURES</span> / <span style="border: 1px solid black; padding: 2px;">SWAP</span> / <span style="border: 1px solid black; padding: 2px;">OPTION</span>
> category	String	Currency category. Note: this parameter is already deprecated
> baseCcy	String	Base currency, e.g. <span style="border: 1px solid black; padding: 2px;">BTC</span> in <span style="border: 1px solid black; padding: 2px;">BTC-USDT</span> Only applicable to <span style="border: 1px solid black; padding: 2px;">SPOT</span> / <span style="border: 1px solid black; padding: 2px;">MARGIN</span>
> quoteCcy	String	Quote currency, e.g. <span style="border: 1px solid black; padding: 2px;">USDT</span> in <span style="border: 1px solid black; padding: 2px;">BTC-USDT</span> Only applicable to <span style="border: 1px solid black; padding: 2px;">SPOT</span> / <span style="border: 1px solid black; padding: 2px;">MARGIN</span>
> settleCcy	String	Settlement and margin currency, e.g. <span style="border: 1px solid black; padding: 2px;">BTC</span> Only applicable to <span style="border: 1px solid black; padding: 2px;">FUTURES</span> / <span style="border: 1px solid black; padding: 2px;">SWAP</span> / <span style="border: 1px solid black; padding: 2px;">OPTION</span>
> ctVal	String	Contract value
> ctMult	String	Contract multiplier
> ctValCcy	String	Contract value currency
> optType	String	Option type (C: Call

Parameter	Type	Description
		<p>P: Put Only applicable to <code>OPTION</code></p>
> stk	String	Strike price Only applicable to <code>OPTION</code>
> listTime	String	Listing time Only applicable to <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>
> auctionEndTime	String	The end time of call auction, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code> Only applicable to <code>SPOT</code> that are listed through call auctions, return "" in other cases (deprecated, use <code>contTdSwTime</code> )
> contTdSwTime	String	Continuous trading switch time. The switch time from call auction, prequote to continuous trading, Unix timestamp format in milliseconds. e.g. <code>1597026383085</code> . Only applicable to <code>SPOT</code> / <code>MARGIN</code> that are listed through call auction or prequote, return "" in other cases.
> preMktSwTime	String	The time premarket swap switched to normal swap, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code> . Only applicable premarket <code>SWAP</code>
> openType	String	Open type <code>fix_price</code> : fix price opening <code>pre_quote</code> : pre-quote <code>call_auction</code> : call auction Only applicable to <code>SPOT</code> / <code>MARGIN</code> , return "" for all other business lines
> expTime	String	Expiry time Applicable to <code>SPOT</code> / <code>MARGIN</code> / <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code> . For <code>FUTURES</code> / <code>OPTION</code> , it is the delivery/exercise time. It can also be the delisting time of the trading instrument. Update once change.
> lever	String	Max Leverage Not applicable to <code>SPOT</code> / <code>OPTION</code> , used to distinguish between <code>MARGIN</code> and <code>SPOT</code> .
> tickSz	String	Tick size, e.g. <code>0.0001</code> For Option, it is minimum tickSz among tick band.
> lotSz	String	Lot size If it is a derivatives contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in <code>base currency</code>
> minSz	String	Minimum order size If it is a derivatives contract, the value is the number of contracts. If it is <code>SPOT</code> / <code>MARGIN</code> , the value is the quantity in <code>base currency</code>
> ctType	String	Contract type <code>linear</code> : linear contract <code>inverse</code> : inverse contract Only applicable to <code>FUTURES</code> / <code>SWAP</code>
> alias	String	<p>Alias</p> <code>this_week</code> <code>next_week</code> <code>this_month</code> <code>next_month</code> <code>quarter</code> <code>next_quarter</code> Only applicable to <code>FUTURES</code> <b>Not recommended for use, users are encouraged to rely on the expTime field to determine the delivery time of the contract</b>
> state	String	Instrument status <code>live</code>

Parameter	Type	Description
		<p><code>suspend</code>  <code>expired</code>  <code>preopen</code>: e.g. There will be preopen before the Futures and Options new contracts state is live.  <code>test</code>: Test pairs, can't be traded</p>
> state	String	<p>Instrument status</p> <p><code>live</code>  <code>suspend</code>  <code>expired</code>  <code>preopen</code>: e.g. Futures and options contracts rollover from generation to trading start; certain symbols before they go live  <code>test</code>: Test pairs, can't be traded</p>
> ruleType	String	<p>Trading rule types</p> <p><code>normal</code>: normal trading  <code>pre_market</code>: pre-market trading</p>
> maxLmtSz	String	<p>The maximum order quantity of a single limit order.  If it is a derivatives contract, the value is the number of contracts.  If it is <code>SPOT</code>/<code>MARGIN</code>, the value is the quantity in <code>base currency</code>.</p>
> maxMktSz	String	<p>The maximum order quantity of a single market order.  If it is a derivatives contract, the value is the number of contracts.  If it is <code>SPOT</code>/<code>MARGIN</code>, the value is the quantity in <code>USDT</code>.</p>
> maxTwapSz	String	<p>The maximum order quantity of a single TWAP order.  If it is a derivatives contract, the value is the number of contracts.  If it is <code>SPOT</code>/<code>MARGIN</code>, the value is the quantity in <code>base currency</code>.</p>
> maxIcebergSz	String	<p>The maximum order quantity of a single iceBerg order.  If it is a derivatives contract, the value is the number of contracts.  If it is <code>SPOT</code>/<code>MARGIN</code>, the value is the quantity in <code>base currency</code>.</p>
> maxTriggerSz	String	<p>The maximum order quantity of a single trigger order.  If it is a derivatives contract, the value is the number of contracts.  If it is <code>SPOT</code>/<code>MARGIN</code>, the value is the quantity in <code>base currency</code>.</p>
> maxStopSz	String	<p>The maximum order quantity of a single stop market order.  If it is a derivatives contract, the value is the number of contracts.  If it is <code>SPOT</code>/<code>MARGIN</code>, the value is the quantity in <code>USDT</code>.</p>
> futureSettlement	Boolean	<p>Whether daily settlement for expiry feature is enabled  Applicable to <code>FUTURES</code> <code>cross</code></p>
> instIdCode	Integer	<p>Instrument ID code.  For simple binary encoding, you must use <code>instIdCode</code> instead of <code>instId</code>.  For the same <code>instId</code>, its value may be different between production and demo trading.</p>

Instrument status will trigger pushing of incremental data from instruments channel. When a new contract is going to be listed, the instrument data of the new contract will be available with status preopen. When a product is going to be delisted (e.g. when a FUTURES contract is settled or OPTION contract is exercised), the instrument status will be changed to expired.

#### listTime and contTdSwTime

For spot symbols listed through a call auction or pre-open, listTime represents the start time of the auction or pre-open, and contTdSwTime indicates the end of the auction or pre-open and the start of continuous trading. For other scenarios, listTime will mark the beginning of continuous trading, and contTdSwTime will return an empty value "".

state

The state will always change from `preopen` to `live` when the listTime is reached. Certain symbols will now have `state:preopen` before they go live. Before going live, the instruments channel will push data for pre-listing symbols with `state:preopen`. If the listing is cancelled, the channel will send full data excluding the cancelled symbol, without additional notification. When the symbol goes live (reaching listTime), the channel will push data with `state:live`. Users can also query the corresponding data via the REST endpoint.

When a product is going to be delisted (e.g. when a FUTURES contract is settled or OPTION contract is exercised), the instrument will not be available.

## OPEN INTEREST CHANNEL

Retrieve the open interest. Data will be pushed every 3 seconds when there are updates.

## URL PATH

/ws/v5/public

## Request Example

```
import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/public")
    await ws.start()
    args = [
        {
            "channel": "open-interest",
            "instId": "LTC-USD-SWAP"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>open-interest</code>
> instId	String	Yes	Instrument ID

### Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
```

```

"arg": {
  "channel": "open-interest",
  "instId": "LTC-USD-SWAP"
},
"connId": "a4d3ae55"
}

```

## Failure Response Example

```

{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \"subscribe\", \\"args\\\": [{ \\"channel\\\" : \"open-interest\", \\"instId\\\" : \"LTC-USD-SWAP\\\" }]}",
  "connId": "a4d3ae55"
}

```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event   
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> instId	String	Yes	Instrument ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

## Push Data Example

```

{
  "arg": {
    "channel": "open-interest",
    "instId": "BTC-USDT-SWAP"
  },
  "data": [
    {
      "instId": "BTC-USDT-SWAP",
      "instType": "SWAP",
      "oi": "2216113.01000000309",
      "oiCcy": "22161.1301000000309",
      "oiUsd": "1939251795.54769270396321",
      "ts": "1743041250440"
    }
  ]
}

```

## PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name

Parameter	Type	Description
> instId	String	Instrument ID
data	Array of objects	Subscribed data
> instType	String	Instrument type
> instId	String	Instrument ID, e.g. <code>BTC-USDT-SWAP</code>
> oi	String	Open interest, in units of contracts.
> oiCcy	String	Open interest, in currency units, like BTC.
> oiUsd	String	Open interest in number of USD
> ts	String	The time when the data was updated, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## FUNDING RATE CHANNEL

Retrieve funding rate. Data will be pushed in 30s to 90s.

### URL PATH

/ws/v5/public

#### Request Example

```
import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/public")
    await ws.start()
    args = [
        {
            "channel": "funding-rate",
            "instId": "BTC-USD-SWAP"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels

Parameter	Type	Required	Description
> channel	String	Yes	Channel name funding-rate
> instId	String	Yes	Instrument ID

#### Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "funding-rate",
    "instId": "BTC-USD-SWAP"
  },
  "connId": "a4d3ae55"
}
```

#### Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\": \"funding-rate\", \"instId\": \"BTC-USD-SWAP\"}]}",
  "connId": "a4d3ae55"
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event subscribe unsubscribe error
arg	Object	No	Subscribed channel
> channel	String	yes	Channel name
> instId	String	No	Instrument ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

#### Push Data Example

```
{
  "arg": {
    "channel": "funding-rate",
    "instId": "BTC-USD-SWAP"
  },
  "data": [
    {
      "formulaType": "noRate",
      "fundingRate": "0.0001875391284828",
      "fundingTime": "170072640000",
      "impactValue": "",
      "instId": "BTC-USD-SWAP",
      "instType": "SWAP",
      "rate": "0.0001875391284828"
    }
  ]
}
```

```

"interestRate": "",
"method": "current_period",
"maxFundingRate": "0.00375",
"minFundingRate": "-0.00375",
"nextFundingRate": "",
"nextFundingTime": "1700755200000",
"premium": "0.0001233824646391",
"settFundingRate": "0.0001699799259033",
"settState": "settled",
"ts": "1700724675402"
}
]
}

```

## PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> instId	String	Instrument ID
data	Array of objects	Subscribed data
> instType	String	Instrument type, <code>SWAP</code>
> instId	String	Instrument ID, e.g. <code>BTC-USD-SWAP</code>
> method	String	Funding rate mechanism <code>current_period</code> <code>next_period</code> (no longer supported)
> formulaType	String	Formula type <code>noRate</code> : Old funding rate formula <code>withRate</code> : new funding rate formula
> fundingRate	String	Current funding rate
> nextFundingRate	String	Forecasted funding rate for the next period The nextFundingRate will be "" if the method is <code>current_period</code> (no longer supported)
> fundingTime	String	Settlement time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> nextFundingTime	String	Forecasted funding time for the next period, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> minFundingRate	String	The lower limit of the predicted funding rate of the next cycle
> maxFundingRate	String	The upper limit of the predicted funding rate of the next cycle
> interestRate	String	Interest rate
> impactValue	String	Depth weighted amount (in the unit of quote currency)
> settState	String	Settlement state of funding rate <code>processing</code> <code>settled</code>
> settFundingRate	String	If settState = <code>processing</code> , it is the funding rate that is being used for current settlement cycle. If settState = <code>settled</code> , it is the funding rate that is being used for previous settlement cycle
> premium	String	Premium index formula: $[\text{Max}(0, \text{Impact bid price} - \text{Index price}) - \text{Max}(0, \text{Index price} - \text{Impact ask price})] / \text{Index price}$
> ts	String	Data return time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

For some altcoins perpetual swaps with significant fluctuations in funding rates, OKX will closely monitor market changes. When necessary, the funding rate collection frequency, currently set at 8 hours, may be adjusted to higher frequencies such as 6 hours, 4 hours, 2 hours, or 1 hour. Thus, users should focus on the difference between `fundingTime` and `nextFundingTime` fields to determine the funding fee interval of a contract.

## PRICE LIMIT CHANNEL

Retrieve the maximum buy price and minimum sell price of instruments. Data will be pushed every 200ms when there are changes in limits, and will not be pushed when there is no changes on limit.

### URL PATH

/ws/v5/public

#### Request Example

```
import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/public")
    await ws.start()
    args = [
        {
            "channel": "price-limit",
            "instId": "LTC-USD-190628"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>price-limit</code>
> instId	String	Yes	Instrument ID

#### Successful Response Example

```
{  
    "id": "1512",  
    "event": "subscribe",  
    "arg": {  
        "channel": "price-limit",  
        "instId": "LTC-USD-190628"  
    }  
}
```

```
{
  "connId": "a4d3ae55"
}
```

## Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \"subscribe\", \\"argss\\": [{ \\"channel\\\" : \"price-limit\", \\"instId\\\" : \"LTC-USD-190628\"}]}",
  "connId": "a4d3ae55"
}
```

### RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event   
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> instId	String	Yes	Instrument ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

## Push Data Example

```
{
  "arg": {
    "channel": "price-limit",
    "instId": "LTC-USD-190628"
  },
  "data": [
    {
      "instId": "LTC-USD-190628",
      "buyLmt": "200",
      "sellLmt": "300",
      "ts": "1597026383085",
      "enabled": true
    }
  ]
}
```

### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> instId	String	Instrument ID
data	Array of objects	Subscribed data
> instType	String	Instrument type

Parameter	Type	Description
> instId	String	Instrument ID, e.g. <code>BTC-USDT</code>
> buyLmt	String	Maximum buy price Return "" when enabled is false
> sellLmt	String	Minimum sell price Return "" when enabled is false
> ts	String	Price update time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> enabled	Boolean	Whether price limit is effective <code>true</code> : the price limit is effective <code>false</code> : the price limit is not effective

## OPTION SUMMARY CHANNEL

Retrieve detailed pricing information of all OPTION contracts. Data will be pushed at once.

### URL PATH

/ws/v5/public

#### Request Example

```
import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/public")
    await ws.start()
    args = [
        {
            "channel": "opt-summary",
            "instFamily": "BTC-USD"
        }
    ]

    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>opt-summary</code>

Parameter	Type	Required	Description
> instFamily	String	Yes	Instrument family

### Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "opt-summary",
    "instFamily": "BTC-USD"
  },
  "connId": "a4d3ae55"
}
```

### Failure example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \\"subscribe\\", \\"argss\\": [{ \\"channel\\": \\"opt-summary\\", \\"uly\\": \\"BTC-USD\\"}]}",
  "connId": "a4d3ae55"
}
```

### RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event   
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> instFamily	String	Yes	Instrument family
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

### Push Data Example

```
{
  "arg": {
    "channel": "opt-summary",
    "instFamily": "BTC-USD"
  },
  "data": [
    {
      "instType": "OPTION",
      "instId": "BTC-USD-241013-70000-P",
      "uly": "BTC-USD",
      "delta": "-1.1180902625",
      "gamma": "2.2361957091",
      "vega": "0.000000001",
      "theta": "0.0000032334",
      "lever": "8.465747567",
      "r": "0.0001"
    }
  ]
}
```

```

    "markVol": "0.3675503331",
    "bidVol": "0",
    "askVol": "1.1669998535",
    "realVol": "",
    "deltaBS": "-0.9999672034",
    "gammaBS": "0.000000002",
    "thetaBS": "28.2649858387",
    "vegaBS": "0.0000114332",
    "ts": "1728703155650",
    "fwdPx": "62604.6993093463",
    "volLv": "0.2044711229"
  }
]
}

```

#### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> instFamily	String	Instrument family
data	Array of objects	Subscribed data
> instType	String	Instrument type, <span style="border: 1px solid #ccc; padding: 2px;">OPTION</span>
> instId	String	Instrument ID
> uly	String	Underlying
> delta	String	Sensitivity of option price to <span style="border: 1px solid #ccc; padding: 2px;">uly</span> price
> gamma	String	The delta is sensitivity to <span style="border: 1px solid #ccc; padding: 2px;">uly</span> price
> vega	String	Sensitivity of option price to implied volatility
> theta	String	Sensitivity of option price to remaining maturity
> deltaBS	String	Sensitivity of option price to <span style="border: 1px solid #ccc; padding: 2px;">uly</span> price in BS mode
> gammaBS	String	The delta is sensitivity to <span style="border: 1px solid #ccc; padding: 2px;">uly</span> price in BS mode
> vegaBS	String	Sensitivity of option price to implied volatility in BS mode
> thetaBS	String	Sensitivity of option price to remaining maturity in BS mode
> lever	String	Leverage
> markVol	String	Mark volatility
> bidVol	String	Bid volatility
> askVol	String	Ask Volatility
> realVol	String	Realized volatility (not currently used)
> volLv	String	Implied volatility of at-the-money options
> fwdPx	String	Forward price
> ts	String	Price update time, Unix timestamp format in milliseconds, e.g. <span style="border: 1px solid #ccc; padding: 2px;">1597026383085</span>

Retrieve the estimated delivery/exercise/settlement price of **FUTURES** and **OPTION** contracts.

Only the estimated price will be pushed in an hour before delivery/exercise/settlement, and will be pushed if there is any price change.

## URL PATH

/ws/v5/public

### Request Example

```
import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/public")
    await ws.start()
    args = [
        {
            "channel": "estimated-price",
            "instType": "FUTURES",
            "instFamily": "BTC-USD"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <b>subscribe</b> <b>unsubscribe</b>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <b>estimated-price</b>
> instType	String	Yes	Instrument type <b>OPTION</b> <b>FUTURES</b>
> instFamily	String	Conditional	Instrument family Either <b>instFamily</b> or <b>instId</b> is required.
> instId	String	Conditional	Instrument ID Either <b>instFamily</b> or <b>instId</b> is required.

### Successful Response Example

```
{  
    "id": "1512",
```

```

"event": "subscribe",
"arg": {
  "channel": "estimated-price",
  "instType": "FUTURES",
  "instFamily": "BTC-USD"
},
"connId": "a4d3ae55"
}

```

## Failure Response Example

```

{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"estimated-price\", \"instId\" : \"FUTURES\", \"uly\" : \"BTC-USD\" }]}",
  "connId": "a4d3ae55"
}

```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event subscribe unsubscribe error
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> instType	String	Yes	Instrument type OPTION FUTURES
> instFamily	String	Conditional	Instrument family
> instId	String	Conditional	Instrument ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

## Push Data Example

```

{
  "arg": {
    "channel": "estimated-price",
    "instType": "FUTURES",
    "instFamily": "XRP-USDT"
  },
  "data": [
    {
      "instId": "XRP-USDT-250307",
      "instType": "FUTURES",
      "settlePx": "2.4230631578947368",
      "settleType": "settlement",
      "ts": "1741244598708"
    }
  ]
}

```

## PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> instType	String	Instrument type <code>FUTURES</code> <code>OPTION</code>
> instFamily	String	Instrument family
> instId	String	Instrument ID
data	Array of objects	Subscribed data
> instType	String	Instrument type
> instId	String	Instrument ID, e.g. <code>BTC-USD-170310</code>
> settleType	String	Type <code>settlement</code> : Futures settlement <code>delivery</code> : Futures delivery <code>exercise</code> : Option exercise
> settlePx	String	Estimated price
> ts	String	Data update time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## MARK PRICE CHANNEL

Retrieve the mark price. Data will be pushed every 200 ms when the mark price changes, and will be pushed every 10 seconds when the mark price does not change.

### URL PATH

/ws/v5/public

#### Request Example

```
import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/public")
    await ws.start()
    args = [
        {"channel": "mark-price",
        "instId": "BTC-USDT"
    }]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request.

Parameter	Type	Required	Description
			A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation  
			A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name 
> instId	String	Yes	Instrument ID

#### Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "mark-price",
    "instId": "LTC-USD-190628"
  },
  "connId": "a4d3ae55"
}
```

#### Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \\"subscribe\\", \\"argss\\": [{ \\"channel\\": \\"mark-price\\", \\"instId\\": \\"LTC-USD-190628\\\"}]}",
  "connId": "a4d3ae55"
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
			Event
event	String	Yes	  
			Subscribed channel
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> instId	String	No	Instrument ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

#### Push Data Example

```
{
  "arg": {
    "channel": "mark-price",
    "instId": "LTC-USD-190628"
}
```

```

},
"data": [
{
  "instType": "FUTURES",
  "instId": "LTC-USD-190628",
  "markPx": "0.1",
  "ts": "1597026383085"
}
]
}

```

## PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> instId	String	Instrument ID
data	Array of objects	Subscribed data
> instType	String	Instrument type
> instId	String	Instrument ID
> markPx	String	Mark price
> ts	String	Price update time, Unix timestamp format in milliseconds, e.g. 1597026383085

## INDEX TICKERS CHANNEL

Retrieve index tickers data. Push data every 100ms if there are any changes, otherwise push once a minute.

### URL PATH

/ws/v5/public

#### Request Example

```

import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/public")
    await ws.start()
    args = [
        {
            "channel": "index-tickers",
            "instId": "BTC-USDT"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	<a href="#">subscribe</a> <a href="#">unsubscribe</a>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <a href="#">index-tickers</a>
> instId	String	Yes	Index with USD, USDT, BTC, USDC as the quote currency, e.g. <a href="#">BTC-USDT</a> Same as <a href="#">uly</a> .

#### Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "index-tickers",
    "instId": "BTC-USDT"
  },
  "connId": "a4d3ae55"
}
```

#### Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \\"subscribe\\", \\"args\\": [{ \\"channel\\": \\"index-tickers\\\", \\"instId\\": \\"BTC-USDT\\\" }]}",
  "connId": "a4d3ae55"
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	<a href="#">subscribe</a> <a href="#">unsubscribe</a> <a href="#">error</a>
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name <a href="#">index-tickers</a>
> instId	String	Yes	Index with USD, USDT, BTC, USDC as the quote currency, e.g. <a href="#">BTC-USDT</a>
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

#### Push Data Example

```
{
  "arg": {
    "channel": "index-tickers",
    "instId": "BTC-USDT"
  }
}
```

```

},
"data": [
{
  "instId": "BTC-USDT",
  "idxPx": "0.1",
  "high24h": "0.5",
  "low24h": "0.1",
  "open24h": "0.1",
  "sodUtc0": "0.1",
  "sodUtc8": "0.1",
  "ts": "1597026383085"
}
]
}

```

## PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> instId	String	Index with USD, USDT, or BTC as quote currency, e.g. <code>BTC-USDT</code> .
data	Array of objects	Subscribed data
> instId	String	Index
> idxPx	String	Latest Index Price
> open24h	String	Open price in the past 24 hours
> high24h	String	Highest price in the past 24 hours
> low24h	String	Lowest price in the past 24 hours
> sodUtc0	String	Open price in the UTC 0
> sodUtc8	String	Open price in the UTC 8
> ts	String	Update time of the index ticker, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## MARK PRICE CANDLESTICKS CHANNEL

Retrieve the candlesticks data of the mark price. The push frequency is the fastest interval 1 second push the data.

### URL PATH

/ws/v5/business

#### Request Example

```

import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/business")
    await ws.start()
    args = [
        {
            "channel": "mark-price-candleID",
            "instId": "BTC-USD-190628"
        }
    ]

    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

```

```

await ws.unsubscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>mark-price-candle3M</code> <code>mark-price-candle1M</code> <code>mark-price-candle1W</code> <code>mark-price-candle1D</code> <code>mark-price-candle2D</code> <code>mark-price-candle3D</code> <code>mark-price-candle5D</code> <code>mark-price-candle12H</code> <code>mark-price-candle6H</code> <code>mark-price-candle4H</code> <code>mark-price-candle2H</code> <code>mark-price-candle1H</code> <code>mark-price-candle30m</code> <code>mark-price-candle15m</code> <code>mark-price-candle5m</code> <code>mark-price-candle3m</code> <code>mark-price-candle1m</code> <code>mark-price-candle1Yutc</code> <code>mark-price-candle3Mutc</code> <code>mark-price-candle1Mutc</code> <code>mark-price-candle1Wutc</code> <code>mark-price-candle1Dutc</code> <code>mark-price-candle2Dutc</code> <code>mark-price-candle3Dutc</code> <code>mark-price-candle5Dutc</code> <code>mark-price-candle12Hutc</code> <code>mark-price-candle6Hutc</code>
> instId	String	Yes	Instrument ID

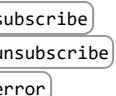
## Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "mark-price-candle1D",
    "instId": "BTC-USD-190628"
  },
  "connId": "a4d3ae55"
}
```

## Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \"subscribe\\\", \\"argss\\\": [{ \\"channel\\\" : \"mark-price-candle1D\\\", \\"instId\\\" : \"BTC-USD-190628\\\"}]}",
  "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event 
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> instId	String	Yes	Instrument ID
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

## Push Data Example

```
{
  "arg": {
    "channel": "mark-price-candle1D",
    "instId": "BTC-USD-190628"
  },
  "data": [
    ["1597026383085", "3.721", "3.743", "3.677", "3.708", "0"],
    ["1597026383085", "3.731", "3.799", "3.494", "3.72", "1"]
  ]
}
```

## PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> instId	String	Instrument ID
data	Array of Arrays	Subscribed data
> ts	String	Opening time of the candlestick, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> o	String	Open price
> h	String	Highest price
> l	String	Lowest price
> c	String	Close price

Parameter	Type	Description
> confirm	String	The state of candlesticks. 0 represents that it is uncompleted, 1 represents that it is completed.

## INDEX CANDLESTICKS CHANNEL

Retrieve the candlesticks data of the index. The push frequency is the fastest interval 1 second push the data. .

### URL PATH

/ws/v5/business

#### Request Example

```
import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/business")
    await ws.start()
    args = [
        {
            "channel": "index-candle30m",
            "instId": "BTC-USD"
        }
    ]

    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>index-candle3M</code> <code>index-candle1M</code> <code>index-candle1W</code> <code>index-candle1D</code> <code>index-candle2D</code> <code>index-candle3D</code> <code>index-candle5D</code> <code>index-candle12H</code> <code>index-candle6H</code> <code>index-candle4H</code> <code>index-candle2H</code> <code>index-candle1H</code> <code>index-candle30m</code> <code>index-candle15m</code>

Parameter	Type	Required	Description
			<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">index-candle5m</div> <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">index-candle3m</div> <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">index-candle1m</div> <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">index-candle3Mutc</div> <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">index-candle1Mutc</div> <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">index-candle1Wutc</div> <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">index-candle1Dutc</div> <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">index-candle2Dutc</div> <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">index-candle3Dutc</div> <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">index-candle5Dutc</div> <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">index-candle12Hutc</div> <div style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">index-candle6Hutc</div> </div>
> instId	String	Yes	Index, e.g. <span style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">BTC-USD</span> Same as <span style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">uly</span> .

### Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "index-candle30m",
    "instId": "BTC-USD"
  },
  "connId": "a4d3ae55"
}
```

### Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \\"subscribe\\", \\"argss\\": [{ \\"channel\\": \\"index-candle30m\\", \\"instId\\": \\"BTC-USD\\"}]}",
  "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	<span style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">subscribe</span> <span style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">unsubscribe</span>
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
> instId	String	No	Index, e.g. <span style="border: 1px solid #ccc; padding: 2px; border-radius: 5px;">BTC-USD</span>
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

### Push Data Example

```
{
  "arg": {
    "channel": "index-candle30m",
    "instId": "BTC-USD"
  }
}
```

```

"instId": "BTC-USD"
},
"data": [[ "1597026383085", "3811.31", "3811.31", "3811.31", "3811.31", "0"]]
}

```

## PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> instId	String	Index
data	Array of Arrays	Subscribed data
> ts	String	Opening time of the candlestick, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> o	String	Open price
> h	String	Highest price
> l	String	Lowest price
> c	String	Close price
> confirm	String	The state of candlesticks. 0 represents that it is uncompleted, 1 represents that it is completed.

The order of the returned values is: [ts,o,h,l,c,confirm]

## LIQUIDATION ORDERS CHANNEL

Retrieve the recent liquidation orders. For futures and swaps, each contract will only show a maximum of one order per one-second period. This data doesn't represent the total number of liquidations on OKX.

### URL PATH

/ws/v5/public

### Request Example

```

import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/public")
    await ws.start()
    args = [
        {
            "channel": "liquidation-orders",
            "instType": "SWAP"
        }
    ]

    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <a href="#">subscribe</a> <a href="#">unsubscribe</a>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <a href="#">liquidation-orders</a>
> instType	String	Yes	Instrument type <a href="#">SWAP</a> <a href="#">FUTURES</a> <a href="#">MARGIN</a> <a href="#">OPTION</a>

## Response Example

```
{
  "id": "1512",
  "arg": {
    "channel": "liquidation-orders",
    "instType": "SWAP"
  },
  "data": [
    {
      "details": [
        {
          "bkLoss": "0",
          "bkPx": "0.007831",
          "ccy": "",
          "posSide": "short",
          "side": "buy",
          "sz": "13",
          "ts": "1692266434010"
        }
      ],
      "instFamily": "IOST-USDT",
      "instId": "IOT-USDT-SWAP",
      "instType": "SWAP",
      "uly": "IOT-USDT"
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
id	String	Unique identifier of the message
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> instId	String	Instrument ID

Parameter	Type	Description
data	Array of objects	Subscribed data
> instType	String	Instrument type
> instId	String	Instrument ID, e.g. <code>BTC-USD-SWAP</code>
> uly	String	Underlying Applicable to <code>FUTURES</code> / <code>SWAP</code> / <code>OPTION</code>
> details	Array of objects	Liquidation details
>> side	String	Order side <code>buy</code> <code>sell</code> Applicable to <code>FUTURES</code> / <code>SWAP</code>
>> posSide	String	Position mode side <code>long</code> : Hedge mode long <code>short</code> : Hedge mode short <code>net</code> : Net mode
>> bkPx	String	Bankruptcy price. The price of the transaction with the system's liquidation account, only applicable to <code>FUTURES</code> / <code>SWAP</code>
>> sz	String	Quantity of liquidation, only applicable to <code>MARGIN</code> / <code>FUTURES</code> / <code>SWAP</code> . For <code>MARGIN</code> , the unit is base currency. For <code>FUTURES/SWAP</code> , the unit is contract.
>> bkLoss	String	Bankruptcy loss
>> ccy	String	Liquidation currency, only applicable to <code>MARGIN</code>
>> ts	String	Liquidation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

#### ADL WARNING CHANNEL

Auto-deleveraging warning channel.

In the `normal` state, data will be pushed once every minute to display the balance of security fund and etc.

In the warning state or when there is ADL risk (`warning/ad1`), data will be pushed every second to display information such as the real-time decline rate of security fund.

For more ADL details, please refer to [Introduction to Auto-deleveraging](#)

#### URL PATH

`/ws/v5/public`

#### Request Example

```
import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/public")
    await ws.start()
    args = [
        "channel": "adl-warning",
        "instType": "FUTURES",
        "instFamily": "BTC-USDT"
    ]
```

```

}]

await ws.subscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

await ws.unsubscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>ad1-warning</code>
> instType	String	Yes	Instrument type <code>SWAP</code> <code>FUTURES</code> <code>OPTION</code>
> instFamily	String	No	Instrument family

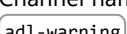
## Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "ad1-warning",
    "instType": "FUTURES",
    "instFamily": "BTC-USDT"
  },
  "connId": "48d8960a"
}
```

## Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "msg": "Illegal request: { \"event\": \"subscribe\", \"arg\": { \"channel\": \"ad1-warning\", \"instType\": \"FUTURES\", \"instFamily\": \"BTC-USDT\" } }",
  "code": "60012",
  "connId": "48d8960a"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event   
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name 
> instType	String	Yes	Instrument type
> instFamily	String	No	Instrument family
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

### Push Data Example

```
{
  "arg": {
    "channel": "adl-warning",
    "instType": "FUTURES",
    "instFamily": "BTC-USDT"
  },
  "data": [
    {
      "maxBal": "",
      "adlRecBal": "8000.0",
      "bal": "280784384.9564228289548144",
      "instType": "FUTURES",
      "ccy": "USDT",
      "instFamily": "BTC-USDT",
      "maxBalTs": "",
      "adlType": "",
      "state": "normal",
      "adlBal": "0",
      "ts": "1700210763001"
    }
  ]
}
```

### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Subscribed channel
> channel	String	Channel name 
> instType	String	Instrument type
> instFamily	String	Instrument family
data	Array of objects	Subscribed data
> instType	String	Instrument type

Parameter	Type	Description
> instFamily	String	Instrument family
> state	String	<p>state</p> <p><code>normal</code></p> <p><code>warning</code></p> <p><code>adl</code></p>
> bal	String	Real-time security fund balance
> ccy	String	The corresponding currency of security fund balance
> maxBal	String	<p>Maximum security fund balance in the past eight hours</p> <p>Applicable when state is <code>warning</code> or <code>adl</code></p>
> maxBalTs	String	Timestamp when security fund balance reached maximum in the past eight hours, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> adlType	String	<p>ADL related events</p> <p><code>rate_adl_start</code>: ADL begins due to high security fund decline rate</p> <p><code>bal_adl_start</code>: ADL begins due to security fund balance falling</p> <p><code>pos_adl_start</code>: ADL begins due to the volume of liquidation orders falls to a certain level (only applicable to premarket symbols)</p> <p><code>adl_end</code>: ADL ends</p>
> adlBal	String	security fund balance that triggers ADL
> adlRecBal	String	security fund balance that turns off ADL
> ts	String	Data push time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> decRate	String	<p>Real-time security fund decline rate (compare bal and maxBal)</p> <p>Applicable when state is <code>warning</code> or <code>adl</code> (Deprecated)</p>
> adlRate	String	<del>security fund decline rate that triggers ADL</del> (Deprecated)
> adlRecRate	String	<del>security fund decline rate that turns off ADL</del> (Deprecated)

## ECONOMIC CALENDAR CHANNEL

This endpoint is only supported in production environment.

Retrieve the most up-to-date economic calendar data. This endpoint is only applicable to VIP 1 and above users in the trading fee tier.

### URL PATH

/ws/v5/business (required login)

Request Example

```

import asyncio
from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
    )

```

```

url = "wss://ws.okx.com:8443/ws/v5/business",
useServerTime=False
)
await ws.start()
args = [
{
  "channel": "economic-calendar"
}
]

await ws.subscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

await ws.unsubscribe(args, callback=callbackFunc)
await asyncio.sleep(10)

asyncio.run(main())

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>economic-calendar</code>

## Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "economic-calendar"
  },
  "connId": "a4d3ae55"
}
```

## Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \\"subscribe\\\", \\"args\\": [{ \\"channel\\\" : \\"economic-calendar\\\", \\"instId\\\" : \\"LTC-USD-190628\\\" }]}",
  "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Event <code>subscribe</code>

Parameter	Type	Required	Description
			<div style="display: flex; justify-content: space-around; align-items: center;"> <span>unsubscribe</span> <span>error</span> </div>
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

### Push Data Example

```
{
  "arg": {
    "channel": "economic-calendar"
  },
  "data": [
    {
      "calendarId": "319275",
      "date": "1597026383085",
      "region": "United States",
      "category": "Manufacturing PMI",
      "event": "S&P Global Manufacturing PMI Final",
      "refDate": "1597026383085",
      "actual": "49.2",
      "previous": "47.3",
      "forecast": "49.3",
      "importance": "2",
      "prevInitial": "",
      "ccy": "",
      "unit": "",
      "ts": "1698648096590"
    }
  ]
}
```

### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">economic-calendar</div>
data	Array of objects	Subscribed data
> event	string	Event name
> region	string	Country, region or entity
> category	string	Category name
> actual	string	The actual value of this event
> previous	string	Latest actual value of the previous period The value will be revised if revision is applicable
> forecast	string	Average forecast among a representative group of economists
> prevInitial	string	The initial value of the previous period Only applicable when revision happens

Parameter	Type	Description
> date	string	Estimated release time of the value of actual field, millisecond format of Unix timestamp, e.g. <code>1597026383085</code>
> refDate	string	Date for which the datapoint refers to
> calendarId	string	Calendar ID
> unit	string	Unit of the data
> ccy	string	Currency of the data
		Level of importance ①: low ②: medium ③: high
> importance	string	
> ts	string	The time of the latest update

# Trading Statistics

## REST API

The API endpoints of `Trading Statistics` do not require authentication.

### GET SUPPORT COIN

Retrieve the currencies supported by the trading statistics endpoints.

#### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: IP

#### HTTP REQUEST

`GET /api/v5/rubik/stat/trading-data/support-coin`

#### Request Example

```
import okx.TradingData as TradingData_api

flag = "0" # Production trading:0 , demo trading:1

tradingDataAPI = TradingData_api.TradingDataAPI(flag=flag)

# Retrieve the currencies supported by the trading statistics endpoints
result = tradingDataAPI.get_support_coin()
print(result)
```

#### Response Example

```
{
  "code": "0",
  "data": {
    "contract": [
      "ADA",
      "BTC"
    ],
    "option": [
      "BTC"
    ],
    "spot": [
      "ADA",
      "BTC"
    ]
  }
}
```

```
        },  
        "msg": "  
    }  
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
contract	Array of strings	Currency supported by derivatives trading data
option	Array of strings	Currency supported by option trading data
spot	Array of strings	Currency supported by spot trading data

[GET CONTRACT OPEN INTEREST HISTORY](#)

Retrieve the contract open interest statistics of futures and perp. This endpoint can retrieve the latest 1,440 data entries.

For period=1D, the data time range is up to January 1, 2024; for other periods, the data time range is up to early February 2024.

**RATE LIMIT: 10 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP + INSTRUMENT ID**

## HTTP REQUEST

GET /api/v5/rubik/stat/contracts/open-interest-history

## Request example

```
import okx.TradingData as TradingData_api

flag = "0" # Production trading:0 , demo trading:1

tradingDataAPI = TradingData_api.TradingDataAPI(flag=flag)

# Retrieve the open interest history
result = tradingDataAPI.get_open_interest_history(
    instId="BTC-USDT-SWAP"
)

print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	string	Yes	Instrument ID, eg: BTC-USDT-SWAP Only applicable to <b>FUTURES</b> , <b>SWAP</b>
period	string	No	Bar size, the default is <b>5m</b> , e.g. <b>[5m/15m/30m/1H/2H/4H]</b> UTC+8 opening price k-line: <b>[6H/12H/1D/2D/3D/5D/1W/1M/3M]</b> UTC+0 opening price k-line: <b>[6Hutc/12Hutc/1Dutc/2Dutc/3Dutc/5Dutc/1Wutc/1Mutc/3Mutc]</b>
end	string	No	Pagination of data to return records earlier than the requested <b>ts</b>
begin	string	No	return records newer than the requested <b>ts</b>
limit	string	No	Number of results per request. The maximum is <b>100</b> . The default is <b>100</b> .

### Response example

```
{  
  "code": "0",  
  "msg": "",  
  "data": [  
    {  
      "id": "1701417600000",  
      "time": "2017-01-17T14:17:00.000Z",  
      "value": 176000000  
    }  
  ]  
}
```

```

    "731377.57500501", // open interest (oi, contracts)
    "111",           // open interest (oiCcy, coin)
    "8888888"       // open interest (oiUsd, USD)
  ],
  [
    "1701417500000", // timestamp
    "731377.57500501", // open interest (oi, contracts)
    "111",           // open interest (oiCcy, coin)
    "8888888"       // open interest (oiUsd, USD)
  ]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Timestamp, millisecond format of Unix timestamp, e.g. <code>1597026383085</code>
oi	String	Open interest in the unit of contracts
oiCcy	String	Open interest in the unit of crypto
oiUsd	String	Open interest in the unit of USD

The data returned will be arranged in an array like this: [ts, oi, oiCcy, oiUsd].

## GET TAKER VOLUME

Retrieve the taker volume for both buyers and sellers.

### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: IP

### HTTP REQUEST

```
GET /api/v5/rubik/stat/taker-volume
```

#### Request Example

```

import okx.TradingData as TradingData_api

flag = "0" # Production trading:0 , demo trading:1

tradingDataAPI = TradingData_api.TradingDataAPI(flag=flag)

# Retrieve the taker volume for both buyers and sellers
result = tradingDataAPI.get_taker_volume(
    ccy="BTC",
    instType="SPOT"
)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	Yes	Currency
instType	String	Yes	Instrument type <code>SPOT</code> <code>CONTRACTS</code>
begin	String	No	Begin time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
end	String	No	End time, Unix timestamp format in milliseconds, e.g. <code>1597026383011</code>

Parameter	Type	Required	Description
period	String	No	<p>Period, the default is <code>5m</code>, e.g. <code>[5m/1H/1D]</code></p> <p><code>5m</code> granularity can only query data within two days at most</p> <p><code>1H</code> granularity can only query data within 30 days at most</p> <p><code>1D</code> granularity can only query data within 180 days at most</p>

### Response Example

```
{
  "code": "0",
  "data": [
    [
      "1630425600000",
      "7596.2651",
      "7149.4855"
    ],
    [
      "1630339200000",
      "5312.7876",
      "7002.7541"
    ]
  ],
  "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Timestamp
sellVol	String	Sell volume
buyVol	String	Buy volume

The return value array order is: [ts,sellVol,buyVol]

### GET CONTRACT TAKER VOLUME

Retrieve the contract taker volume for both buyers and sellers. This endpoint can retrieve the latest 1,440 data entries.

For period=1D, the data time range is up to January 1, 2024; for other periods, the data time range is up to early February 2024.

#### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: IP + INSTRUMENT ID

#### HTTP REQUEST

```
GET /api/v5/rubik/stat/taker-volume-contract
```

### Request example

```
import okx.TradingData as TradingData_api

flag = "0" # Production trading:0 , demo trading:1

tradingDataAPI = TradingData_api.TradingDataAPI(flag=flag)

# Retrieve the contract taker volume for both buyers and sellers
result = tradingDataAPI.get_contract_taker_volume(
    instId="BTC-USDT-SWAP"
)

print(result)
```

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
instId	string	Yes	Instrument ID, eg: BTC-USDT-SWAP Only applicable to <b>FUTURES</b> , <b>SWAP</b>
period	string	No	Bar size, the default is <b>5m</b> , e.g. <b>5m/15m/30m/1H/2H/4H</b> UTC+8 opening price k-line: <b>[6H/12H/1D/2D/3D/5D/1W/1M/3M]</b> UTC+0 opening price k-line: <b>[6Hutc/12Hutc/1Dutc/2Dutc/3Dutc/5Dutc/1Wutc/1Mutc/3Mutc]</b>
unit	string	No	The unit of buy/sell volume, the default is <b>1</b> <b>0</b> : Crypto <b>1</b> : Contracts <b>2</b> : U
end	string	No	return records earlier than the requested <b>ts</b>
begin	string	No	return records newer than the requested <b>ts</b>
limit	string	No	Number of results per request. The maximum is <b>100</b> . The default is <b>100</b> .

**Response example**

```
{
  "code": "0",
  "msg": "",
  "data": [
    [
      [
        "1701417600000", // timestamp
        "200",           // taker sell volume
        "380"           // taker buy volume
      ],
      [
        "1701417600000", // timestamp
        "100",           // taker sell volume
        "300"           // taker buy volume
      ]
    ]
  ]
}
```

**RESPONSE PARAMETERS**

Parameter	Type	Description
ts	String	Timestamp, millisecond format of Unix timestamp, e.g. <b>1597026383085</b>
sellVol	String	taker sell volume
buyVol	String	taker buy volume

The data returned will be arranged in an array like this: [ts, sellVol, buyVol].

**GET MARGIN LONG/SHORT RATIO**

Retrieve the ratio of cumulative amount of quote currency to base currency.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS****RATE LIMIT RULE: IP****HTTP REQUEST**

```
GET /api/v5/rubik/stat/margin/loan-ratio
```

**Request Example**

```

import okx.TradingData as TradingData_api

flag = "0" # Production trading:0 , demo trading:1

tradingDataAPI = TradingData_api.TradingDataAPI(flag=flag)

# Retrieve the ratio of cumulative amount between currency margin quote currency and base currency
result = tradingDataAPI.get_margin_lending_ratio(
    ccy="BTC",
)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	Yes	Currency
begin	String	No	Begin time, e.g. <code>1597026383085</code>
end	String	No	End time, e.g. <code>1597026383085</code>
period	String	No	Period <code>m</code> : Minute, <code>H</code> : Hour, <code>D</code> : Day the default is <code>5m</code> , e.g. <code>[5m/1H/1D]</code> <code>5m</code> granularity can only query data within two days at most <code>1H</code> granularity can only query data within 30 days at most <code>1D</code> granularity can only query data within 180 days at most

## Response Example

```
{
  "code": "0",
  "data": [
    [
      "1630492800000",
      "0.4614"
    ],
    [
      "1630492500000",
      "0.5767"
    ]
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Timestamp
ratio	String	Margin lending ratio

The return value array order is: [ts,ratio]

## GET TOP TRADERS CONTRACT LONG/SHORT RATIO

Retrieve the account net long/short ratio of a contract for top traders. Top traders refer to the top 5% of traders with the largest open position value. This endpoint can retrieve the latest 1,440 data entries. The data time range is up to March 22, 2024.

### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: IP + INSTRUMENT ID

### HTTP REQUEST

## Request Example

```

import okx.TradingData as TradingData_api

flag = "0" # Production trading:0 , demo trading:1

tradingDataAPI = TradingData_api.TradingDataAPI(flag=flag)

# Retrieve the top trader long short account ratio
result = tradingDataAPI.get_top_trader_long_short_account_ratio(
    instId="BTC-USDT-SWAP"
)

print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	string	Yes	Instrument ID, eg: BTC-USDT-SWAP Only applicable to <code>FUTURES</code> , <code>SWAP</code>
period	string	No	Bar size, the default is <code>5m</code> , e.g. <code>[5m/15m/30m/1H/2H/4H]</code> UTC+8 opening price k-line: <code>[6H/12H/1D/2D/3D/5D/1W/1M/3M]</code> UTC+0 opening price k-line: <code>[6Hutc/12Hutc/1Dutc/2Dutc/3Dutc/5Dutc/1Wutc/1Mutc/3Mutc]</code>
end	string	No	return records earlier than the requested <code>ts</code>
begin	string	No	return records newer than the requested <code>ts</code>
limit	string	No	Number of results per request. The maximum is <code>100</code> . The default is <code>100</code> .

## Response example

```
{
  "code": "0",
  "msg": "",
  "data": [
    [
      "1701417600000", // timestamp
      "1.1739" // long/short account num ratio of top traders
    ],
    [
      "1701417600000", // timestamp
      "0.1236" // long/short account num ratio of top traders
    ],
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Timestamp, millisecond format of Unix timestamp, e.g. <code>1597026383085</code>
longShortAcctRatio	String	Long/short account num ratio of top traders

The data returned will be arranged in an array like this: [ts, longShortAcctRatio].

## GET TOP TRADERS CONTRACT LONG/SHORT RATIO (BY POSITION)

Retrieve the position long/short ratio of a contract for top traders. Top traders refer to the top 5% of traders with the largest open position value. This endpoint can retrieve the latest 1,440 data entries. The data time range is up to March 22, 2024.

## RATE LIMIT: 5 REQUESTS PER 2 SECONDS

## HTTP REQUEST

GET /api/v5/rubik/stat/contracts/long-short-position-ratio-contract-top-trader

## Request example

```

import okx.TradingData as TradingData_api

flag = "0" # Production trading:0 , demo trading:1

tradingDataAPI = TradingData_api.TradingDataAPI(flag=flag)

# Retrieve the top trader long short position ratio
result = tradingDataAPI.get_top_trader_long_short_position_ratio(
    instId="BTC-USDT-SWAP"
)

print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
instId	string	Yes	Instrument ID, e.g. <code>BTC-USDT-SWAP</code> Only applicable to <code>FUTURES</code> / <code>SWAP</code>
period	string	No	Bar size, the default is <code>5m</code> , e.g. <code>[5m/15m/30m/1H/2H/4H]</code> UTC+8 opening price k-line: <code>[6H/12H/1D/2D/3D/5D/1W/1M/3M]</code> UTC+0 opening price k-line: <code>[6Hutc/12Hutc/1Dutc/2Dutc/3Dutc/5Dutc/1Wutc/1Mutc/3Mutc]</code>
end	string	No	return records earlier than the requested <code>ts</code>
begin	string	No	return records newer than the requested <code>ts</code>
limit	string	No	Number of results per request. The maximum is <code>100</code> . The default is <code>100</code> .

## Response example

```
{
  "code": "0",
  "msg": "",
  "data": [
    [
      "1701417600000", // timestamp
      "1.1739" // long/short position num ratio of top traders
    ],
    [
      "1701417600000", // timestamp
      "0.1236" // long/short position num ratio of top traders
    ],
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Timestamp, millisecond format of Unix timestamp, e.g. <code>1597026383085</code>
longShortPosRatio	String	Long/short position ratio of top traders

The data returned will be arranged in an array like this: [ts, longShortPosRatio].

## GET CONTRACT LONG/SHORT RATIO

Retrieve the account long/short ratio of a contract. This endpoint can retrieve the latest 1,440 data entries.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS****RATE LIMIT RULE: IP + INSTRUMENT ID****HTTP REQUEST**

GET /api/v5/rubik/stat/contracts/long-short-account-ratio-contract

## Request example

```
import okx.TradingData as TradingData_api

flag = "0" # Production trading:0 , demo trading:1

tradingDataAPI = TradingData_api.TradingDataAPI(flag=flag)

# Retrieve the account long short ratio of a contract
result = tradingDataAPI.get_contract_long_short_ratio(
    instId="BTC-USDT-SWAP"
)

print(result)
```

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
instId	string	Yes	Instrument ID, eg: BTC-USDT-SWAP Only applicable to <a href="#">FUTURES</a> , <a href="#">SWAP</a>
period	string	No	Bar size, the default is <a href="#">5m</a> , e.g. <a href="#">5m/15m/30m/1H/2H/4H</a> UTC+8 opening price k-line: <a href="#">[6H/12H/1D/2D/3D/5D/1W/1M/3M]</a> UTC+0 opening price k-line: <a href="#">[6Hutc/12Hutc/1Dutc/2Dutc/3Dutc/5Dutc/1Wutc/1Mutc/3Mutc]</a>
end	string	No	return records earlier than the requested <a href="#">ts</a>
begin	string	No	return records newer than the requested <a href="#">ts</a>
limit	string	No	Number of results per request. The maximum is <a href="#">100</a> . The default is <a href="#">100</a> .

## Response example

```
{
  "code": "0",
  "msg": "",
  "data": [
    [
      "1701417600000", // timestamp
      "1.1739" // long/short account num ratio of traders
    ],
    [
      "1701417600000", // timestamp
      "0.1236" // long/short account num ratio of traders
    ],
  ]
}
```

**RESPONSE PARAMETERS**

Parameter	Type	Description
ts	String	Timestamp, millisecond format of Unix timestamp, e.g. <a href="#">1597026383085</a>
longShortAcctRatio	String	Long/short position num ratio of all traders

The data returned will be arranged in an array like this: [ts, longAcctPosRatio].

## GET LONG/SHORT RATIO

Retrieve the ratio of users with net long vs net short positions for Expiry Futures and Perpetual Futures.

### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

### RATE LIMIT RULE: IP

### HTTP REQUEST

```
GET /api/v5/rubik/stat/contracts/long-short-account-ratio
```

#### Request Example

```
import okx.TradingData as TradingData_api

flag = "0" # Production trading:0 , demo trading:1

tradingDataAPI = TradingData_api.TradingDataAPI(flag=flag)

# Retrieve the ratio of users with net long vs net short positions for Expiry Futures and Perpetual Futures
result = tradingDataAPI.get_long_short_ratio(
    ccy="BTC",
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	Yes	Currency
begin	String	No	Begin time, e.g. <code>1597026383085</code>
end	String	No	End time, e.g. <code>1597026383011</code>
period	String	No	Period, the default is <code>5m</code> , e.g. <code>[5m/1H/1D]</code> <code>5m</code> granularity can only query data within two days at most <code>1H</code> granularity can only query data within 30 days at most <code>1D</code> granularity can only query data within 180 days at most

#### Response Example

```
{
  "code": "0",
  "data": [
    [
      "1630502100000",
      "1.25"
    ]
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Timestamp
ratio	String	Long/Short ratio

The return value array order is: [ts,ratio]

## GET CONTRACTS OPEN INTEREST AND VOLUME

Retrieve the open interest and trading volume for Expiry Futures and Perpetual Futures.

## RATE LIMIT RULE: IP

## HTTP REQUEST

GET /api/v5/rubik/stat/contracts/open-interest-volume

## Request Example

```
import okx.TradingData as TradingData_api

flag = "0" # Production trading:0 , demo trading:1

tradingDataAPI = TradingData_api.TradingDataAPI(flag=flag)

# Retrieve the open interest and trading volume for Expiry Futures and Perpetual Futures
result = tradingDataAPI.get_contracts_interest_volume(
    ccy="BTC",
)
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	Yes	Currency
begin	String	No	Begin time, e.g. <code>1597026383085</code>
end	String	No	End time, e.g. <code>1597026383011</code>
period	String	No	Period, the default is <code>5m</code> , e.g. <code>[5m/1H/1D]</code> <code>5m</code> granularity can only query data within two days at most <code>1H</code> granularity can only query data within 30 days at most <code>1D</code> granularity can only query data within 180 days at most

## Response Example

```
{
  "code": "0",
  "data": [
    [
      "1630502400000",
      "1713028741.6898",
      "39800873.554"
    ]
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Timestamp
oi	String	Total open interest (USD)
vol	String	Total trading volume (USD)

The return value array order is: [ts,oi,vol]

## GET OPTIONS OPEN INTEREST AND VOLUME

Retrieve the open interest and trading volume for options.

## RATE LIMIT RULE: IP

## HTTP REQUEST

GET /api/v5/rubik/stat/option/open-interest-volume

## Request Example

```
import okx.TradingData as TradingData_api

flag = "0" # Production trading:0 , demo trading:1

tradingDataAPI = TradingData_api.TradingDataAPI(flag=flag)

# Retrieve the open interest and trading volume for options
result = tradingDataAPI.get_options_interest_volume(
    ccy="BTC",
)
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	Yes	Currency
period	String	No	Period, the default is 8H. e.g. [8H/1D] Each granularity can only query 72 pieces of data at the earliest

## Response Example

```
{
  "code": "0",
  "data": [
    [
      "1630368000000",
      "3458.1000",
      "78.8000"
    ]
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Timestamp
oi	String	Total open interest , unit in (ccy) (in request parameter)
vol	String	Total trading volume , unit in (ccy) (in request parameter)

The return value array order is: [ts,oi,vol]

## GET PUT/CALL RATIO

Retrieve the open interest ratio and trading volume ratio of calls vs puts.

GET /api/v5/rubik/stat/option/open-interest-volume-ratio

## Request Example

```
import okx.TradingData as TradingData_api

flag = "0" # Production trading:0 , demo trading:1

tradingDataAPI = TradingData_api.TradingDataAPI(flag=flag)

# Retrieve the open interest ratio and trading volume ratio of calls vs puts
result = tradingDataAPI.get_put_call_ratio(
    ccy="BTC",
)
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	Yes	Currency
period	String	No	Period, the default is 8H. e.g. [8H/1D] Each granularity can only query 72 pieces of data at the earliest

## Response Example

```
{
    "code": "0",
    "data": [
        [
            "1630512000000",
            "2.7261",
            "2.3447"
        ],
        [
            "1630425600000",
            "2.8101",
            "2.3438"
        ]
    ],
    "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Timestamp of data generation time
oiRatio	String	Long/Short open interest ratio
volRatio	String	Long/Short trading volume ratio

The return value array order is: [ts,oiRatio,volRatio]

### GET OPEN INTEREST AND VOLUME (EXPIRY)

Retrieve the open interest and trading volume of calls and puts for each upcoming expiration.

#### RATE LIMIT: 5 REQUESTS PER 2 SECONDS

#### RATE LIMIT RULE: IP

#### HTTP REQUEST

```
GET /api/v5/rubik/stat/option/open-interest-volume-expiry
```

## Request Example

```

import okx.TradingData as TradingData_api

flag = "0" # Production trading:0 , demo trading:1

tradingDataAPI = TradingData_api.TradingDataAPI(flag=flag)

# Retrieve the open interest and trading volume of calls and puts for each upcoming expiration
result = tradingDataAPI.get_interest_volume_expiry(
    ccy="BTC"
)
print(result)

```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	Yes	Currency
period	String	No	Period, the default is 8H. e.g. [8H/1D] Each granularity can provide only one latest piece of data

#### Response Example

```
{
    "code": "0",
    "data": [
        [
            "1630540800000",
            "20210902",
            "6.4",
            "18.4",
            "0.7",
            "0.4"
        ],
        [
            "1630540800000",
            "20210903",
            "47",
            "36.6",
            "1",
            "10.7"
        ]
    ],
    "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Timestamp
expTime	String	Contract expiry date, the format is YYYYMMDD, e.g. 20210623
callOI	String	Total call open interest (coin as the unit)
putOI	String	Total put open interest (coin as the unit)
callVol	String	Total call trading volume (coin as the unit)
putVol	String	Total put trading volume (coin as the unit)

The return value array order is: [ts,expTime,callOI,putOI,callVol,putVol]

Retrieve the taker volume for both buyers and sellers of calls and puts.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

**HTTP REQUEST**

```
GET /api/v5/rubik/stat/option/open-interest-volume-strike
```

Request Example

```
import okx.TradingData as TradingData_api

flag = "0" # Production trading:0 , demo trading:1

tradingDataAPI = TradingData_api.TradingDataAPI(flag=flag)

# Retrieve the taker volume for both buyers and sellers of calls and puts
result = tradingDataAPI.get_interest_volume_strike(
    ccy="BTC",
    expTime="20210623"
)
print(result)
```

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
ccy	String	Yes	Currency
expTime	String	Yes	Contract expiry date, the format is <code>YYYYMMdd</code> , e.g. <code>20210623</code>
period	String	No	Period, the default is <code>8H</code> . e.g. <code>[8H/1D]</code> Each granularity can provide only one latest piece of data

Response Example

```
{
  "code": "0",
  "data": [
    [
      "1630540800000",
      "10000",
      "0",
      "0.5",
      "0",
      "0",
      "0"
    ],
    [
      "1630540800000",
      "14000",
      "0",
      "5.2",
      "0",
      "0"
    ]
  ],
  "msg": ""
}
```

**RESPONSE PARAMETERS**

Parameter	Type	Description
ts	String	Timestamp
strike	String	Strike price
callOI	String	Total call open interest ( <code>coin</code> as the unit)

Parameter	Type	Description
putOI	String	Total put open interest ([coin] as the unit)
callVol	String	Total call trading volume ([coin] as the unit)
putVol	String	Total put trading volume ([coin] as the unit)

The return value array order is: [ts,strike,callOI,putOI,callVol,putVol]

#### GET TAKER FLOW

This shows the relative buy/sell volume for calls and puts. It shows whether traders are bullish or bearish on price and volatility.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: IP**

#### HTTP REQUEST

GET /api/v5/rubik/stat/option/taker-block-volume

#### Request Example

```
import okx.TradingData as TradingData_api

flag = "0" # Production trading:0 , demo trading:1

tradingDataAPI = TradingData_api.TradingDataAPI(flag=flag)

# This shows the relative buy/sell volume for calls and puts. It shows whether traders are bullish or bearish on price and volatility
result = tradingDataAPI.get_taker_block_volume(
    ccy="BTC",
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	Yes	currency
period	String	No	period, the default is 8H. e.g. [8H/1D] Each granularity can provide only one latest piece of data

#### Response Example

```
{
  "code": "0",
  "data": [
    "163051200000",
    "8.55",
    "67.3",
    "16.05",
    "16.3",
    "126.4",
    "40.7"
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Timestamp

Parameter	Type	Description
callBuyVol	String	call option buy volume, in settlement currency
callSellVol	String	call option sell volume, in settlement currency
putBuyVol	String	put option buy volume, in settlement currency
putSellVol	String	put option sell volume, in settlement currency
callBlockVol	String	call block volume
putBlockVol	String	put block volume

The return value array order is: [ts,callBuyVol,callSellVol,putBuyVol,putSellVol,callBlockVol,putBlockVol]

## Funding Account

The API endpoints of [Funding Account](#) require authentication.

## REST API

### GET CURRENCIES

Retrieve a list of all currencies available which are related to the current account's KYC entity.

**RATE LIMIT: 6 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

### HTTP REQUEST

`GET /api/v5/asset/currencies`

Request Example

```
import okx.Funding as Funding

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading: 0, Demo trading: 1

fundingAPI = Funding.FundingAPI(apikey, secretkey, passphrase, False, flag)

# Get currencies
result = fundingAPI.get_currencies()
print(result)
```

### REQUEST PARAMETERS

Parameters	Types	Required	Description
ccy	String	No	Single currency or multiple currencies separated with comma, e.g. <code>BTC</code> or <code>BTC,ETH</code> .

## Response Example

```
{  
  "code": "0",  
  "msg": "",  
  "data": [  
    {  
      "burningFeeRate": "",  
      "canDep": true,  
      "canInternal": true,  
      "canWd": true,  
      "ccy": "BTC",  
      "chain": "BTC-Bitcoin",  
      "ctAddr": "",  
      "depEstOpenTime": "",  
      "depQuotaFixed": "",  
      "depQuoteDailyLayer2": "",  
      "fee": "0.0005",  
      "logoLink": "https://static.coinall.ltd/cdn/oksupport/asset/currency/icon/btc20230419112752.png",  
      "mainNet": true,  
      "maxFee": "0.0005",  
      "maxFeeForCtAddr": "",  
      "maxWd": "500",  
      "minDep": "0.0005",  
      "minDepArrivalConfirm": "1",  
      "minFee": "0.0005",  
      "minFeeForCtAddr": "",  
      "minInternal": "0.0001",  
      "minWd": "0.0005",  
      "minWdUnlockConfirm": "2",  
      "name": "Bitcoin",  
      "needTag": false,  
      "usedDepQuotaFixed": "",  
      "usedWdQuota": "0",  
      "wdEstOpenTime": "",  
      "wdQuota": "10000000",  
      "wdTickSz": "8"  
    }  
  ]  
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency, e.g. <code>BTC</code>
name	String	Name of currency. There is no related name when it is not shown.
logoLink	String	The logo link of currency
chain	String	Chain name, e.g. <code>USDT-ERC20</code> , <code>USDT-TRC20</code>
ctAddr	String	Contract address
canDep	Boolean	The availability to deposit from chain <code>false</code> : not available <code>true</code> : available
canWd	Boolean	The availability to withdraw to chain <code>false</code> : not available <code>true</code> : available
canInternal	Boolean	The availability to internal transfer <code>false</code> : not available <code>true</code> : available
depEstOpenTime	String	Estimated opening time for deposit, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code> if <code>canDep</code> is <code>true</code> , it returns <code>""</code>

Parameter	Type	Description
wdEstOpenTime	String	Estimated opening time for withdraw, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code> if <code>canWd</code> is <code>true</code> , it returns <code>""</code>
minDep	String	The minimum deposit amount of currency in a single transaction
minWd	String	The minimum <code>on-chain withdrawal</code> amount of currency in a single transaction
minInternal	String	The minimum <code>internal transfer</code> amount of currency in a single transaction No maximum <code>internal transfer</code> limit in a single transaction, subject to the withdrawal limit in the past 24 hours( <code>wdQuota</code> ).
maxWd	String	The maximum amount of currency <code>on-chain withdrawal</code> in a single transaction
wdTickSz	String	The withdrawal precision, indicating the number of digits after the decimal point. The withdrawal fee precision kept the same as withdrawal precision. The accuracy of internal transfer withdrawal is 8 decimal places.
wdQuota	String	The withdrawal limit in the past 24 hours (including <code>on-chain withdrawal</code> and <code>internal transfer</code> ), unit in <code>USD</code>
usedWdQuota	String	The amount of currency withdrawal used in the past 24 hours, unit in <code>USD</code>
fee	String	The fixed withdrawal fee Apply to <code>on-chain withdrawal</code>
minFee	String	<del>The minimum withdrawal fee for normal address</del> Apply to <code>on-chain withdrawal</code> (Deprecated)
maxFee	String	<del>The maximum withdrawal fee for normal address</del> Apply to <code>on-chain withdrawal</code> (Deprecated)
minFeeForCtAddr	String	<del>The minimum withdrawal fee for contract address</del> Apply to <code>on-chain withdrawal</code> (Deprecated)
maxFeeForCtAddr	String	<del>The maximum withdrawal fee for contract address</del> Apply to <code>on-chain withdrawal</code> (Deprecated)
burningFeeRate	String	Burning fee rate, e.g "0.05" represents "5%". Some currencies may charge combustion fees. The burning fee is deducted based on the withdrawal quantity (excluding gas fee) multiplied by the burning fee rate. Apply to <code>on-chain withdrawal</code>
mainNet	Boolean	If current chain is main net, then it will return <code>true</code> , otherwise it will return <code>false</code>
needTag	Boolean	Whether tag/memo information is required for withdrawal, e.g. <code>EOS</code> will return <code>true</code>
minDepArrivalConfirm	String	The minimum number of blockchain confirmations to acknowledge fund deposit. The account is credited after that, but the deposit can not be withdrawn
minWdUnlockConfirm	String	The minimum number of blockchain confirmations required for withdrawal of a deposit
depQuotaFixed	String	The fixed deposit limit, unit in <code>USD</code> Return empty string if there is no deposit limit
usedDepQuotaFixed	String	The used amount of fixed deposit quota, unit in <code>USD</code> Return empty string if there is no deposit limit
depQuoteDailyLayer2	String	The layer2 network daily deposit limit

Only asset information of a currency with a balance greater than 0 will be returned.

#### RATE LIMIT: 6 REQUESTS PER SECOND

#### RATE LIMIT RULE: USER ID

#### PERMISSION: READ

#### HTTP REQUEST

GET /api/v5/asset/balances

Request Example

```
import okx.Funding as Funding

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading: 0, Demo trading: 1

fundingAPI = Funding.FundingAPI(apikey, secretkey, passphrase, False, flag)

# Get balance
result = fundingAPI.get_balances()
print(result)
```

#### REQUEST PARAMETERS

Parameters	Types	Required	Description
ccy	String	No	Single currency or multiple currencies (no more than 20) separated with comma, e.g. <code>BTC</code> or <code>BTC,ETH</code> .

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "availBal": "37.11827078",
      "bal": "37.11827078",
      "ccy": "ETH",
      "frozenBal": "0"
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency
bal	String	Balance
frozenBal	String	Frozen balance
availBal	String	Available balance

#### GET NON-TRADEABLE ASSETS

Retrieve the funding account balances of all the assets and the amount that is available or on hold.

## RATE LIMIT RULE: USER ID

## PERMISSION: READ

## HTTP REQUEST

GET /api/v5/asset/non-tradable-assets

## Request Example

```
import okx.Funding as Funding

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

fundingAPI = Funding.FundingAPI(apikey, secretkey, passphrase, False, flag)

result = fundingAPI.get_non_tradable_assets()
print(result)
```

## REQUEST PARAMETERS

Parameters	Types	Required	Description
ccy	String	No	Single currency or multiple currencies (no more than 20) separated with comma, e.g. <code>BTC</code> or <code>BTC,ETH</code> .

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "bal": "989.84719571",
      "burningFeeRate": "",
      "canWd": true,
      "ccy": "CELT",
      "chain": "CELT-OKTC",
      "ctAddr": "f403fb",
      "fee": "2",
      "feeCcy": "USDT",
      "logoLink": "https://static.coinall.ltd/cdn/assets/imgs/221/460DA8A592400393.png",
      "minWd": "0.1",
      "name": "",
      "needTag": false,
      "wdAll": false,
      "wdTickSz": "8"
    },
    {
      "bal": "0.001",
      "burningFeeRate": "",
      "canWd": true,
      "ccy": "MEME",
      "chain": "MEME-ERC20",
      "ctAddr": "09b760",
      "fee": "5",
      "feeCcy": "USDT",
      "logoLink": "https://static.coinall.ltd/cdn/assets/imgs/207/2E664E470103C613.png",
      "minWd": "0.001",
      "name": "MEME Inu",
      "needTag": false,
      "wdAll": false,
      "wdTickSz": "8"
    }
  ],
  "msg": ""
}
```

**RESPONSE PARAMETERS**

Parameter	Type	Description
ccy	String	Currency, e.g. <code>CELT</code>
name	String	Chinese name of currency. There is no related name when it is not shown.
logoLink	String	Logo link of currency
bal	String	Withdrawable balance
canWd	Boolean	Availability to withdraw to chain. <code>false</code> : not available <code>true</code> : available
chain	String	Chain for withdrawal
minWd	String	Minimum withdrawal amount of currency in a single transaction
wdAll	Boolean	Whether all assets in this currency must be withdrawn at one time
fee	String	Fixed withdrawal fee
feeCcy	String	Fixed withdrawal fee unit, e.g. <code>USDT</code>
burningFeeRate	String	Burning fee rate, e.g "0.05" represents "5%". Some currencies may charge combustion fees. The burning fee is deducted based on the withdrawal quantity (excluding gas fee) multiplied by the burning fee rate.
ctAddr	String	Last 6 digits of contract address
wdTickSz	String	Withdrawal precision, indicating the number of digits after the decimal point
needTag	Boolean	Whether tag/memo information is required for withdrawal

**GET ACCOUNT ASSET VALUATION**

View account asset valuation

**RATE LIMIT: 1 REQUEST PER SECOND****RATE LIMIT RULE: USER ID****PERMISSION: READ****HTTP REQUEST**

`GET /api/v5/asset/asset-valuation`

**Request Example**

```
import okx.Funding as Funding

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading: 0, Demo trading: 1

fundingAPI = Funding.FundingAPI(apikey, secretkey, passphrase, False, flag)

# Get account asset valuation
result = fundingAPI.get_asset_valuation()
print(result)
```

**REQUEST PARAMETERS**

Parameters	Types	Required	Description
ccy	String	No	Asset valuation calculation unit BTC, USDT USD, CNY, JP, KRW, RUB, EUR VND, IDR, INR, PHP, THB, TRY AUD, SGD, ARS, SAR, AED, IQD The default is the valuation in BTC.

### Response Example

```
{
  "code": "0",
  "data": [
    {
      "details": {
        "classic": "124.6",
        "earn": "1122.73",
        "funding": "0.09",
        "trading": "2544.28"
      },
      "totalBal": "3790.09",
      "ts": "1637566660769"
    }
  ],
  "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
totalBal	String	Valuation of total account assets
ts	String	Unix timestamp format in milliseconds, e.g. 1597026383085
details	Object	Asset valuation details for each account
> funding	String	Funding account
> trading	String	Trading account
> classic	String	[Deprecated] Classic account
> earn	String	Earn account

### FUNDS TRANSFER

Only API keys with **Trade** privilege can call this endpoint.

This endpoint supports the transfer of funds between your funding account and trading account, and from the master account to sub-accounts.

Sub-account can transfer out to master account by default. Need to call Set permission of transfer out to grant privilege first if you want sub-account transferring to another sub-account (sub-accounts need to belong to same master account.)

The success or failure of the request does not necessarily reflect the actual transfer result. Recommend checking the transfer status by calling "Get funds transfer state" to confirm the final result.

### RATE LIMIT: 2 REQUESTS PER SECOND

### RATE LIMIT RULE: USER ID + CURRENCY

### PERMISSION: TRADE

### HTTP REQUEST

POST /api/v5/asset/transfer

## Request Example

```

import okx.Funding as Funding

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading: 0, Demo trading: 1

fundingAPI = Funding.FundingAPI(apikey, secretkey, passphrase, False, flag)

# Funds transfer
result = fundingAPI.funds_transfer(
    ccy="USDT",
    amt="1.5",
    from_= "6",
    to="18"
)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
type	String	No	<p>Transfer type</p> <p>0: transfer within account</p> <p>1: master account to sub-account (Only applicable to API Key from master account)</p> <p>2: sub-account to master account (Only applicable to API Key from master account)</p> <p>3: sub-account to master account (Only applicable to APIKey from sub-account)</p> <p>4: sub-account to sub-account (Only applicable to APIKey from sub-account, and target account needs to be another sub-account which belongs to same master account. Sub-account directly transfer out permission is disabled by default, set permission please refer to Set permission of transfer out) The default is 0.</p> <p>If you want to make transfer between sub-accounts by master account API key, refer to Master accounts manage the transfers between sub-accounts</p>
ccy	String	Yes	Transfer currency, e.g. <code>USDT</code>
amt	String	Yes	Amount to be transferred
from	String	Yes	<p>The remitting account</p> <p>6: Funding account</p> <p>18: Trading account</p>
to	String	Yes	<p>The beneficiary account</p> <p>6: Funding account</p> <p>18: Trading account</p>
subAcct	String	Conditional	<p>Name of the sub-account</p> <p>When <code>type</code> is 1/2/4, this parameter is required.</p>
loanTrans	Boolean	No	<p>Whether or not borrowed coins can be transferred out under <code>Spot mode</code>/<code>Multi-currency margin</code>/<code>Portfolio margin</code></p> <p><code>true</code>: borrowed coins can be transferred out</p> <p><code>false</code>: borrowed coins cannot be transferred out</p> <p>the default is <code>false</code></p>
omitPosRisk	String	No	<p>Ignore position risk</p> <p>Default is <code>false</code></p> <p>Applicable to <code>Portfolio margin</code></p>
clientId	String	No	<p>Client-supplied ID</p> <p>A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.</p>

## Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "transId": "754147",
      "ccy": "USDT",
      "clientId": "",
      "from": "6",
      "amt": "0.1",
      "to": "18"
    }
  ]
}
```

## RESPONSE PARAMETERS

### Response Example

Parameter	Type	Description
transId	String	Transfer ID
clientId	String	Client-supplied ID
ccy	String	Currency
from	String	The remitting account
amt	String	Transfer amount
to	String	The beneficiary account

## GET FUNDS TRANSFER STATE

Retrieve the transfer state data of the last 2 weeks.

**RATE LIMIT: 10 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

## HTTP REQUEST

```
GET /api/v5/asset/transfer-state
```

### Request Example

```
import okx.Funding as Funding

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

fundingAPI = Funding.FundingAPI(apikey, secretkey, passphrase, False, flag)

# Get funds transfer state
result = fundingAPI.transfer_state(
    transId="248424899",
    type="0"
)
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
transId	String	Conditional	Transfer ID Either transId or clientId is required. If both are passed, transId will be used.
clientId	String	Conditional	Client-supplied ID A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
type	String	No	<p>Transfer type</p> <p>0: transfer within account</p> <p>1: master account to sub-account (Only applicable to API Key from master account)</p> <p>2: sub-account to master account (Only applicable to API Key from master account)</p> <p>3: sub-account to master account (Only applicable to APIKey from sub-account)</p> <p>4: sub-account to sub-account (Only applicable to APIKey from sub-account, and target account needs to be another sub-account which belongs to same master account)</p> <p>The default is 0.</p> <p>For Custody accounts, can choose not to pass this parameter or pass 0.</p>

### Response Example

```
{
  "code": "0",
  "data": [
    {
      "amt": "1.5",
      "ccy": "USDT",
      "clientId": "",
      "from": "18",
      "instId": "", //deprecated
      "state": "success",
      "subAcct": "test",
      "to": "6",
      "toInstId": "", //deprecated
      "transId": "1",
      "type": "1"
    }
  ],
  "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
transId	String	Transfer ID
clientId	String	Client-supplied ID
ccy	String	Currency, e.g. USDT
amt	String	Amount to be transferred
type	String	<p>Transfer type</p> <p>0: transfer within account</p> <p>1: master account to sub-account (Only applicable to API Key from master account)</p> <p>2: sub-account to master account (Only applicable to APIKey from master account)</p> <p>3: sub-account to master account (Only applicable to APIKey from sub-account)</p> <p>4: sub-account to sub-account (Only applicable to APIKey from sub-account, and target account needs to be another sub-account which belongs to same master account)</p>
from	String	<p>The remitting account</p> <p>6: Funding account</p> <p>18: Trading account</p>

Parameter	Type	Description
to	String	The beneficiary account ⑥: Funding account ⑯: Trading account
subAcct	String	Name of the sub-account
instId	String	deprecated
tolInstId	String	deprecated
state	String	Transfer state success pending failed

## ASSET BILLS DETAILS

Query the billing record in the past month.

**RATE LIMIT: 6 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/asset/bills`

Request Example

```
import okx.Funding as Funding

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading: 0, Demo trading: 1

fundingAPI = Funding.FundingAPI(apikey, secretkey, passphrase, False, flag)

# Get asset bills details
result = fundingAPI.get_bills()
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	No	Currency
type	String	No	Bill type ①: Deposit ②: Withdrawal ⑯: Canceled withdrawal ⑯: Transfer to sub account (for master account) ⑯: Transfer from sub account (for master account) ⑯: Transfer out from sub to master account (for sub-account) ⑯: Transfer in from master to sub account (for sub-account) ⑯: Manually claimed Airdrop ⑯: System reversal ⑯: Event Reward ⑯: Event Giveaway ⑯: Fee rebate (by rebate card) ⑯: Token received

Parameter	Type	Required	Description
			73: Token given away
			74: Token refunded
			75: [Simple earn flexible] Subscription
			76: [Simple earn flexible] Redemption
			77: Jumpstart distribute
			78: Jumpstart lock up
			80: DEFI/Staking subscription
			82: DEFI/Staking redemption
			83: Staking yield
			84: Violation fee
			89: Deposit yield
			116: [Fiat] Place an order
			117: [Fiat] Fulfill an order
			118: [Fiat] Cancel an order
			124: Jumpstart unlocking
			130: Transferred from Trading account
			131: Transferred to Trading account
			132: [P2P] Frozen by customer service
			133: [P2P] Unfrozen by customer service
			134: [P2P] Transferred by customer service
			135: Cross chain exchange
			137: [ETH Staking] Subscription
			138: [ETH Staking] Swapping
			139: [ETH Staking] Earnings
			146: Customer feedback
			150: Affiliate commission
			151: Referral reward
			152: Broker reward
			160: Dual Investment subscribe
			161: Dual Investment collection
			162: Dual Investment profit
			163: Dual Investment refund
			172: [Affiliate] Sub-affiliate commission
			173: [Affiliate] Fee rebate (by trading fee)
			174: Jumpstart Pay
			175: Locked collateral
			176: Loan
			177: Added collateral
			178: Returned collateral
			179: Repayment
			180: Unlocked collateral
			181: Airdrop payment
			185: [Broker] Convert reward
			187: [Broker] Convert transfer
			189: Mystery box bonus
			195: Untradable asset withdrawal
			196: Untradable asset withdrawal revoked
			197: Untradable asset deposit
			198: Untradable asset collection reduce
			199: Untradable asset collection increase
			200: Buy
			202: Price Lock Subscribe
			203: Price Lock Collection
			204: Price Lock Profit
			205: Price Lock Refund
			207: Dual Investment Lite Subscribe
			208: Dual Investment Lite Collection
			209: Dual Investment Lite Profit
			210: Dual Investment Lite Refund
			212: [Flexible loan] Multi-collateral loan collateral locked
			215: [Flexible loan] Multi-collateral loan collateral released
			217: [Flexible loan] Multi-collateral loan borrowed

Parameter	Type	Required	Description
			218: [Flexible loan] Multi-collateral loan repaid
			232: [Flexible loan] Subsidized interest received
			220: Delisted crypto
			221: Blockchain's withdrawal fee
			222: Withdrawal fee refund
			223: SWAP lead trading profit share
			225: Shark Fin subscribe
			226: Shark Fin collection
			227: Shark Fin profit
			228: Shark Fin refund
			229: Airdrop
			232: Subsidized interest received
			233: Broker rebate compensation
			240: Snowball subscribe
			241: Snowball refund
			242: Snowball profit
			243: Snowball trading failed
			249: Seagull subscribe
			250: Seagull collection
			251: Seagull profit
			252: Seagull refund
			263: Strategy bots profit share
			265: Signal revenue
			266: SPOT lead trading profit share
			270: DCD broker transfer
			271: DCD broker rebate
			272: [Convert] Buy Crypto/Fiat
			273: [Convert] Sell Crypto/Fiat
			284: [Custody] Transfer out trading sub-account
			285: [Custody] Transfer in trading sub-account
			286: [Custody] Transfer out custody funding account
			287: [Custody] Transfer in custody funding account
			288: [Custody] Fund delegation
			289: [Custody] Fund undelegation
			299: Affiliate recommendation commission
			300: Fee discount rebate
			303: Snowball market maker transfer
			304: [Simple Earn Fixed] Order submission
			305: [Simple Earn Fixed] Order redemption
			306: [Simple Earn Fixed] Principal distribution
			307: [Simple Earn Fixed] Interest distribution (early termination compensation)
			308: [Simple Earn Fixed] Interest distribution
			309: [Simple Earn Fixed] Interest distribution (extension compensation)
			311: Crypto dust auto-transfer in
			313: Sent by gift
			314: Received from gift
			315: Refunded from gift
			328: [SOL staking] Send Liquidity Staking Token reward
			329: [SOL staking] Subscribe Liquidity Staking Token staking
			330: [SOL staking] Mint Liquidity Staking Token
			331: [SOL staking] Redeem Liquidity Staking Token order
			332: [SOL staking] Settle Liquidity Staking Token order
			333: Trial fund reward
			339: [Simple Earn Fixed] Order submission
			340: [Simple Earn Fixed] Order failure refund
			341: [Simple Earn Fixed] Redemption
			342: [Simple Earn Fixed] Principal
			343: [Simple Earn Fixed] Interest
			344: [Simple Earn Fixed] Compensatory interest
			345: [Institutional Loan] Principal repayment
			346: [Institutional Loan] Interest repayment
			347: [Institutional Loan] Overdue penalty

Parameter	Type	Required	Description
			<ul style="list-style-type: none"> <li><code>348</code>: [BTC staking] Subscription</li> <li><code>349</code>: [BTC staking] Redemption</li> <li><code>350</code>: [BTC staking] Earnings</li> <li><code>351</code>: [Institutional Loan] Loan disbursement</li> <li><code>354</code>: Copy and bot rewards</li> <li><code>361</code>: Deposit from closed sub-account</li> <li><code>372</code>: Asset segregation</li> <li><code>373</code>: Asset release</li> <li><code>400</code>: Auto earn interest (USDG earn)</li> </ul>
clientId	String	No	Client-supplied ID for transfer or withdrawal A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
after	String	No	Pagination of data to return records earlier than the requested <code>ts</code> , Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
before	String	No	Pagination of data to return records newer than the requested <code>ts</code> , Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. The maximum is <code>100</code> . The default is <code>100</code> .

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "billId": "12344",
      "ccy": "BTC",
      "clientId": "",
      "balChg": "2",
      "bal": "12",
      "type": "1",
      "ts": "1597026383085",
      "notes": ""
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
billId	String	Bill ID
ccy	String	Account balance currency
clientId	String	Client-supplied ID for transfer or withdrawal
balChg	String	Change in balance at the account level
bal	String	Balance at the account level
type	String	Bill type
notes	String	Notes
ts	String	Creation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

#### ASSET BILLS HISTORY

Query the billing records of all time since 1 February, 2021.

**⚠️ IMPORTANT:** Data updates occur every 30 seconds. Update frequency may vary based on data volume - please be aware of potential delays during high-traffic periods.

## RATE LIMIT: 1 REQUESTS PER SECOND

### RATE LIMIT RULE: USER ID

### PERMISSION: READ

### HTTP REQUEST

```
GET /api/v5/asset/bills-history
```

#### Request Example

```
import okx.Funding as Funding

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading: 0, Demo trading: 1

fundingAPI = Funding.FundingAPI(apikey, secretkey, passphrase, False, flag)

# Get asset bills details
result = fundingAPI.get_bills_history()
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	No	Currency
type	String	No	Bill type 1: Deposit 2: Withdrawal 13: Canceled withdrawal 20: Transfer to sub account (for master account) 21: Transfer from sub account (for master account) 22: Transfer out from sub to master account (for sub-account) 23: Transfer in from master to sub account (for sub-account) 28: Manually claimed Airdrop 47: System reversal 48: Event Reward 49: Event Giveaway 68: Fee rebate (by rebate card) 72: Token received 73: Token given away 74: Token refunded 75: [Simple earn flexible] Subscription 76: [Simple earn flexible] Redemption 77: Jumpstart distribute 78: Jumpstart lock up 80: DEFI/Staking subscription 82: DEFI/Staking redemption 83: Staking yield 84: Violation fee 89: Deposit yield 116: [Fiat] Place an order 117: [Fiat] Fulfill an order 118: [Fiat] Cancel an order 124: Jumpstart unlocking 130: Transferred from Trading account

Parameter	Type	Required	Description
131			Transferred to Trading account
132			[P2P] Frozen by customer service
133			[P2P] Unfrozen by customer service
134			[P2P] Transferred by customer service
135			Cross chain exchange
137			[ETH Staking] Subscription
138			[ETH Staking] Swapping
139			[ETH Staking] Earnings
146			Customer feedback
150			Affiliate commission
151			Referral reward
152			Broker reward
160			Dual Investment subscribe
161			Dual Investment collection
162			Dual Investment profit
163			Dual Investment refund
172			[Affiliate] Sub-affiliate commission
173			[Affiliate] Fee rebate (by trading fee)
174			Jumpstart Pay
175			Locked collateral
176			Loan
177			Added collateral
178			Returned collateral
179			Repayment
180			Unlocked collateral
181			Airdrop payment
185			[Broker] Convert reward
187			[Broker] Convert transfer
189			Mystery box bonus
195			Untradable asset withdrawal
196			Untradable asset withdrawal revoked
197			Untradable asset deposit
198			Untradable asset collection reduce
199			Untradable asset collection increase
200			Buy
202			Price Lock Subscribe
203			Price Lock Collection
204			Price Lock Profit
205			Price Lock Refund
207			Dual Investment Lite Subscribe
208			Dual Investment Lite Collection
209			Dual Investment Lite Profit
210			Dual Investment Lite Refund
212			[Flexible loan] Multi-collateral loan collateral locked
215			[Flexible loan] Multi-collateral loan collateral released
217			[Flexible loan] Multi-collateral loan borrowed
218			[Flexible loan] Multi-collateral loan repaid
232			[Flexible loan] Subsidized interest received
220			Delisted crypto
221			Blockchain's withdrawal fee
222			Withdrawal fee refund
223			SWAP lead trading profit share
225			Shark Fin subscribe
226			Shark Fin collection
227			Shark Fin profit
228			Shark Fin refund
229			Airdrop
232			Subsidized interest received
233			Broker rebate compensation
240			Snowball subscribe
241			Snowball refund
242			Snowball profit

Parameter	Type	Required	Description
			<p>243: Snowball trading failed</p> <p>249: Seagull subscribe</p> <p>250: Seagull collection</p> <p>251: Seagull profit</p> <p>252: Seagull refund</p> <p>263: Strategy bots profit share</p> <p>265: Signal revenue</p> <p>266: SPOT lead trading profit share</p> <p>270: DCD broker transfer</p> <p>271: DCD broker rebate</p> <p>272: [Convert] Buy Crypto/Fiat</p> <p>273: [Convert] Sell Crypto/Fiat</p> <p>284: [Custody] Transfer out trading sub-account</p> <p>285: [Custody] Transfer in trading sub-account</p> <p>286: [Custody] Transfer out custody funding account</p> <p>287: [Custody] Transfer in custody funding account</p> <p>288: [Custody] Fund delegation</p> <p>289: [Custody] Fund undelegation</p> <p>299: Affiliate recommendation commission</p> <p>300: Fee discount rebate</p> <p>303: Snowball market maker transfer</p> <p>304: [Simple Earn Fixed] Order submission</p> <p>305: [Simple Earn Fixed] Order redemption</p> <p>306: [Simple Earn Fixed] Principal distribution</p> <p>307: [Simple Earn Fixed] Interest distribution (early termination compensation)</p> <p>308: [Simple Earn Fixed] Interest distribution</p> <p>309: [Simple Earn Fixed] Interest distribution (extension compensation)</p> <p>311: Crypto dust auto-transfer in</p> <p>313: Sent by gift</p> <p>314: Received from gift</p> <p>315: Refunded from gift</p> <p>328: [SOL staking] Send Liquidity Staking Token reward</p> <p>329: [SOL staking] Subscribe Liquidity Staking Token staking</p> <p>330: [SOL staking] Mint Liquidity Staking Token</p> <p>331: [SOL staking] Redeem Liquidity Staking Token order</p> <p>332: [SOL staking] Settle Liquidity Staking Token order</p> <p>333: Trial fund reward</p> <p>339: [Simple Earn Fixed] Order submission</p> <p>340: [Simple Earn Fixed] Order failure refund</p> <p>341: [Simple Earn Fixed] Redemption</p> <p>342: [Simple Earn Fixed] Principal</p> <p>343: [Simple Earn Fixed] Interest</p> <p>344: [Simple Earn Fixed] Compensatory interest</p> <p>345: [Institutional Loan] Principal repayment</p> <p>346: [Institutional Loan] Interest repayment</p> <p>347: [Institutional Loan] Overdue penalty</p> <p>348: [BTC staking] Subscription</p> <p>349: [BTC staking] Redemption</p> <p>350: [BTC staking] Earnings</p> <p>351: [Institutional Loan] Loan disbursement</p> <p>354: Copy and bot rewards</p> <p>361: Deposit from closed sub-account</p> <p>372: Asset segregation</p> <p>373: Asset release</p> <p>400: Auto earn interest (USDG earn)</p>
clientId	String	No	<p>Client-supplied ID for transfer or withdrawal</p> <p>A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.</p>
after	String	No	<p>Pagination of data to return records earlier than the requested <code>ts</code> or <code>billId</code>, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code></p>

Parameter	Type	Required	Description
before	String	No	Pagination of data to return records newer than the requested <code>ts</code> , Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. The maximum is <code>100</code> . The default is <code>100</code> .
pagingType	String	No	PagingType <code>1</code> : Timestamp of the bill record <code>2</code> : Bill ID of the bill record The default is <code>1</code>

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "billId": "12344",
      "ccy": "BTC",
      "clientId": "",
      "balChg": "2",
      "bal": "12",
      "type": "1",
      "ts": "1597026383085",
      "notes": ""
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
billId	String	Bill ID
ccy	String	Account balance currency
clientId	String	Client-supplied ID for transfer or withdrawal
balChg	String	Change in balance at the account level
bal	String	Balance at the account level
type	String	Bill type
notes	String	Notes
ts	String	Creation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

#### GET DEPOSIT ADDRESS

Retrieve the deposit addresses of currencies, including previously-used addresses.

#### RATE LIMIT: 6 REQUESTS PER SECOND

#### RATE LIMIT RULE: USER ID

#### PERMISSION: READ

#### HTTP REQUEST

```
GET /api/v5/asset/deposit-address
```

#### Request Example

```
import okx.Funding as Funding

# API initialization
```

```

apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading: 0, Demo trading: 1

fundingAPI = Funding.FundingAPI(apikey, secretkey, passphrase, False, flag)

# Get deposit address
result = fundingAPI.get_deposit_address(
    ccy="USDT"
)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	Yes	Currency, e.g. <code>BTC</code>

## Response Example

```

{
  "code": "0",
  "data": [
    {
      "chain": "BTC-Bitcoin",
      "ctAddr": "",
      "ccy": "BTC",
      "to": "6",
      "addr": "39XNxK1Ryqgg3Bsyn6HzoqV4Xji25pNkv6",
      "verifiedName": "John Corner",
      "selected": true
    },
    {
      "chain": "BTC-OKC",
      "ctAddr": "",
      "ccy": "BTC",
      "to": "6",
      "addr": "0x66d0edc2e63b6b992381ee668fbcb01f20ae0428",
      "verifiedName": "John Corner",
      "selected": true
    },
    {
      "chain": "BTC-ERC20",
      "ctAddr": "5807cf",
      "ccy": "BTC",
      "to": "6",
      "addr": "0x66d0edc2e63b6b992381ee668fbcb01f20ae0428",
      "verifiedName": "John Corner",
      "selected": true
    }
  ],
  "msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
addr	String	Deposit address
tag	String	Deposit tag (This will not be returned if the currency does not require a tag for deposit)
memo	String	Deposit memo (This will not be returned if the currency does not require a memo for deposit)
pmtId	String	Deposit payment ID (This will not be returned if the currency does not require a payment_id for deposit)
addrEx	Object	Deposit address attachment (This will not be returned if the currency does not require this) e.g. <code>TONCOIN</code> attached tag name is <code>comment</code> , the return will be <code>{'comment': '123456'}</code>

Parameter	Type	Description
ccy	String	Currency, e.g. <code>BTC</code>
chain	String	Chain name, e.g. <code>USDT-ERC20</code> , <code>USDT-TRC20</code>
to	String	The beneficiary account <code>6</code> : Funding account <code>18</code> : Trading account The users under some entity (e.g. Brazil) only support deposit to trading account.
verifiedName	String	Verified name (for recipient)
selected	Boolean	Return <code>true</code> if the current deposit address is selected by the website page
ctAddr	String	Last 6 digits of contract address

#### GET DEPOSIT HISTORY

Retrieve the deposit records according to the currency, deposit status, and time range in reverse chronological order. The 100 most recent records are returned by default.

WebSocket API is also available, refer to Deposit info channel.

**RATE LIMIT: 6 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/asset/deposit-history`

Request Example

```
import okx.Funding as Funding

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading: 0, Demo trading: 1

fundingAPI = Funding.FundingAPI(apikey, secretkey, passphrase, False, flag)

# Get deposit history
result = fundingAPI.get_deposit_history()
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	No	Currency, e.g. <code>BTC</code>
depId	String	No	Deposit ID
fromWdId	String	No	Internal transfer initiator's withdrawal ID If the deposit comes from internal transfer, this field displays the withdrawal ID of the internal transfer initiator
txId	String	No	Hash record of the deposit
type	String	No	Deposit Type <code>3</code> : internal transfer <code>4</code> : deposit from chain
state	String	No	Status of deposit <code>0</code> : waiting for confirmation

Parameter	Type	Required	Description
			<p>1: deposit credited      2: deposit successful      8: pending due to temporary deposit suspension on this crypto currency      11: match the address blacklist      12: account or deposit is frozen      13: sub-account deposit interception      14: KYC limit      17: Pending response from Travel Rule vendor</p>
after	String	No	Pagination of data to return records earlier than the requested ts, Unix timestamp format in milliseconds, e.g. <b>1654041600000</b>
before	String	No	Pagination of data to return records newer than the requested ts, Unix timestamp format in milliseconds, e.g. <b>1656633600000</b>
limit	string	No	Number of results per request. The maximum is <b>100</b> ; The default is <b>100</b>

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "actualDepBlkConfirm": "2",
      "amt": "1",
      "areaCodeFrom": "",
      "ccy": "USDT",
      "chain": "USDT-TRC20",
      "depId": "88****33",
      "from": "",
      "fromWdId": "",
      "state": "2",
      "to": "TN4hGjVXMzy*****9b4N1aGizqs",
      "ts": "1674038705000",
      "txId": "fee235b3e812*****857d36bb0426917f0df1802"
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency
chain	String	Chain name
amt	String	Deposit amount
from	String	Deposit account If the deposit comes from an internal transfer, this field displays the account information of the internal transfer initiator, which can be a mobile phone number or email address (masked), and will return "" in other cases
areaCodeFrom	String	If <b>from</b> is a phone number, this parameter return area code of the phone number
to	String	Deposit address If the deposit comes from the on-chain, this field displays the on-chain address, and will return "" in other cases
txId	String	Hash record of the deposit
ts	String	The timestamp that the deposit record is created, Unix timestamp format in milliseconds, e.g. <b>1655251200000</b>
state	String	Status of deposit <b>0</b> : Waiting for confirmation

Parameter	Type	Description
		<ul style="list-style-type: none"> <li><input type="checkbox"/> 1: Deposit credited</li> <li><input type="checkbox"/> 2: Deposit successful</li> <li><input type="checkbox"/> 8: Pending due to temporary deposit suspension on this crypto currency</li> <li><input type="checkbox"/> 11: Match the address blacklist</li> <li><input type="checkbox"/> 12: Account or deposit is frozen</li> <li><input type="checkbox"/> 13: Sub-account deposit interception</li> <li><input type="checkbox"/> 14: KYC limit</li> </ul>
depId	String	Deposit ID
fromWdId	String	<p>Internal transfer initiator's withdrawal ID If the deposit comes from internal transfer, this field displays the withdrawal ID of the internal transfer initiator, and will return "" in other cases</p>
actualDepBlkConfirm	String	The actual amount of blockchain confirmed in a single deposit

#### About deposit state

**Waiting for confirmation** is that the required number of blockchain confirmations has not been reached.

**Deposit credited** is that there is sufficient number of blockchain confirmations for the currency to be credited to the account, but it cannot be withdrawn yet.

**Deposit successful** means the crypto has been credited to the account and it can be withdrawn.

#### WITHDRAWAL

Only supported withdrawal of assets from funding account. Common sub-account does not support withdrawal.

The API can only make withdrawal to verified addresses/account, and verified addresses can be set by WEB/APP.

#### About tag

Some token deposits require a deposit address and a tag (e.g. Memo/Payment ID), which is a string that guarantees the uniqueness of your deposit address. Follow the deposit procedure carefully, or you may risk losing your assets.

For currencies with labels, if it is a withdrawal between OKX users, please use internal transfer instead of online withdrawal

#### The following content only applies to users residing in the United Arab Emirates

Due to local laws and regulations in your country or region, a certain ratio of user assets must be stored in cold wallets. We will perform cold-to-hot wallet asset transfers from time to time. However, if assets in hot wallets are not sufficient to meet user withdrawal demands, an extra step is needed to transfer cold wallet assets to the hot wallet. This may cause delays of up to 24 hours to receive withdrawals.

Learn more (<https://www.okx.com/help/what-is-a-segregated-wallet-and-why-is-my-withdrawal-delayed>)

#### Users under certain entities need to provide additional information for withdrawal

Bahamas entity users refer to [https://www.okx.com/docs-v5/log\\_en/#2024-08-08-withdrawal-api-adjustment-for-bahama-entity-users](https://www.okx.com/docs-v5/log_en/#2024-08-08-withdrawal-api-adjustment-for-bahama-entity-users)

#### RATE LIMIT: 6 REQUESTS PER SECOND

#### RATE LIMIT RULE: USER ID

#### PERMISSION: WITHDRAW

#### HTTP REQUEST

`POST /api/v5/asset/withdrawal`

#### Request Example

```

import okx.Funding as Funding

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading: 0, Demo trading: 1

fundingAPI = Funding.FundingAPI(apikey, secretkey, passphrase, False, flag)

# Withdrawal
result = fundingAPI.withdrawal(
    ccy="USDT",
    toAddr="T0xtvfb7cdrn6VX9H49mgio8bUxZ3DGfvYF",
    amt="100",
    dest="4",
    chain="USDT-TRC20"
)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	Yes	Currency, e.g. <code>USDT</code>
amt	String	Yes	<p>Withdrawal amount Withdrawal fee is not included in withdrawal amount. Please reserve sufficient transaction fees when withdrawing. You can get fee amount by Get currencies. For <code>internal transfer</code>, transaction fee is always 0.</p>
dest	String	Yes	<p>Withdrawal method ③: internal transfer ④: on-chain withdrawal</p>
toAddr	String	Yes	<p><code>toAddr</code> should be a trusted address/account. If your <code>dest</code> is ④, some crypto currency addresses are formatted as <code>'address:tag'</code>, e.g. <code>'ARDOR-7JF3-8F2E-QUWZ-CAN7F:123456'</code> If your <code>dest</code> is ③, <code>toAddr</code> should be a recipient address which can be UID, email, phone or login account name (account name is only for sub-account).</p>
toAddrType	String	No	<p>Address type ①: wallet address, email, phone, or login account name ②: UID (only for whitelisted users; applicable only when <code>dest=③</code>)</p>
chain	String	Conditional	<p>Chain name There are multiple chains under some currencies, such as <code>USDT</code> has <code>USDT-ERC20</code>, <code>USDT-TRC20</code> If the parameter is not filled in, the default will be the main chain. When you withdrawal the non-tradable asset, if the parameter is not filled in, the default will be the unique withdrawal chain. Apply to <code>on-chain withdrawal</code>. You can get supported chain name by the endpoint of Get currencies.</p>
areaCode	String	Conditional	<p>Area code for the phone number, e.g. <code>86</code> If <code>toAddr</code> is a phone number, this parameter is required. Apply to <code>internal transfer</code></p>
rcvrlInfo	Object	Conditional	<p>Recipient information For the specific entity users to do on-chain withdrawal/lightning withdrawal, this information is required.</p>

Parameter	Type	Required	Description
> walletType	String	Yes	<p>Wallet Type</p> <p><code>exchange</code>: Withdraw to exchange wallet</p> <p><code>private</code>: Withdraw to private wallet</p> <p>For the wallet belongs to business recipient, <code>rcvrFirstName</code> may input the company name, <code>rcvLastname</code> may input "N/A", location info may input the registered address of the company.</p>
> exchld	String	Conditional	<p>Exchange ID</p> <p>You can query supported exchanges through the endpoint of Get exchange list (public)</p> <p>If the exchange is not in the exchange list, fill in '0' in this field.</p> <p>Apply to walletType = <code>exchange</code></p>
> rcvrFirstName	String	Conditional	Receiver's first name, e.g. <code>Bruce</code>
> rcvrLastName	String	Conditional	Receiver's last name, e.g. <code>Wayne</code>
> rcvrCountry	String	Conditional	<p>The recipient's country, e.g. <code>United States</code></p> <p>You must enter an English country name or a two letter country code (ISO 3166-1). Please refer to the <code>Country Name</code> and <code>Country Code</code> in the country information table below.</p>
> rcvrCountrySubDivision	String	Conditional	State/Province of the recipient, e.g. <code>California</code>
> rcvrTownName	String	Conditional	The town/city where the recipient is located, e.g. <code>San Jose</code>
> rcvrStreetName	String	Conditional	Recipient's street address, e.g. <code>Clementi Avenue 1</code>
clientId	String	No	<p>Client-supplied ID</p> <p>A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.</p>

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "amt": "0.1",
      "wdId": "67485",
      "ccy": "BTC",
      "clientId": "",
      "chain": "BTC-Bitcoin"
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency
chain	String	Chain name, e.g. <code>USDT-ERC20</code> , <code>USDT-TRC20</code>
amt	String	Withdrawal amount
wdId	String	Withdrawal ID
clientId	String	<p>Client-supplied ID</p> <p>A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.</p>

#### COUNTRY INFORMATION

Country name	Country code
Afghanistan	AF
Albania	AL
Algeria	DZ
Andorra	AD
Angola	AO
Anguilla	AI
Antigua and Barbuda	AG
Argentina	AR
Armenia	AM
Australia	AU
Austria	AT
Azerbaijan	AZ
Bahamas	BS
Bahrain	BH
Bangladesh	BD
Barbados	BB
Belarus	BY
Belgium	BE
Belize	BZ
Benin	BJ
Bermuda	BM
Bhutan	BT
Bolivia	BO
Bosnia and Herzegovina	BA
Botswana	BW
Brazil	BR
British Virgin Islands	VG
Brunei	BN
Bulgaria	BG
Burkina Faso	BF
Burundi	BI
Cambodia	KH

Country name	Country code
Cameroon	CM
Canada	CA
Cape Verde	CV
Cayman Islands	KY
Central African Republic	CF
Chad	TD
Chile	CL
Colombia	CO
Comoros	KM
Congo (Republic)	CG
Congo (Democratic Republic)	CD
Costa Rica	CR
Cote d'Ivoire (Ivory Coast)	CI
Croatia	HR
Cuba	CU
Cyprus	CY
Czech Republic	CZ
Denmark	DK
Djibouti	DJ
Dominica	DM
Dominican Republic	DO
Ecuador	EC
Egypt	EG
El Salvador	SV
Equatorial Guinea	GQ
Eritrea	ER
Estonia	EE
Ethiopia	ET
Fiji	FJ
Finland	FI
France	FR
Gabon	GA

Country name	Country code
Gambia	GM
Georgia	GE
Germany	DE
Ghana	GH
Greece	GR
Grenada	GD
Guatemala	GT
Guinea	GN
Guinea-Bissau	GW
Guyana	GY
Haiti	HT
Honduras	HN
Hong Kong	HK
Hungary	HU
Iceland	IS
India	IN
Indonesia	ID
Iran	IR
Iraq	IQ
Ireland	IE
Israel	IL
Italy	IT
Jamaica	JM
Japan	JP
Jordan	JO
Kazakhstan	KZ
Kenya	KE
Kiribati	KI
North Korea	KP
South Korea	KR
Kuwait	KW
Kyrgyzstan	KG

Country name	Country code
Laos	LA
Latvia	LV
Lebanon	LB
Lesotho	LS
Liberia	LR
Libya	LY
Liechtenstein	LI
Lithuania	LT
Luxembourg	LU
Macau	MO
Macedonia	MK
Madagascar	MG
Malawi	MW
Malaysia	MY
Maldives	MV
Mali	ML
Malta	MT
Marshall Islands	MH
Mauritania	MR
Mauritius	MU
Mexico	MX
Micronesia	FM
Moldova	MD
Monaco	MC
Mongolia	MN
Montenegro	ME
Morocco	MA
Mozambique	MZ
Myanmar (Burma)	MM
Namibia	NA
Nauru	NR
Nepal	NP

Country name	Country code
Netherlands	NL
New Zealand	NZ
Nicaragua	NI
Niger	NE
Nigeria	NG
Norway	NO
Oman	OM
Pakistan	PK
Palau	PW
Panama	PA
Papua New Guinea	PG
Paraguay	PY
Peru	PE
Philippines	PH
Poland	PL
Portugal	PT
Qatar	QA
Romania	RO
Russia	RU
Rwanda	RW
Saint Kitts and Nevis	KN
Saint Lucia	LC
Saint Vincent and the Grenadines	VC
Samoa	WS
San Marino	SM
Sao Tome and Principe	ST
Saudi Arabia	SA
Senegal	SN
Serbia	RS
Seychelles	SC
Sierra Leone	SL
Singapore	SG

Country name	Country code
Slovakia	SK
Slovenia	SI
Solomon Islands	SB
Somalia	SO
South Africa	ZA
Spain	ES
Sri Lanka	LK
Sudan	SD
Suriname	SR
Swaziland	SZ
Sweden	SE
Switzerland	CH
Syria	SY
Taiwan	TW
Tajikistan	TJ
Tanzania	TZ
Thailand	TH
Timor-Leste (East Timor)	TL
Togo	TG
Tonga	TO
Trinidad and Tobago	TT
Tunisia	TN
Turkey	TR
Turkmenistan	TM
Tuvalu	TV
U.S. Virgin Islands	VI
Uganda	UG
Ukraine	UA
United Arab Emirates	AE
United Kingdom	GB
United States	US
Uruguay	UY

Country name	Country code
Uzbekistan	UZ
Vanuatu	VU
Vatican City	VA
Venezuela	VE
Vietnam	VN
Yemen	YE
Zambia	ZM
Zimbabwe	ZW
Kosovo	XK
South Sudan	SS
China	CN
Palestine	PS
Curacao	CW
Dominican Republic	DO
Dominican Republic	DO
Gibraltar	GI
New Caledonia	NC
Cook Islands	CK
Reunion	RE
Guernsey	GG
Guadeloupe	GP
Martinique	MQ
French Polynesia	PF
Faroe Islands	FO
Greenland	GL
Jersey	JE
Aruba	AW
Puerto Rico	PR
Isle of Man	IM
Guam	GU
Sint Maarten	SX
Turks and Caicos	TC

Country name	Country code
Åland Islands	AX
Caribbean Netherlands	BQ
British Indian Ocean Territory	IO
Christmas as Island	CX
Cocos (Keeling) Islands	CC
Falkland Islands (Islas Malvinas)	FK
Mayotte	YT
Niue	NU
Norfolk Island	NF
Northern Mariana Islands	MP
Pitcairn Islands	PN
Saint Helena, Ascension and Tristan da Cunha	SH
Collectivity of Saint Martin	MF
Saint Pierre and Miquelon	PM
Tokelau	TK
Wallis and Futuna	WF
American Samoa	AS

#### CANCEL WITHDRAWAL

You can cancel normal withdrawal requests, but you cannot cancel withdrawal requests on Lightning.

#### RATE LIMIT: 6 REQUESTS PER SECOND

#### RATE LIMIT RULE: USER ID

#### PERMISSION: TRADE

#### HTTP REQUEST

POST /api/v5/asset/cancel-withdrawal

#### Request Example

```
import okx.Funding as Funding

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading: 0, Demo trading: 1

fundingAPI = Funding.FundingAPI(apikey, secretkey, passphrase, False, flag)

# Cancel withdrawal
result = fundingAPI.cancel_withdrawal(
    wdId="123456"
)
print(result)
```

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
wdId	String	Yes	Withdrawal ID

**Response Example**

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "wdId": "1123456"
    }
  ]
}
```

**RESPONSE PARAMETERS**

Parameter	Type	Description
wdId	String	Withdrawal ID

If the code is equal to 0, it cannot be strictly considered that the withdrawal has been revoked. It only means that your request is accepted by the server. The actual result is subject to the status in the withdrawal history.

**GET WITHDRAWAL HISTORY**

Retrieve the withdrawal records according to the currency, withdrawal status, and time range in reverse chronological order. The 100 most recent records are returned by default.

WebSocket API is also available, refer to Withdrawal info channel.

**RATE LIMIT: 6 REQUESTS PER SECOND****RATE LIMIT RULE: USER ID****PERMISSION: READ****HTTP REQUEST**

```
GET /api/v5/asset/withdrawal-history
```

**Request Example****REQUEST PARAMETERS**

Parameter	Type	Required	Description
ccy	String	No	Currency, e.g. <code>BTC</code>
wdId	String	No	Withdrawal ID
clientId	String	No	Client-supplied ID A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
txId	String	No	Hash record of the deposit
type	String	No	Withdrawal type <code>3</code> : Internal transfer <code>4</code> : On-chain withdrawal
state	String	No	Status of withdrawal <ul style="list-style-type: none"> <li>Stage 1 : Pending withdrawal <code>19</code>: insufficient balance in the hot wallet <code>17</code>: Pending response from Travel Rule vendor</li> </ul>

Parameter	Type	Required	Description
			<p>10: Waiting transfer      0: Waiting withdrawal      4/5/6/8/9/12: Waiting manual review      7: Approved      &gt; 0, 17, 19 can be cancelled, other statuses cannot be cancelled</p> <ul style="list-style-type: none"> <li>Stage 2 : Withdrawal in progress (Applicable to on-chain withdrawals, internal transfers do not have this stage)           <ul style="list-style-type: none"> <li>1: Broadcasting your transaction to chain</li> <li>15: Pending transaction validation</li> <li>16: Due to local laws and regulations, your withdrawal may take up to 24 hours to arrive</li> <li>-3: Canceling</li> </ul> </li> <li>Final stage           <ul style="list-style-type: none"> <li>-2: Canceled</li> <li>-1: Failed</li> <li>2: Success</li> </ul> </li> </ul>
after	String	No	Pagination of data to return records earlier than the requested ts, Unix timestamp format in milliseconds, e.g. <code>1654041600000</code>
before	String	No	Pagination of data to return records newer than the requested ts, Unix timestamp format in milliseconds, e.g. <code>1656633600000</code>
limit	String	No	Number of results per request. The maximum is <code>100</code> ; The default is <code>100</code>

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "note": "",
      "chain": "ETH-Ethereum",
      "fee": "0.007",
      "feeCcy": "ETH",
      "ccy": "ETH",
      "clientId": "",
      "toAddrType": "1",
      "amt": "0.029809",
      "txId": "0x35c*****b360a174d",
      "from": "156****359",
      "areaCodeFrom": "86",
      "to": "0xa301fab*****7CF18C7B6C579",
      "areaCodeTo": "",
      "state": "2",
      "ts": "1655251200000",
      "nonTradableAsset": false,
      "wdId": "15447421"
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency
chain	String	Chain name, e.g. <code>USDT-ERC20</code> , <code>USDT-TRC20</code>
nonTradableAsset	Boolean	Whether it is a non-tradable asset or not <code>true</code> : non-tradable asset, <code>false</code> : tradable asset
amt	String	Withdrawal amount

Parameter	Type	Description
ts	String	Time the withdrawal request was submitted, Unix timestamp format in milliseconds, e.g. <code>1655251200000</code> .
from	String	Withdrawal account It can be <code>email</code> / <code>phone</code> / <code>sub-account name</code>
areaCodeFrom	String	Area code for the phone number If <code>from</code> is a phone number, this parameter returns the area code for the phone number
to	String	Receiving address
areaCodeTo	String	Area code for the phone number If <code>to</code> is a phone number, this parameter returns the area code for the phone number
toAddrType	String	Address type <code>1</code> : wallet address, email, phone, or login account name <code>2</code> : UID
tag	String	Some currencies require a tag for withdrawals. This is not returned if not required.
pmtId	String	Some currencies require a payment ID for withdrawals. This is not returned if not required.
memo	String	Some currencies require this parameter for withdrawals. This is not returned if not required.
addrEx	Object	Withdrawal address attachment (This will not be returned if the currency does not require this) e.g. TONCOIN attached tag name is comment, the return will be <code>{'comment':'123456'}</code>
txId	String	Hash record of the withdrawal This parameter will return <code>""</code> for internal transfers.
fee	String	Withdrawal fee amount
feeCcy	String	Withdrawal fee currency, e.g. <code>USDT</code>
state	String	Status of withdrawal
wdId	String	Withdrawal ID
clientId	String	Client-supplied ID
note	String	Withdrawal note

#### GET DEPOSIT WITHDRAW STATUS

Retrieve deposit's and withdrawal's detailed status and estimated complete time.

**RATE LIMIT: 1 REQUEST PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/asset/deposit-withdraw-status`

Request Example

#### REQUEST PARAMETERS

Parameters	Types	Required	Description
wdId	String	Conditional	Withdrawal ID, use to retrieve withdrawal status Required to input one and only one of <code>wdId</code> and <code>txId</code>

Parameters	Types	Required	Description
txId	String	Conditional	Hash record of the deposit, use to retrieve deposit status Required to input one and only one of <code>wdId</code> and <code>txId</code>
ccy	String	Conditional	Currency type, e.g. <code>USDT</code> Required when retrieving deposit status with <code>txId</code>
to	String	Conditional	To address, the destination address in deposit Required when retrieving deposit status with <code>txId</code>
chain	String	Conditional	Currency chain information, e.g. USDT-ERC20 Required when retrieving deposit status with <code>txId</code>

### Response Example

```
{
  "code": "0",
  "data": [
    {
      "wdId": "200045249",
      "txId": "16f3638329xxxxxx42d988f97",
      "state": "Pending withdrawal: Wallet is under maintenance, please wait.",
      "estCompleteTime": "01/09/2023, 8:10:48 PM"
    }
  ],
  "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
estCompleteTime	String	Estimated complete time The timezone is <code>UTC+8</code> . The format is MM/dd/yyyy, h:mm:ss AM/PM estCompleteTime is only an approximate estimated time, for reference only.
state	String	The detailed stage and status of the deposit/withdrawal The message in front of the colon is the stage; the message after the colon is the ongoing status.
txId	String	Hash record on-chain For withdrawal, if the <code>txId</code> has already been generated, it will return the value, otherwise, it will return "".
wdId	String	Withdrawal ID When retrieving deposit status, wdId returns blank "".

### Stage References

#### Deposit

Stage 1: On-chain transaction detection

Stage 2: Push deposit data to associated account

Stage 3: Receiving account credit

Final stage: Deposit complete

#### Withdrawal

Stage 1: Pending withdrawal

Stage 2: Withdrawal in progress

Final stage: Withdrawal complete / cancellation complete

### GET EXCHANGE LIST (PUBLIC)

Authentication is not required for this public endpoint.

### RATE LIMIT: 6 REQUESTS PER SECOND

### RATE LIMIT RULE: IP

## HTTP REQUEST

```
GET /api/v5/asset/exchange-list
```

### Request Example

## REQUEST PARAMETERS

None

### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "exchId": "did:ethr:0xfeb4f99829a9acdf52979abee87e83addf22a7e1",
      "exchName": "1xbet"
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
exchName	String	Exchange name, e.g. <code>1xbet</code>
exchId	String	Exchange ID, e.g. <code>did:ethr:0xfeb4f99829a9acdf52979abee87e83addf22a7e1</code>

## APPLY FOR MONTHLY STATEMENT (LAST YEAR)

Apply for monthly statement in the past year.

## RATE LIMIT: 20 REQUESTS PER MONTH

## RATE LIMIT RULE: USER ID

## PERMISSION: READ

## HTTP REQUEST

```
POST /api/v5/asset/monthly-statement
```

### Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
month	String	No	Month, last month by default. Valid value is <code>Jan</code> , <code>Feb</code> , <code>Mar</code> , <code>Apr</code> , <code>May</code> , <code>Jun</code> , <code>Jul</code> , <code>Aug</code> , <code>Sep</code> , <code>Oct</code> , <code>Nov</code> , <code>Dec</code>

### Response Example

```
{
  "code": "0",
  "data": [
    {
      "ts": "1646892328000"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ts	String	Download link generation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

#### GET MONTHLY STATEMENT (LAST YEAR)

Retrieve monthly statement in the past year.

**RATE LIMIT: 10 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

#### HTTP REQUEST

`GET /api/v5/asset/monthly-statement`

Request Example

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
month	String	Yes	Month, valid value is <code>Jan</code> , <code>Feb</code> , <code>Mar</code> , <code>Apr</code> , <code>May</code> , <code>Jun</code> , <code>Jul</code> , <code>Aug</code> , <code>Sep</code> , <code>Oct</code> , <code>Nov</code> , <code>Dec</code>

Response Example

```
{
  "code": "0",
  "data": [
    {
      "fileHref": "http://xxx",
      "state": "finished",
      "ts": 1646892328000
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
fileHref	String	Download file link
ts	Int	Download link generation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
state	String	Download link status "finished" "ongoing"

#### GET CONVERT CURRENCIES

**RATE LIMIT: 6 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

#### HTTP REQUEST

`GET /api/v5/asset/convert/currencies`

Request Example

#### RESPONSE PARAMETERS

none

Response Example

```
{
  "code": "0",
  "data": [
    {
      "min": "", // Deprecated
      "max": "", // Deprecated
      "ccy": "BTC"
    },
    {
      "min": "",
      "max": "",
      "ccy": "ETH"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency, e.g. BTC
min	String	Minimum amount to convert ( Deprecated )
max	String	Maximum amount to convert ( Deprecated )

#### GET CONVERT CURRENCY PAIR

**RATE LIMIT: 6 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

#### HTTP REQUEST

```
GET /api/v5/asset/convert/currency-pair
```

Request Example

#### RESPONSE PARAMETERS

Parameters	Types	Required	Description
fromCcy	String	Yes	Currency to convert from, e.g. <code>USDT</code>
toCcy	String	Yes	Currency to convert to, e.g. <code>BTC</code>

Response Example

```
{
  "code": "0",
  "data": [
    {
      "baseCcy": "BTC",
      "baseCcyMax": "0.5",
      "baseCcyMin": "0.0001",
      "instId": "BTC-USDT",
      "quoteCcy": "USDT",
      "quoteCcyMax": "10000",
      "quoteCcyMin": "1"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
instId	String	Currency pair, e.g. <code>BTC-USDT</code>
baseCcy	String	Base currency, e.g. <code>BTC</code> in <code>BTC-USDT</code>
baseCcyMax	String	Maximum amount of base currency
baseCcyMin	String	Minimum amount of base currency
quoteCcy	String	Quote currency, e.g. <code>USDT</code> in <code>BTC-USDT</code>
quoteCcyMax	String	Maximum amount of quote currency
quoteCcyMin	String	Minimum amount of quote currency

#### ESTIMATE QUOTE

RATE LIMIT: 10 REQUESTS PER SECOND

RATE LIMIT RULE: USER ID

RATE LIMIT: 1 REQUEST PER 5 SECONDS

RATE LIMIT RULE: INSTRUMENT ID

PERMISSION: TRADE

#### HTTP REQUEST

`POST /api/v5/asset/convert/estimate-quote`

Request Example

#### REQUEST PARAMETERS

Parameters	Types	Required	Description
baseCcy	String	Yes	Base currency, e.g. <code>BTC</code> in <code>BTC-USDT</code>
quoteCcy	String	Yes	Quote currency, e.g. <code>USDT</code> in <code>BTC-USDT</code>
side	String	Yes	Trade side based on <code>baseCcy</code> <code>buy</code> <code>sell</code>
rfqSz	String	Yes	RFQ amount
rfqSzCcy	String	Yes	RFQ currency
clQReqId	String	No	Client Order ID as assigned by the client A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
tag	String	No	Order tag Applicable to broker user

Response Example

```
{
  "code": "0",
  "data": [
    {
      "baseCcy": "ETH",
      "baseSz": "0.01023052",
      "clQReqId": "",
      "cnvtPx": "2932.40104429",
      "origRfqSz": "30",
      "quoteCcy": "USDT",
      "quoteId": "quoterETH-USDT16461885104612381",
      "quoteSz": "30",
      "quoteTime": "1646188510461"
    }
  ]
}
```

```

        "rfqSz": "30",
        "rfqSzCcy": "USDT",
        "side": "buy",
        "ttlMs": "10000"
    },
],
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
quoteTime	String	Quotation generation time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
ttlMs	String	Validity period of quotation in milliseconds
clQReqId	String	Client Order ID as assigned by the client
quotId	String	Quote ID
baseCcy	String	Base currency, e.g. <code>BTC</code> in <code>BTC-USDT</code>
quoteCcy	String	Quote currency, e.g. <code>USDT</code> in <code>BTC-USDT</code>
side	String	Trade side based on <code>baseCcy</code>
origRfqSz	String	Original RFQ amount
rfqSz	String	Real RFQ amount
rfqSzCcy	String	RFQ currency
cnvtPx	String	Convert price based on quote currency
baseSz	String	Convert amount of base currency
quoteSz	String	Convert amount of quote currency

## CONVERT TRADE

You should make estimate quote before convert trade.

Only assets in the trading account supported convert.

### RATE LIMIT: 10 REQUESTS PER SECOND

### RATE LIMIT RULE: USER ID

### PERMISSION: TRADE

For the same side (buy/sell), there's a trading limit of 1 request per 5 seconds.

### HTTP REQUEST

`POST /api/v5/asset/convert/trade`

Request Example

## REQUEST PARAMETERS

Parameters	Types	Required	Description
quotId	String	Yes	Quote ID
baseCcy	String	Yes	Base currency, e.g. <code>BTC</code> in <code>BTC-USDT</code>

Parameters	Types	Required	Description
quoteCcy	String	Yes	Quote currency, e.g. <code>USDT</code> in <code>BTC-USDT</code>
side	String	Yes	Trade side based on <code>baseCcy</code> <code>buy</code> <code>sell</code>
sz	String	Yes	Quote amount The quote amount should no more then RFQ amount
szCcy	String	Yes	Quote currency
clTReqId	String	No	Client Order ID as assigned by the client A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
tag	String	No	Order tag Applicable to broker user

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "baseCcy": "ETH",
      "clTReqId": "",
      "fillBaseSz": "0.01023052",
      "fillPx": "2932.40104429",
      "fillQuoteSz": "30",
      "instId": "ETH-USDT",
      "quoteCcy": "USDT",
      "quoteId": "quoterETH-USDT16461885104612381",
      "side": "buy",
      "state": "fullyFilled",
      "tradeId": "trader16461885203381437",
      "ts": "1646188520338"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
tradId	String	Trade ID
quoteId	String	Quote ID
clTReqId	String	Client Order ID as assigned by the client
state	String	Trade state <code>fullyFilled</code> : success <code>rejected</code> : failed
instId	String	Currency pair, e.g. <code>BTC-USDT</code>
baseCcy	String	Base currency, e.g. <code>BTC</code> in <code>BTC-USDT</code>
quoteCcy	String	Quote currency, e.g. <code>USDT</code> in <code>BTC-USDT</code>
side	String	Trade side based on <code>baseCcy</code> <code>buy</code> <code>sell</code>
fillPx	String	Filled price based on quote currency

Parameter	Type	Description
fillBaseSz	String	Filled amount for base currency
fillQuoteSz	String	Filled amount for quote currency
ts	String	Convert trade time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## GET CONVERT HISTORY

**RATE LIMIT: 6 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

### HTTP REQUEST

`GET /api/v5/asset/convert/history`

Request Example

### REQUEST PARAMETERS

Parameters	Types	Required	Description
clTReqId	String	No	Client Order ID as assigned by the client A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
after	String	No	Pagination of data to return records earlier than the requested <code>ts</code> , Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
before	String	No	Pagination of data to return records newer than the requested <code>ts</code> , Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. The maximum is <code>100</code> . The default is <code>100</code> .
tag	String	No	Order tag Applicable to broker user If the convert trading used <code>tag</code> , this parameter is also required.

Response Example

```
{
  "code": "0",
  "data": [
    {
      "clTReqId": "",
      "instId": "ETH-USDT",
      "side": "buy",
      "fillPx": "2932.401044",
      "baseCcy": "ETH",
      "quoteCcy": "USDT",
      "fillBaseSz": "0.01023052",
      "state": "fullyFilled",
      "tradeId": "trader16461885203381437",
      "fillQuoteSz": "30",
      "ts": "1646188520000"
    }
  ],
  "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
tradeId	String	Trade ID
clTReqId	String	Client Order ID as assigned by the client

Parameter	Type	Description
state	String	Trade state fullyFilled : success rejected : failed
instId	String	Currency pair, e.g. <code>BTC-USDT</code>
baseCcy	String	Base currency, e.g. <code>BTC</code> in <code>BTC-USDT</code>
quoteCcy	String	Quote currency, e.g. <code>USDT</code> in <code>BTC-USDT</code>
side	String	Trade side based on <code>baseCcy</code> <code>buy</code> <code>sell</code>
fillPx	String	Filled price based on quote currency
fillBaseSz	String	Filled amount for base currency
fillQuoteSz	String	Filled amount for quote currency
ts	String	Convert trade time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

#### GET DEPOSIT PAYMENT METHODS

To display all the available fiat deposit payment methods

**RATE LIMIT: 3 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/ fiat/ deposit-payment-methods`

Request Example

(Empty box for request example)

#### REQUEST PARAMETERS

Parameters	Types	Required	Description
ccy	String	Yes	Fiat currency, ISO-4217 3 digit currency code, e.g. <code>TRY</code>

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ccy": "TRY",
      "paymentMethod": "TR_BANKS",
      "feeRate": "0",
      "minFee": "0",
      "limits": {
        "dailyLimit": "2147483647",
        "dailyLimitRemaining": "2147483647",
        "weeklyLimit": "2147483647",
        "weeklyLimitRemaining": "2147483647",
        "monthlyLimit": "",
        "monthlyLimitRemaining": "",
        "maxAmt": "1000000",
        "minAmt": "1",
        "lifetimeLimit": "2147483647"
      }
    }
  ]
}
```

```

"accounts": [
  {
    "paymentAcctId": "1",
    "acctNum": "TR740001592093703829602611",
    "recipientName": "John Doe",
    "bankName": "VakifBank",
    "bankCode": "TVBATR2AXXX",
    "state": "active"
  },
  {
    "paymentAcctId": "2",
    "acctNum": "TR740001592093703829602622",
    "recipientName": "John Doe",
    "bankName": "FBHLTRISXXX",
    "bankCode": "",
    "state": "active"
  }
]
}
]
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Fiat currency
paymentMethod	String	<p>The payment method associated with the currency</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <span>TR_BANKS</span> <span>PIX</span> <span>SEPA</span> <span>XPULSE</span> <span>NPP</span> </div>
feeRate	String	<p>The fee rate for each deposit, expressed as a percentage</p> <p>e.g. <code>0.02</code> represents 2 percent fee for each transaction.</p>
minFee	String	The minimum fee for each deposit
limits	Object	An object containing limits for various transaction intervals
> dailyLimit	String	The daily transaction limit
> dailyLimitRemaining	String	The remaining daily transaction limit
> weeklyLimit	String	The weekly transaction limit
> weeklyLimitRemaining	String	The remaining weekly transaction limit
> monthlyLimit	String	The monthly transaction limit
> monthlyLimitRemaining	String	The remaining monthly transaction limit
> maxAmt	String	The maximum amount allowed per transaction
> minAmt	String	The minimum amount allowed per transaction
> lifetimeLimit	String	The lifetime transaction limit. Return the configured value, <code>""</code> if not configured
accounts	Array of Object	An array containing information about payment accounts associated with the currency and method.
> paymentAcctId	String	The account ID for withdrawal
> acctNum	String	The account number, which can be an IBAN or other bank account number.

Parameter	Type	Description
> recipientName	String	The name of the recipient
> bankName	String	The name of the bank associated with the account
> bankCode	String	The SWIFT code / BIC / bank code associated with the account
> state	String	The state of the account <code>active</code>

#### GET WITHDRAWAL PAYMENT METHODS

To display all the available fiat withdrawal payment methods

**RATE LIMIT: 3 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/ fiat/ withdrawal-methods`

Request Example

(Empty response box)

#### REQUEST PARAMETERS

Parameters	Types	Required	Description
ccy	String	Yes	Fiat currency, ISO-4217 3 digit currency code. e.g. <code>TRY</code>

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ccy": "TRY",
      "paymentMethod": "TR_BANKS",
      "feeRate": "0.02",
      "minFee": "1",
      "limits": {
        "dailyLimit": "",
        "dailyLimitRemaining": "",
        "weeklyLimit": "",
        "weeklyLimitRemaining": "",
        "monthlyLimit": "",
        "monthlyLimitRemaining": "",
        "maxAmt": "",
        "minAmt": "",
        "lifetimeLimit": ""
      },
      "accounts": [
        {
          "paymentAcctId": "1",
          "acctNum": "TR740001592093703829602668",
          "recipientName": "John Doe",
          "bankName": "VakifBank",
          "bankCode": "TVBAGR2AXXX",
          "state": "active"
        },
        {
          "paymentAcctId": "2",
          "acctNum": "TR740001592093703829603024",
          "recipientName": "John Doe",
          "bankName": "Şekerbank",
          "state": "active"
        }
      ]
    }
  ]
}
```

```

        "bankCode": "SEKETR2AXXX",
        "state": "active"
    }
]
}
]
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Fiat currency
paymentMethod	String	<p>The payment method associated with the currency</p> <div style="display: flex; justify-content: center; gap: 10px;"> <span>TR_BANKS</span> <span>PIX</span> <span>SEPA</span> <span>XPULSE</span> <span>NPP</span> </div>
feeRate	String	<p>The fee rate for each deposit, expressed as a percentage</p> <p>e.g. <code>0.02</code> represents 2 percent fee for each transaction.</p>
minFee	String	The minimum fee for each deposit
limits	Object	An object containing limits for various transaction intervals
> dailyLimit	String	The daily transaction limit
> dailyLimitRemaining	String	The remaining daily transaction limit
> weeklyLimit	String	The weekly transaction limit
> weeklyLimitRemaining	String	The remaining weekly transaction limit
> monthlyLimit	String	The monthly transaction limit
> monthlyLimitRemaining	String	The remaining monthly transaction limit
> minAmt	String	The minimum amount allowed per transaction
> maxAmt	String	The maximum amount allowed per transaction
> lifetimeLimit	String	The lifetime transaction limit. Return the configured value, <code>""</code> if not configured
accounts	Array of Object	An array containing information about payment accounts associated with the currency and method.
> paymentAcctId	String	The account ID for withdrawal
> acctNum	String	The account number, which can be an IBAN or other bank account number.
> recipientName	String	The name of the recipient
> bankName	String	The name of the bank associated with the account
> bankCode	String	The SWIFT code / BIC / bank code associated with the account
> state	String	<p>The state of the account</p> <div style="display: flex; justify-content: center; gap: 10px;"> <span>active</span> </div>

#### CREATE WITHDRAWAL ORDER

Initiate a fiat withdrawal request (Authenticated endpoint, Only for API keys with "Withdrawal" access)

Only supported withdrawal of assets from funding account.

**RATE LIMIT: 3 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: WITHDRAW**

**HTTP REQUEST**

POST /api/v5/ fiat/create-withdrawal

Request Example

**REQUEST PARAMETERS**

Parameters	Type	Required	Description
paymentAcctId	String	Yes	Payment account id to withdraw to, retrieved from get withdrawal payment methods API
ccy	String	Yes	Currency for withdrawal, must match currency allowed for paymentMethod
amt	String	Yes	Requested withdrawal amount before fees. Has to be less than or equal to 2 decimal points double
paymentMethod	String	Yes	Payment method to use for withdrawal  <div style="display: flex; justify-content: space-around; align-items: center;"><span>TR_BANKS</span><span>PIX</span><span>SEPA</span><span>XPULSE</span><span>NPP</span></div>
clientId	String	Yes	Client-supplied ID, A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters  e.g. 194a6975e98246538faeb0fab0d502df

Response Example

```
{  
  "code": "0",  
  "msg": "",  
  "data": [  
    {  
      "cTime": "1707429385000",  
      "uTime": "1707429385000",  
      "ordId": "124041201450544699",  
      "paymentMethod": "TR_BANKS",  
      "paymentAcctId": "20",  
      "fee": "0",  
      "amt": "100",  
      "ccy": "TRY",  
      "state": "completed",  
      "clientId": "194a6975e98246538faeb0fab0d502df"  
    }  
  ]  
}
```

**RESPONSE PARAMETERS**

Parameter	Type	Description
ordId	String	The unique order Id
clientId	String	The client ID associated with the transaction

Parameter	Type	Description
amt	String	The requested amount for the transaction
ccy	String	The currency of the transaction
fee	String	The transaction fee
paymentAcctId	String	The Id of the payment account used
paymentMethod	String	Payment Method TR_BANKS PIX SEPA
state	String	The State of the transaction processing completed
cTime	String	The creation time of the transaction, Unix timestamp format in milliseconds, e.g. 1597026383085
uTime	String	The update time of the transaction, Unix timestamp format in milliseconds, e.g. 1597026383085

#### CANCEL WITHDRAWAL ORDER

Cancel a pending fiat withdrawal order, currently only applicable to TRY

**RATE LIMIT: 3 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

POST /api/v5/coin/cancel-withdrawal

Request Example

#### REQUEST PARAMETERS

Parameters	Types	Required	Description
ordId	String	Yes	Payment Order Id

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ordId": "124041201450544699",
      "state": "canceled"
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ordId	String	Payment Order ID

Parameter	Type	Description
state	String	The state of the transaction, e.g. <code>canceled</code>

#### GET WITHDRAWAL ORDER HISTORY

Get fiat withdrawal order history

**RATE LIMIT: 3 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

#### HTTP REQUEST

`GET /api/v5/ fiat/withdrawal-order-history`

Request Example

#### REQUEST PARAMETERS

Parameters	Types	Required	Description
ccy	String	No	Fiat currency, ISO-4217 3 digit currency code, e.g. <code>TRY</code>
paymentMethod	String	No	<p>Payment Method</p> <p><code>TR_BANKS</code></p> <p><code>PIX</code></p> <p><code>SEPA</code></p> <p><code>XPULSE</code></p> <p><code>NPP</code></p>
state	String	No	<p>State of the order</p> <p><code>completed</code></p> <p><code>failed</code></p> <p><code>pending</code></p> <p><code>canceled</code></p> <p><code>inqueue</code></p> <p><code>processing</code></p>
after	String	No	Filter with a begin timestamp. Unix timestamp format in milliseconds (inclusive), e.g. <code>1597026383085</code>
before	String	No	Filter with an end timestamp. Unix timestamp format in milliseconds (inclusive), e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. Maximum and default is <code>100</code>

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "cTime": "1707429385000",
      "uTime": "1707429385000",
      "ordId": "124041201450544699",
      "paymentMethod": "TR_BANKS",
      "paymentAcctId": "20",
      "amt": "10000",
      "fee": "0",
      "ccy": "TRY",
      "state": "completed",
      "clientId": "194a6975e98246538faeb0fab0d502df"
    },
    {
      "cTime": "1707429385000",
      "uTime": "1707429385000",
      "ordId": "124041201450544699",
      "paymentMethod": "TR_BANKS",
      "paymentAcctId": "20",
      "amt": "10000",
      "fee": "0",
      "ccy": "TRY",
      "state": "completed",
      "clientId": "194a6975e98246538faeb0fab0d502df"
    }
  ]
}
```

```

        "cTime": "1707429385000",
        "uTime": "1707429385000",
        "ordId": "124041201450544690",
        "paymentMethod": "TR_BANKS",
        "paymentAcctId": "20",
        "amt": "5000",
        "fee": "0",
        "ccy": "TRY",
        "state": "completed",
        "clientId": "164a6975e48946538faeb0fab0d414fg"
    }
]
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ordId	String	Unique Order Id
clientId	String	Client Id of the transaction
amt	String	Final amount of the transaction
ccy	String	Currency of the transaction
fee	String	Transaction fee
paymentAcctId	String	ID of the payment account used
paymentMethod	String	Payment method type
state	String	State of the transaction <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">completed</div> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">failed</div> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">pending</div> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">canceled</div> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">inqueue</div> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">processing</div>
cTime	String	Creation time of the transaction, Unix timestamp format in milliseconds, e.g. <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">1597026383085</div>
uTime	String	Update time of the transaction, Unix timestamp format in milliseconds, e.g. <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">1597026383085</div>

#### GET WITHDRAWAL ORDER DETAIL

Get fiat withdraw order detail

**RATE LIMIT: 3 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

#### HTTP REQUEST

`GET /api/v5/coin/withdrawal`

Request Example

#### REQUEST PARAMETERS

Parameters	Types	Required	Description
ordId	String	Yes	Order ID

## Response Example

```
{  
  "code": "0",  
  "msg": "",  
  "data": [  
    {  
      "cTime": "1707429385000",  
      "uTime": "1707429385000",  
      "ordId": "024041201450544699",  
      "paymentMethod": "TR_BANKS",  
      "paymentAcctId": "20",  
      "amt": "100",  
      "fee": "0",  
      "ccy": "TRY",  
      "state": "completed",  
      "clientId": "194a6975e98246538faeb0fab0d502df"  
    }  
  ]  
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ordId	String	Order ID
clientId	String	The original request ID associated with the transaction
ccy	String	The currency of the transaction
amt	String	Amount of the transaction
fee	String	The transaction fee
paymentAcctId	String	The ID of the payment account used
paymentMethod	String	Payment method, e.g. <code>TR_BANKS</code> <code>PIX</code> <code>SEPA</code> <code>XPULSE</code> <code>NPP</code>
state	String	The state of the transaction <code>completed</code> <code>failed</code> <code>pending</code> <code>canceled</code> <code>inqueue</code> <code>processing</code>
cTime	String	The creation time of the transaction, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
uTime	String	The update time of the transaction, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## GET DEPOSIT ORDER HISTORY

Get fiat deposit order history

**RATE LIMIT: 3 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

## HTTP REQUEST

```
GET /api/v5/fiat/deposit-order-history
```

## Request Example

### REQUEST PARAMETERS

Parameters	Types	Required	Description
ccy	String	No	ISO-4217 3 digit currency code
paymentMethod	String	No	Payment Method <code>TR_BANKS</code> <code>PIX</code> <code>SEPA</code> <code>XPULSE</code> <code>NPP</code>
state	String	No	State of the order <code>completed</code> <code>failed</code> <code>pending</code> <code>canceled</code> <code>inqueue</code> <code>processing</code>
after	String	No	Filter with a begin timestamp. Unix timestamp format in milliseconds (inclusive), e.g. <code>1597026383085</code>
before	String	No	Filter with an end timestamp. Unix timestamp format in milliseconds (inclusive), e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. Maximum and default is 100

### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "cTime": "1707429385000",
      "uTime": "1707429385000",
      "ordId": "024041201450544699",
      "paymentMethod": "TR_BANKS",
      "paymentAcctId": "20",
      "amt": "10000",
      "fee": "0",
      "ccy": "TRY",
      "state": "completed",
      "clientId": ""
    },
    {
      "cTime": "1707429385000",
      "uTime": "1707429385000",
      "ordId": "024041201450544690",
      "paymentMethod": "TR_BANKS",
      "paymentAcctId": "20",
      "amt": "50000",
      "fee": "0",
      "ccy": "TRY",
      "state": "completed",
      "clientId": ""
    }
  ]
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
ordId	String	Unique Order ID
clientId	String	Client Id of the transaction
ccy	String	Currency of the transaction
amt	String	Final amount of the transaction
fee	String	Transaction fee
paymentAcctId	String	ID of the payment account used
paymentMethod	String	Payment Method, e.g. <code>TR_BANKS</code>
state	String	State of the transaction <code>completed</code> <code>failed</code> <code>pending</code> <code>canceled</code> <code>inqueue</code>
cTime	String	Creation time of the transaction, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
uTime	String	Update time of the transaction, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

#### GET DEPOSIT ORDER DETAIL

Get fiat deposit order detail

**RATE LIMIT: 3 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

#### HTTP REQUEST

`GET /api/v5/fiat/deposit`

Request Example

#### REQUEST PARAMETERS

Parameters	Types	Required	Description
ordId	String	Yes	Order ID

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "cTime": "1707429385000",
      "uTime": "1707429385000",
      "ordId": "024041201450544699",
      "paymentMethod": "TR_BANKS",
      "paymentAcctId": "20",
      "amt": "100",
      "fee": "0",
      "ccy": "TRY",
      "state": "completed",
      "clientId": ""
    }
  ]
}
```

```
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ordId	String	Order ID
clientId	String	The original request ID associated with the transaction. If it's a deposit, it's most likely an empty string ("").
amt	String	Amount of the transaction
ccy	String	The currency of the transaction
fee	String	The transaction fee
paymentAcctId	String	The ID of the payment account used
paymentMethod	String	Payment method, e.g. <code>TR_BANKS</code> <code>PIX</code> <code>SEPA</code> <code>XPULSE</code> <code>NPP</code>
state	String	The state of the transaction <code>completed</code> <code>failed</code> <code>pending</code> <code>canceled</code> <code>inqueue</code> <code>processing</code>
cTime	String	The creation time of the transaction, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
uTime	String	The update time of the transaction, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## GET BUY/SELL CURRENCIES

**RATE LIMIT: 6 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

### HTTP REQUEST

```
GET /api/v5/fiat/buy-sell/currencies
```

Request Example

Response Example

```
{
  "code": "0",
  "data": [
    {
      "fiatCcyList": [
        {
          "ccy": "USD"
        },
        {
          "ccy": "EUR"
        },
        ...
      ],
      "cryptoCcyList": [
        ...
      ]
    }
  ]
}
```

```

    "ccy": "BTC"
  },
  ...
],
],
"msg": ""
}
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
fiatCcyList	Array of objects	Fiat currency list
>ccy	String	Currency, e.g. <code>BTC</code>
cryptoCcyList	Array of objects	Crypto currency list
>ccy	String	Currency, e.g. <code>USD</code>

This feature is only available to Bahamas institutional users at the moment.

## GET BUY/SELL CURRENCY PAIR

**RATE LIMIT: 6 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

```
GET /api/v5/flat/buy-sell/currency-pair
```

Request Example

## REQUEST PARAMETERS

Parameters	Types	Required	Description
fromCcy	String	Yes	Currency to sell, e.g. <code>USD</code>
toCcy	String	Yes	Currency to buy, e.g. <code>BTC</code>

Response Example

```
{
  "code": "0",
  "data": [
    {
      "side": "buy",
      "fromCcy": "USD",
      "toCcy": "BTC",
      "singleTradeMax": "1",
      "singleTradeMin": "0.01",
      "fixedPxRemainingDailyQuota": "",
      "fixedPxDailyLimit": "",
      "paymentMethods": ["balance"]
    }
  ],
  "msg": ""
}

{
  "code": "0",
  "data": [

```

```

  "code": "0",
  "data": [

```

```

    "side": "sell",
    "fromCcy": "BTC",
    "toCcy": "USD",
    "singleTradeMax": "1",
    "singleTradeMin": "0.01",
    "fixedPxRemainingDailyQuota": "",
    "fixedPxDailyLimit": "",
    "paymentMethods": ["balance"]
  }
],
"msg": ""
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
side	String	Side buy: Fiat to crypto sell: Crypto to fiat May support both sides in the future, separated with a comma, e.g. <code>buy,sell</code> .
fromCcy	String	Currency to sell, e.g. <code>USD</code>
toCcy	String	Currency to buy, e.g. <code>BTC</code>
singleTradeMax	String	The maximum amount of currency for a single trade, unit in <code>fromCcy</code>
singleTradeMin	String	The minimum amount of currency for a single trade, unit in <code>fromCcy</code>
fixedPxDailyLimit	String	Fixed price daily limit Applicable to Fiat to Fiat trade, else return ". If <code>side</code> = <code>buy</code> , unit in <code>fromCcy</code> If <code>side</code> = <code>sell</code> , unit in <code>toCcy</code>
fixedPxRemainingDailyQuota	String	Fixed price remaining daily quota Applicable to Fiat to Fiat trade, else return ". If <code>side</code> = <code>buy</code> , unit in <code>fromCcy</code> If <code>side</code> = <code>sell</code> , unit in <code>toCcy</code>
paymentMethods	Array of strings	Supported payment methods <code>balance</code> e.g. <code>["balance"]</code>

This feature is only available to Bahamas institutional users at the moment.

#### GET BUY/SELL QUOTE

**RATE LIMIT: 10 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**RATE LIMIT: 1 REQUEST PER 5 SECONDS**

**RATE LIMIT RULE: INSTRUMENT ID**

**PERMISSION: READ**

#### HTTP REQUEST

`POST /api/v5/flat/buy-sell/quote`

Request Example

#### REQUEST PARAMETERS

Parameters	Types	Required	Description
side	String	Yes	Side buy: Buy Crypto / Fiat with Fiat sell: Sell Crypto to Crypto / Fiat
fromCcy	String	Yes	Currency to sell
toCcy	String	Yes	Currency to buy
rfqAmt	String	Yes	RFQ amount
rfqCcy	String	Yes	RFQ currency

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "quoteId": "quoterBTC-USD16461885104612381",
      "fromCcy": "USD",
      "toCcy": "BTC",
      "rfqAmt": "30",
      "rfqCcy": "USD",
      "quotePx": "2932.40104429",
      "quoteCcy": "USD",
      "quoteFromAmt": "30",
      "quoteToAmt": "30",
      "quoteTime": "1646188510461",
      "ttlMs": "10000"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
quotId	String	Quote ID
side	String	Side buy: Buy Crypto / Fiat with Fiat sell: Sell Crypto to Crypto / Fiat
fromCcy	String	Currency to sell, e.g. <code>USD</code>
toCcy	String	Currency to buy, e.g. <code>BTC</code>
rfqAmt	String	RFQ amount
rfqCcy	String	RFQ currency
quotePx	String	Quote price
quoteCcy	String	Quote price unit e.g. <code>USD</code>
quoteFromAmt	String	Quote amount, unit in <code>fromCcy</code>
quoteToAmt	String	Quote amount, unit in <code>toCcy</code>
quoteTime	String	Quotation generation time, Unix timestamp format in milliseconds, e.g. 1597026383085

Parameter	Type	Description
ttlMs	String	The validity period of quotation in milliseconds e.g. <code>10000</code> represents the quotation only valid for 10 seconds

This feature is only available to Bahamas institutional users at the moment.

## BUY/SELL TRADE

### RATE LIMIT: 1 REQUEST PER 5 SECONDS

### RATE LIMIT RULE: USER ID

### PERMISSION: TRADE

### HTTP REQUEST

`POST /api/v5/fiat/buy-sell/trade`

Request Example

### REQUEST PARAMETERS

Parameters	Types	Required	Description
quotId	String	Yes	Quote ID Get from Buy/Sell quote API
side	String	Yes	Side <code>buy</code> : Buy Crypto / Fiat with Fiat <code>sell</code> : Sell Crypto to Crypto / Fiat Should be the same as the Quote request
fromCcy	String	Yes	Currency to sell Should be the same as the Quote request
toCcy	String	Yes	Currency to buy Should be the same as the Quote request
rfqAmt	String	Yes	RFQ amount Should be the same as the Quote request
rfqCcy	String	Yes	RFQ currency Should be the same as the Quote request
paymentMethod	String	Yes	paymentMethod <code>balance</code>
clOrdId	String	Yes	Client Order ID as assigned by the client A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.

Response Example

```
{
  "code": "0",
  "data": [
    {
      "ordId": "1234",
      "clOrdId": "",
      "quoteId": "quoterBTC-USD16461885104612381",
      "side": "buy",
      "fromCcy": "USD",
      "toCcy": "BTC",
      "rfqAmt": "30",
      "rfqCcy": "USD",
      "fillPx": "2932.40104429",
      "fillQuoteCcy": "USD",
      "fillFromAmt": "30",
      "fillToCcy": "BTC"
    }
  ]
}
```

```

        "fillToAmt": "0.01",
        "cTime": "1646188510461"
    },
],
"msg": ""
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ordId	String	Order ID
clOrdId	String	Client Order ID as assigned by the client
quoteId	String	Quote ID
state	String	Trade state processing completed failed
side	String	Side buy: Buy Crypto / Fiat with Fiat sell: Sell Crypto to Crypto / Fiat
fromCcy	String	Currency to sell
toCcy	String	Currency to buy
rfqAmt	String	RFQ amount
rfqCcy	String	RFQ currency
fillPx	String	Filled price based on quote currency
fillQuoteCcy	String	Filled price quote currency e.g. USD
fillFromAmt	String	Sold amount, unit in fromCcy
fillToAmt	String	Bought amount, unit in toCcy
cTime	String	Request time, Unix timestamp format in milliseconds, e.g. 1597026383085

This feature is only available to Bahamas institutional users at the moment.

#### GET BUY/SELL TRADE HISTORY

**RATE LIMIT: 6 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

#### HTTP REQUEST

```
GET /api/v5/coin/buy-sell/history
```

Request Example

#### REQUEST PARAMETERS

Parameters	Types	Required	Description
ordId	String	No	Order ID

Parameters	Types	Required	Description
clOrdId	String	No	Client Order ID as assigned by the client A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
state	String	No	Trade state processing completed failed
begin	String	No	Filter with a begin timestamp. Unix timestamp format in milliseconds, e.g. 1597026383085
end	String	No	Filter with an end timestamp. Unix timestamp format in milliseconds, e.g. 1597026383085
limit	String	No	Number of results per request. The maximum is 100. The default is 100.

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "ordId": "1234",
      "clOrdId": "",
      "quoteId": "quoterBTC-USD16461885104612381",
      "state": "completed",
      "side": "buy",
      "fromCcy": "USD",
      "toCcy": "BTC",
      "rfqAmt": "30",
      "rfqCcy": "USD",
      "fillPx": "2932.40104429",
      "fillQuoteCcy": "USD",
      "fillFromAmt": "30",
      "fillToAmt": "0.01",
      "cTime": "1646188510461",
      "uTime": "1646188510461"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ordId	String	Order ID
clOrdId	String	Client Order ID as assigned by the client
quoteId	String	Quote ID
state	String	Trade state processing completed failed
fromCcy	String	Currency to sell
toCcy	String	Currency to buy
rfqAmt	String	RFQ amount
rfqCcy	String	RFQ currency
fillPx	String	Filled price based on quote currency

Parameter	Type	Description
fillQuoteCcy	String	Filled price quote currency e.g. <code>USD</code>
fillFromAmt	String	Filled amount unit in fromCcy
fillToAmt	String	Filled amount unit in toCcy
cTime	String	Request time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
uTime	String	Updated time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

This feature is only available to Bahamas institutional users at the moment.

## WebSocket

### DEPOSIT INFO CHANNEL

A push notification is triggered when a deposit is initiated or the deposit status changes.

Supports subscriptions for accounts

- If it is a master account subscription, you can receive the push of the deposit info of both the master account and the sub-account.
- If it is a sub-account subscription, only the push of sub-account deposit info you can receive.

### URL PATH

`/ws/v5/business` (required login)

#### Request Example

```
import asyncio
from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/business",
        useServerTime=False
    )
    await ws.start()
    args = [
        {
            "channel": "deposit-info"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>deposit-info</code>
> ccy	String	No	Currency, e.g. <code>BTC</code>

#### Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "deposit-info"
  },
  "connId": "a4d3ae55"
}
```

#### Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \\"subscribe\\", \\"argss\\": [{ \\"channel\\": \\"deposit-info\\\" }]}",
  "connId": "a4d3ae55"
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code> <code>error</code>
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name <code>deposit-info</code>
> ccy	String	No	Currency, e.g. <code>BTC</code>
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

#### Push Data Example

```
{
  "arg": {
    "channel": "deposit-info",
    "uid": "289320****60975104"
  },
  "data": [
    {
      "actualDepBlkConfirm": "0",
      "amt": "1",
      "areaCodeFrom": "",
      "ccy": "USDT",
      "chain": "USDT-TRC20",
      "depId": "88165462",
      "from": "",
      "fromWdId": "",
      "pTime": "1674103661147",
      "state": "0",
      "subAcct": "test",
      "to": "TEhFaapuHa3LY*****8ByNoGnrmegeGMw",
      "ts": "1674103661123",
      "txId": "bc5376817*****dbb0d729f6b",
      "uid": "289320****60975104"
    }
  ]
}
```

#### PUSH DATA PARAMETERS

Parameters	Types	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name deposit-info
> uid	String	User Identifier
> ccy	String	Currency, e.g. BTC
data	Array of objects	Subscribed data
> uid	String	User Identifier of the message producer
> subAcct	String	Sub-account name If the message producer is master account, the parameter will return ""
> pTime	String	Push time, the millisecond format of the Unix timestamp, e.g. 1597026383085
> ccy	String	Currency
> chain	String	Chain name
> amt	String	Deposit amount
> from	String	Deposit account Only the internal OKX account (masked mobile phone number or email address) is returned, not the address on the blockchain.
> areaCodeFrom	String	If from is a phone number, this parameter return area code of the phone number
> to	String	Deposit address
> txId	String	Hash record of the deposit
> ts	String	Time of deposit record is created, Unix timestamp format in milliseconds, e.g. 1655251200000
> state	String	Status of deposit 0: waiting for confirmation 1: deposit credited

Parameters	Types	Description
		<p>2: deposit successful      8: pending due to temporary deposit suspension on this crypto currency      11: match the address blacklist      12: account or deposit is frozen      13: sub-account deposit interception      14: KYC limit</p>
> depId	String	Deposit ID
> fromWdId	String	Internal transfer initiator's withdrawal ID If the deposit comes from internal transfer, this field displays the withdrawal ID of the internal transfer initiator, and will return "" in other cases
> actualDepBlkConfirm	String	The actual amount of blockchain confirmed in a single deposit

#### WITHDRAWAL INFO CHANNEL

A push notification is triggered when a withdrawal is initiated or the withdrawal status changes.

Supports subscriptions for accounts

- If it is a master account subscription, you can receive the push of the withdrawal info of both the master account and the sub-account.
- If it is a sub-account subscription, only the push of sub-account withdrawal info you can receive.

#### URL PATH

/ws/v5/business (required login)

#### Request Example

```
import asyncio
from okx.websocket.WsPrivateAsync import WsPrivateAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPrivateAsync(
        apiKey = "YOUR_API_KEY",
        passphrase = "YOUR_PASSPHRASE",
        secretKey = "YOUR_SECRET_KEY",
        url = "wss://ws.okx.com:8443/ws/v5/business",
        useServerTime=False
    )
    await ws.start()
    args = [
        {
            "channel": "withdrawal-info"
        }
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

asyncio.run(main())
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.

Parameter	Type	Required	Description
op	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>withdrawal-info</code>
> ccy	String	No	Currency, e.g. <code>BTC</code>

#### Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "withdrawal-info"
  },
  "connId": "a4d3ae55"
}
```

#### Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \"subscribe\", \\"argss\\\": [{ \\"channel\\\" : \"withdrawal-info\\\" }]}",
  "connId": "a4d3ae55"
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	Operation <code>subscribe</code> <code>unsubscribe</code> <code>error</code>
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name <code>withdrawal-info</code>
> ccy	String	No	Currency, e.g. <code>BTC</code>
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

#### Push Data Example

```
{
  "arg": {
    "channel": "withdrawal-info",
    "uid": "289320*****0975104"
  },
}
```

```

"data": [
  "addrEx": null,
  "amt": "2",
  "areaCodeFrom": "",
  "areaCodeTo": "",
  "ccy": "USDT",
  "chain": "USDT-TRC20",
  "clientId": "",
  "fee": "0.8",
  "feeCcy": "USDT",
  "from": "",
  "memo": "",
  "nonTradableAsset": false,
  "note": "",
  "pTime": "1674103268578",
  "pmtId": "",
  "state": "0",
  "subAcct": "test",
  "tag": "",
  "to": "TN8CKTQMpWfT*****8KipbJ24ErguhF",
  "toAddrType": "1",
  "ts": "1674103268472",
  "txId": "",
  "uid": "289333*****1101696",
  "wdId": "63754560"
}
]
}

```

#### PUSH DATA PARAMETERS

Parameters	Types	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
> uid	String	User Identifier
> ccy	String	Currency, e.g. <code>BTC</code>
data	Array of objects	Subscribed data
> uid	String	User Identifier of the message producer
> subAcct	String	Sub-account name If the message producer is master account, the parameter will return ""
> pTime	String	Push time, the millisecond format of the Unix timestamp, e.g. <code>1597026383085</code>
> ccy	String	Currency
> chain	String	Chain name, e.g. <code>USDT-ERC20</code> , <code>USDT-TRC20</code>
> nonTradableAsset	String	Whether it is a non-tradable asset or not <code>true</code> : non-tradable asset, <code>false</code> : tradable asset
> amt	String	Withdrawal amount
> ts	String	Time the withdrawal request was submitted, Unix timestamp format in milliseconds, e.g. <code>1655251200000</code> .
> from	String	Withdrawal account It can be <code>email</code> / <code>phone</code> / <code>sub-account name</code>
> areaCodeFrom	String	Area code for the phone number If <code>from</code> is a phone number, this parameter returns the area code for the phone number
> to	String	Receiving address

Parameters	Types	Description
> areaCodeTo	String	Area code for the phone number If <code>to</code> is a phone number, this parameter returns the area code for the phone number
> toAddrType	String	Address type ①: wallet address, email, phone, or login account name ②: UID
> tag	String	Some currencies require a tag for withdrawals
> pmtd	String	Some currencies require a payment ID for withdrawals
> memo	String	Some currencies require this parameter for withdrawals
> addrEx	Object	Withdrawal address attachment, e.g. <code>TONCOIN</code> attached tag name is comment, the return will be <code>{'comment':'123456'}</code>
> txld	String	Hash record of the withdrawal This parameter will return "" for internal transfers.
> fee	String	Withdrawal fee amount
> feeCcy	String	Withdrawal fee currency, e.g. <code>USDT</code>
Status of withdrawal		
<ul style="list-style-type: none"> <li>• Stage 1 : Pending withdrawal ⑯: Pending response from Travel Rule vendor ⑯: Waiting transfer ⑯: Waiting withdrawal ④/⑤/⑥/⑧/⑨/⑫: Waiting manual review ⑦: Approved</li> </ul>		
> state	String	<ul style="list-style-type: none"> <li>• Stage 2 : Withdrawal in progress (Applicable to on-chain withdrawals, internal transfers do not have this stage) ⑯: Broadcasting your transaction to chain ⑯: Pending transaction validation ⑯: Due to local laws and regulations, your withdrawal may take up to 24 hours to arrive ⑯: Canceling</li> <li>• Final stage ⑯: Canceled ⑯: Failed ⑯: Success</li> </ul>
> wdld	String	Withdrawal ID
> clientId	String	Client-supplied ID
> note	String	Withdrawal note

## Sub-account

The API endpoints of `sub-account` require authentication.

## GET SUB-ACCOUNT LIST

Applies to master accounts only

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

```
GET /api/v5/users/subaccount/list
```

Request sample

```
import okx.SubAccount as SubAccount

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

subAccountAPI = SubAccount.SubAccountAPI(apikey, secretkey, passphrase, False, flag)

# Get sub-account list
result = subAccountAPI.get_subaccount_list()
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
enable	String	No	Sub-account status [true]: Normal [false]: Frozen
subAcct	String	No	Sub-account name
after	String	No	Query the data earlier than the requested subaccount creation timestamp, the value should be a Unix timestamp in millisecond format. e.g. 1597026383085
before	String	No	Query the data newer than the requested subaccount creation timestamp, the value should be a Unix timestamp in millisecond format. e.g. 1597026383085
limit	String	No	Number of results per request. The maximum is 100. The default is 100.

Returned results

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "canTransOut": false,
      "enable": true,
      "frozenFunc": [
      ],
      "gAuth": false,
      "label": "D456DDDLx",
      "mobile": "",
      "subAcct": "D456DDDL",
      "ts": "1659334756000",
      "type": "1",
      "uid": "3400*****7413",
      "subAcctLv": "1",
      "firstLvSubAcct": "D456DDDL",
      "ifDma": false
    }
  ]
}
```

]

}

## RESPONSE PARAMETERS

Parameter name	Type	Description
type	String	Sub-account type 1: Standard sub-account 2: Managed trading sub-account 5: Custody trading sub-account - Copper 9: Managed trading sub-account - Copper 12: Custody trading sub-account - Komainu
enable	Boolean	Sub-account status true: Normal false: Frozen (global)
subAcct	String	Sub-account name
uid	String	Sub-account uid
label	String	Sub-account note
mobile	String	Mobile number that linked with the sub-account.
gAuth	Boolean	If the sub-account switches on the Google Authenticator for login authentication. true: On false: Off
frozenFunc	Array of strings	Frozen functions trading convert transfer withdrawal deposit flexible_loan
canTransOut	Boolean	Whether the sub-account has the right to transfer out. true: can transfer out false: cannot transfer out
ts	String	Sub-account creation time, Unix timestamp in millisecond format. e.g. 1597026383085
subAcctLv	String	Sub-account level 1: First level sub-account 2: Second level sub-account.
firstLvSubAcct	String	The first level sub-account. For subAcctLv: 1, firstLvSubAcct is equal to subAcct For subAcctLv: 2, subAcct belongs to firstLvSubAcct.
ifDma	Boolean	Whether it is dma broker sub-account. true: Dma broker sub-account false: It is not dma broker sub-account.

## CREATE SUB-ACCOUNT

Applies to master accounts only and master accounts API Key must be linked to IP addresses.

**RATE LIMIT: 1 REQUEST PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

## Request sample

```

import okx.SubAccount as SubAccount

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

subAccountAPI = SubAccount.SubAccountAPI(apikey, secretkey, passphrase, False, flag)

# Reset the API Key of a sub-account
result = subAccountAPI.reset_subaccount_apikey(
    subAcct="hahawang1",
    apiKey="",
    ip=""
)
print(result)

```

## REQUEST PARAMETERS

Parameter name	Type	Required	Description
subAcct	String	Yes	Sub-account name
type	String	Yes	Sub-account type <input checked="" type="radio"/> 1: Standard sub-account <input type="radio"/> 5: Custody trading sub-account - Copper <input type="radio"/> 12: Custody trading sub-account - Komainu
label	String	No	Sub-account notes. 6-32 letters (case sensitive), numbers or special characters like *.
pwd	String	Conditional	Sub-account login password, it is required for KYB users only. Your password must contain: 8 - 32 characters long. 1 lowercase character (a-z). 1 uppercase character (A-Z). 1 number. 1 special character e.g. ! @ # \$ %

## Returned results

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "label": "123456",
      "subAcct": "subAccount002",
      "ts": "1744875304520",
      "uid": "698827017768230914"
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter name	Type	Description
subAcct	String	Sub-account name
label	String	Sub-account notes
uid	String	Sub-account ID

Parameter name	Type	Description
ts	String	Creation time

#### CREATE AN API KEY FOR A SUB-ACCOUNT

Applies to master accounts only and master accounts API Key must be linked to IP addresses.

**RATE LIMIT: 1 REQUEST PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

`POST /api/v5/users/subaccount/apikey`

Request sample

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
subAcct	String	Yes	Sub-account name, supports 6 to 20 characters that include numbers and letters (case sensitive, space symbol is not supported).
label	String	Yes	API Key note
passphrase	String	Yes	API Key password, supports 8 to 32 alphanumeric characters containing at least 1 number, 1 uppercase letter, 1 lowercase letter and 1 special character.
perm	String	No	API Key permissions <input type="checkbox"/> <code>read_only</code> : Read only <input type="checkbox"/> <code>trade</code> : Trade
ip	String	No	Link IP addresses, separate with commas if more than one. Support up to 20 addresses. <b>For security reasons, it is recommended to bind IP addresses.</b> <b>API keys with trading or withdrawal permissions that are not bound to IPs will expire after 14 days of inactivity. (API keys in demo trading will not be deleted.)</b>

Returned result

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "subAcct": "test-1",
      "label": "v5",
      "apiKey": "*****",
      "secretKey": "*****",
      "passphrase": "*****",
      "perm": "read_only,trade",
      "ip": "1.1.1.1,2.2.2.2",
      "ts": "1597026383085"
    }
  ]
}
```

**RESPONSE PARAMETERS**

Parameter name	Type	Description
subAcct	String	Sub-account name
label	String	API Key note
apiKey	String	API public key
secretKey	String	API private key

Parameter name	Type	Description
passphrase	String	API Key password
perm	String	API Key access read_only : Read only trade : Trade
ip	String	IP address that linked with API Key
ts	String	Creation time

#### QUERY THE API KEY OF A SUB-ACCOUNT

Applies to master accounts only

**RATE LIMIT: 20 REQUEST PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

GET /api/v5/users/subaccount/apikey

Request sample

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
subAcct	String	Yes	Sub-account name
apiKey	String	No	API public key

Returned results

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "label": "v5",
      "apiKey": "*****",
      "perm": "read_only,trade",
      "ip": "1.1.1.1,2.2.2.2",
      "ts": "1597026383085"
    },
    {
      "label": "v5.1",
      "apiKey": "*****",
      "perm": "read_only",
      "ip": "1.1.1.1,2.2.2.2",
      "ts": "1597026383085"
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter name	Type	Description
label	String	API Key note
apiKey	String	API public key
perm	String	API Key access read_only: Read only; trade: Trade

Parameter name	Type	Description
ip	String	IP address that linked with API Key
ts	String	Creation time

#### RESET THE API KEY OF A SUB-ACCOUNT

Applies to master accounts only and master accounts API Key must be linked to IP addresses.

**RATE LIMIT: 1 REQUEST PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

POST /api/v5/users/subaccount/modify-apikey

Request sample

```
import okx.SubAccount as SubAccount

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

subAccountAPI = SubAccount.SubAccountAPI(apikey, secretkey, passphrase, False, flag)

# Reset the API Key of a sub-account
result = subAccountAPI.reset_subaccount_apikey(
    subAcct="hahawang1",
    apiKey="",
    ip=""
)
print(result)
```

#### REQUEST PARAMETERS

Parameter name	Type	Required	Description
subAcct	String	Yes	Sub-account name
apiKey	String	Yes	Sub-account APIKey
label	String	No	Sub-account API Key label. The label will be reset if this is passed through.
perm	String	No	Sub-account API Key permissions <input checked="" type="checkbox"/> <b>read_only</b> : Read <input checked="" type="checkbox"/> <b>trade</b> : Trade Separate with commas if more than one. The permission will be reset if this is passed through.
ip	String	No	Sub-account API Key linked IP addresses, separate with commas if more than one. Support up to 20 IP addresses. The IP will be reset if this is passed through. If <input checked="" type="checkbox"/> is set to "", then no IP addresses is linked to the APIKey.

Returned results

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "subAcct": "yongxu",
      "ip": "127.0.0.1"
    }
  ]
}
```

```

    "label": "v5",
    "apiKey": "*****",
    "perm": "read,trade",
    "ip": "1.1.1.1",
    "ts": "1597026383085"
  }]
}

```

#### RESPONSE PARAMETERS

Parameter name	Type	Description
subAcct	String	Sub-account name
apiKey	String	Sub-account API public key
label	String	Sub-account API Key label
perm	String	Sub-account API Key permissions <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">read_only</div> : Read <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">trade</div> : Trade
ip	String	Sub-account API Key IP addresses that linked with API Key
ts	String	Creation time

#### DELETE THE API KEY OF SUB-ACCOUNTS

Applies to master accounts only and master accounts API Key must be linked to IP addresses.

**RATE LIMIT: 1 REQUEST PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

```
POST /api/v5/users/subaccount/delete-apikey
```

Request sample

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
subAcct	String	Yes	Sub-account name
apiKey	String	Yes	API public key

Returned results

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "subAcct": "test00001"
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter name	Type	Description
subAcct	String	Sub-account name

#### GET SUB-ACCOUNT TRADING BALANCE

Query detailed balance info of Trading Account of a sub-account via the master account (applies to master accounts only)

**RATE LIMIT: 6 REQUESTS PER 2 SECONDS**

#### RATE LIMIT RULE: USER ID

## PERMISSION: READ

GET /api/v5/account/{subaccount}/balances

### Request sample

```
import okx.SubAccount as SubAccount

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

subAccountAPI = SubAccount.SubAccountAPI(apikey, secretkey, passphrase, False, flag)

# Get sub-account trading balance
result = subAccountAPI.get_account_balance(
    subAcct="hahawang1"
)
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
subAcct	String	Yes	Sub-account name

## Returned result

```

        "liab": "0",
        "maxLoan": "1015.000000000001",
        "maxSpotInUse": "",
        "mgnRatio": "",
        "mmr": "",
        "notionalLever": "",
        "openAvgPx": "",
        "ordFrozen": "0",
        "rewardBal": "",
        "smtSyncEq": "0",
        "spotBal": "",
        "spotCopyTradingEq": "0",
        "spotInUseAmt": "",
        "spotIsoBal": "0",
        "spotUp1": "",
        "spotUp1Ratio": "",
        "stgyEq": "0",
        "totalPnl": "",
        "totalPnlRatio": "",
        "twap": "0",
        "uTime": "1663854334734",
        "upl": "0",
        "uplLiab": "0"
    },
    ],
    "imr": "0",
    "isoEq": "0",
    "mgnRatio": "",
    "mmr": "0",
    "notionalUsd": "0",
    "notionalUsdForBorrow": "0",
    "notionalUsdForFutures": "0",
    "notionalUsdForOption": "0",
    "notionalUsdForSwap": "0",
    "ordFroz": "0",
    "totalEq": "101.4675200000001",
    "uTime": "1739332269934",
    "upl": "0"
},
],
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameters	Types	Description
uTime	String	Update time of account information, millisecond format of Unix timestamp, e.g. <code>1597026383085</code>
totalEq	String	The total amount of equity in <code>USD</code>
isoEq	String	Isolated margin equity in <code>USD</code> Applicable to <code>Futures mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
adjEq	String	Adjusted / Effective equity in <code>USD</code> The net fiat value of the assets in the account that can provide margins for spot, expiry futures, perpetual futures and options under the cross-margin mode. In multi-ccy or PM mode, the asset and margin requirement will all be converted to USD value to process the order check or liquidation. Due to the volatility of each currency market, our platform calculates the actual USD value of each currency based on discount rates to balance market risks. Applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> and <code>Portfolio margin</code>
availEq	String	Account level available equity, excluding currencies that are restricted due to the collateralized borrowing limit. Applicable to <code>Multi-currency margin</code> / <code>Portfolio margin</code>
ordFroz	String	Cross margin frozen for pending orders in <code>USD</code> Only applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
imr	String	Initial margin requirement in <code>USD</code> The sum of initial margins of all open positions and pending orders under cross-margin mode in <code>USD</code> .

Parameters	Types	Description
Applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>		
mmr	String	<p>Maintenance margin requirement in <code>USD</code></p> <p>The sum of maintenance margins of all open positions and pending orders under cross-margin mode in <code>USD</code>.</p> <p>Applicable to <code>Spot mode</code>/<code>Multi-currency margin</code>/<code>Portfolio margin</code></p>
Only applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code> . It is "" for other margin modes.		
mgnRatio	String	<p>Maintenance margin ratio in <code>USD</code></p> <p>Applicable to <code>Spot mode</code>/<code>Multi-currency margin</code>/<code>Portfolio margin</code></p>
notionalUsd	String	<p>Notional value of positions in <code>USD</code></p> <p>Applicable to <code>Spot mode</code>/<code>Multi-currency margin</code>/<code>Portfolio margin</code></p>
notionalUsdForBorrow	String	<p>Notional value for <code>Borrow</code> in USD</p> <p>Applicable to <code>Spot mode</code>/<code>Multi-currency margin</code>/<code>Portfolio margin</code></p>
notionalUsdForSwap	String	<p>Notional value of positions for <code>Perpetual Futures</code> in USD</p> <p>Applicable to <code>Multi-currency margin</code>/<code>Portfolio margin</code></p>
notionalUsdForFutures	String	<p>Notional value of positions for <code>Expiry Futures</code> in USD</p> <p>Applicable to <code>Multi-currency margin</code>/<code>Portfolio margin</code></p>
notionalUsdForOption	String	<p>Notional value of positions for <code>Option</code> in USD</p> <p>Applicable to <code>Spot mode</code>/<code>Multi-currency margin</code>/<code>Portfolio margin</code></p>
upl	String	<p>Cross-margin info of unrealized profit and loss at the account level in <code>USD</code></p> <p>Applicable to <code>Multi-currency margin</code>/<code>Portfolio margin</code></p>
delta	String	Delta (USD)
deltaLever	String	<p>Delta neutral strategy account level delta leverage</p> <p><math>\text{deltaLever} = \text{delta} / \text{totalEq}</math></p>
deltaNeutralStatus	String	<p>Delta risk status</p> <p><code>0</code>: normal</p> <p><code>1</code>: transfer restricted</p> <p><code>2</code>: delta reducing - cancel all pending orders if delta is greater than 5000 USD, only one delta reducing order allowed per index (spot, futures, swap)</p>
details	Array of objects	Detailed asset information in all currencies
> ccy	String	Currency
> eq	String	Equity of currency
> cashBal	String	Cash balance
> uTime	String	Update time of currency balance information, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> isoEq	String	<p>Isolated margin equity of currency</p> <p>Applicable to <code>Futures mode</code>/<code>Multi-currency margin</code>/<code>Portfolio margin</code></p>
> availEq	String	<p>Available equity of currency</p> <p>Applicable to <code>Futures mode</code>/<code>Multi-currency margin</code>/<code>Portfolio margin</code></p>
> disEq	String	Discount equity of currency in <code>USD</code> .
> fixedBal	String	Frozen balance for <code>Dip Sniper</code> and <code>Peak Sniper</code>

Parameters	Types	Description
> availBal	String	Available balance of currency
> frozenBal	String	Frozen balance of currency
> ordFrozen	String	Margin frozen for open orders Applicable to <code>Spot mode</code> / <code>Futures mode</code> / <code>Multi-currency margin</code>
> liab	String	Liabilities of currency It is a positive value, e.g. <code>21625.64</code> Applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
> upl	String	The sum of the unrealized profit & loss of all margin and derivatives positions of currency. Applicable to <code>Futures mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
> uplLiab	String	Liabilities due to Unrealized loss of currency Applicable to <code>Multi-currency margin</code> / <code>Portfolio margin</code>
> crossLiab	String	Cross liabilities of currency Applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
> rewardBal	String	Trial fund balance
> isoLiab	String	Isolated liabilities of currency Applicable to <code>Multi-currency margin</code> / <code>Portfolio margin</code>
> mgnRatio	String	Cross Maintenance margin ratio of currency The index for measuring the risk of a certain asset in the account. Applicable to <code>Futures mode</code> and when there is cross position
> imr	String	Cross initial margin requirement at the currency level Applicable to <code>Futures mode</code> and when there is cross position
> mmr	String	Cross maintenance margin requirement at the currency level Applicable to <code>Futures mode</code> and when there is cross position
> interest	String	Accrued interest of currency It is a positive value, e.g. <code>9.01</code> Applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
> twap	String	Risk indicator of forced repayment Divided into multiple levels from 0 to 5, the larger the number, the more likely the forced repayment will be triggered. Applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
> frpType	String	Forced repayment (FRP) type ①: no FRP ②: user based FRP ③: platform based FRP  Return ①/② when twap is $\geq 1$ , applicable to <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
> maxLoan	String	Max loan of currency Applicable to <code>cross</code> of <code>Spot mode</code> / <code>Multi-currency margin</code> / <code>Portfolio margin</code>
> eqUsd	String	Equity in <code>USD</code> of currency
> borrowFroz	String	Potential borrowing IMR of currency in <code>USD</code> Applicable to <code>Multi-currency margin</code> / <code>Portfolio margin</code> . It is "" for other margin modes.
> notionalLever	String	Leverage of currency Applicable to <code>Futures mode</code>

Parameters	Types	Description
> stgyEq	String	Strategy equity
> isoUpI	String	Isolated unrealized profit and loss of currency Applicable to <a href="#">Futures mode</a> / <a href="#">Multi-currency margin</a> / <a href="#">Portfolio margin</a>
> spotInUseAmt	String	Spot in use amount Applicable to <a href="#">Portfolio margin</a>
> clSpotInUseAmt	String	User-defined spot risk offset amount Applicable to <a href="#">Portfolio margin</a>
> maxSpotInUse	String	Max possible spot risk offset amount Applicable to <a href="#">Portfolio margin</a>
> spotIsoBal	String	Spot isolated balance Applicable to <a href="#">copy trading</a> Applicable to <a href="#">Spot mode</a> / <a href="#">Futures mode</a> .
> smtSyncEq	String	Smart sync equity The default is "0", only applicable to copy trader.
> spotCopyTradingEq	String	Spot smart sync equity. The default is "0", only applicable to copy trader.
> spotBal	String	Spot balance. The unit is currency, e.g. BTC. More details
> openAvgPx	String	Spot average cost price. The unit is USD. More details
> accAvgPx	String	Spot accumulated cost price. The unit is USD. More details
> spotUpI	String	Spot unrealized profit and loss. The unit is USD. More details
> spotUpIRatio	String	Spot unrealized profit and loss ratio. More details
> totalPnl	String	Spot accumulated profit and loss. The unit is USD. More details
> totalPnlRatio	String	Spot accumulated profit and loss ratio. More details
> colRes	String	Platform level collateral restriction status <span style="border: 1px solid #ccc; border-radius: 50%; padding: 2px 5px;">0</span> : The restriction is not enabled. <span style="border: 1px solid #ccc; border-radius: 50%; padding: 2px 5px;">1</span> : The restriction is not enabled. But the crypto is close to the platform's collateral limit. <span style="border: 1px solid #ccc; border-radius: 50%; padding: 2px 5px;">2</span> : The restriction is enabled. This crypto can't be used as margin for your new orders. This may result in failed orders. But it will still be included in the account's adjusted equity and doesn't impact margin ratio. Refer to <a href="#">Introduction to the platform collateralized borrowing limit</a> for more details.
> colBorrAutoConversion	String	Risk indicator of auto conversion. Divided into multiple levels from 1-5, the larger the number, the more likely the repayment will be triggered. The default will be 0, indicating there is no risk currently. 5 means this user is undergoing auto conversion now, 4 means this user will undergo auto conversion soon whereas 1/2/3 indicates there is a risk for auto conversion. Applicable to <a href="#">Spot mode</a> / <a href="#">Futures mode</a> / <a href="#">Multi-currency margin</a> / <a href="#">Portfolio margin</a> When the total liability for each crypto set as collateral exceeds a certain percentage of the platform's total limit, the auto-conversion mechanism may be triggered. This may result in the automatic sale of excess collateral crypto if you've set this crypto as collateral and have large borrowings. To lower this risk, consider reducing your use of the crypto as collateral or reducing your liabilities. Refer to <a href="#">Introduction to the platform collateralized borrowing limit</a> for more details.
> collateralRestrict	Boolean	<a href="#">Platform level collateralized borrow restriction</a> <span style="border: 1px solid #ccc; border-radius: 50%; padding: 2px 5px;">true</span> <span style="border: 1px solid #ccc; border-radius: 50%; padding: 2px 5px;">false</span> (deprecated, use colRes instead)
> collateralEnabled	Boolean	<span style="border: 1px solid #ccc; border-radius: 50%; padding: 2px 5px;">true</span> : Collateral enabled <span style="border: 1px solid #ccc; border-radius: 50%; padding: 2px 5px;">false</span> : Collateral disabled Applicable to <a href="#">Multi-currency margin</a>

Parameters	Types	Description
> autoLendStatus	String	<p>Auto lend status</p> <p><code>unsupported</code>: auto lend is not supported by this currency</p> <p><code>off</code>: auto lend is supported but turned off</p> <p><code>pending</code>: auto lend is turned on but pending matching</p> <p><code>active</code>: auto lend is turned on and matched</p>
> autoLendMtAmt	String	<p>Auto lend currency matched amount</p> <p>Return "0" when autoLendStatus is <code>unsupported/off/pending</code>. Return matched amount when autoLendStatus is <code>active</code></p>

"" will be returned for inapplicable fields with the current account level.

#### GET SUB-ACCOUNT FUNDING BALANCE

Query detailed balance info of Funding Account of a sub-account via the master account (applies to master accounts only)

**RATE LIMIT: 6 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

#### HTTP REQUEST

`GET /api/v5/asset/subaccount/balances`

Request sample

```
import okx.SubAccount as SubAccount

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

subAccountAPI = SubAccount.SubAccountAPI(apikey, secretkey, passphrase, False, flag)

# Get sub-account funding balance
result = subAccountAPI.get_funding_balance(
    subAcct="hahawang1"
)
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
subAcct	String	Yes	Sub-account name
ccy	String	No	Single currency or multiple currencies (no more than 20) separated with comma, e.g. <code>BTC</code> or <code>BTC,ETH</code> .

Returned result

```
{
    "code": "0",
    "msg": "",
    "data": [
        {
            "availBal": "37.11827078",
            "bal": "37.11827078",
            "ccy": "ETH",
            "frozenBal": "0"
        }
    ]
}
```

]

}

## RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency
bal	String	Balance
frozenBal	String	Frozen balance
availBal	String	Available balance

## GET SUB-ACCOUNT MAXIMUM WITHDRAWALS

Retrieve the maximum withdrawal information of a sub-account via the master account (applies to master accounts only). If no currency is specified, the transferable amount of all owned currencies will be returned.

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

### HTTP REQUEST

GET /api/v5/account/subaccount/max-withdrawal

Request Example

## REQUEST PARAMETERS

Parameter	Type	Required	Description
subAcct	String	Yes	Sub-account name
ccy	String	No	Single currency or multiple currencies (no more than 20) separated with comma, e.g. <code>BTC</code> or <code>BTC,ETH</code> .

Response Example

```
{
  "code": "0",
  "data": [
    {
      "ccy": "BTC",
      "maxWd": "3",
      "maxWdEx": "",
      "spotOffsetMaxWd": "3",
      "spotOffsetMaxWdEx": ""
    },
    {
      "ccy": "ETH",
      "maxWd": "15",
      "maxWdEx": "",
      "spotOffsetMaxWd": "15",
      "spotOffsetMaxWdEx": ""
    },
    {
      "ccy": "USDT",
      "maxWd": "10600",
      "maxWdEx": "",
      "spotOffsetMaxWd": "10600",
      "spotOffsetMaxWdEx": ""
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency
maxWd	String	Max withdrawal (excluding borrowed assets under <a href="#">Multi-currency margin</a> )
maxWdEx	String	Max withdrawal (including borrowed assets under <a href="#">Multi-currency margin</a> )
spotOffsetMaxWd	String	Max withdrawal under Spot-Derivatives risk offset mode (excluding borrowed assets under <a href="#">Portfolio margin</a> ) Applicable to <a href="#">Portfolio margin</a>
spotOffsetMaxWdEx	String	Max withdrawal under Spot-Derivatives risk offset mode (including borrowed assets under <a href="#">Portfolio margin</a> ) Applicable to <a href="#">Portfolio margin</a>

#### GET HISTORY OF SUB-ACCOUNT TRANSFER

This endpoint is only available for master accounts. Transfer records are available from September 28, 2022 onwards.

**RATE LIMIT: 6 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/asset/subaccount/bills`

Request sample

```
import okx.SubAccount as SubAccount

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

subAccountAPI = SubAccount.SubAccountAPI(apikey, secretkey, passphrase, False, flag)

# Get history of sub-account transfer
result = subAccountAPI.bills()
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	No	Currency, such as BTC
type	String	No	Transfer type <input checked="" type="radio"/> 0: Transfers from master account to sub-account <input type="radio"/> 1: Transfers from sub-account to master account.
subAcct	String	No	Sub-account name
after	String	No	Query the data prior to the requested bill ID creation time (exclude), the value should be a Unix timestamp in millisecond format. e.g. <a href="#">1597026383085</a>
before	String	No	Query the data after the requested bill ID creation time (exclude), the value should be a Unix timestamp in millisecond format. e.g. <a href="#">1597026383085</a>
limit	String	No	Number of results per request. The maximum is 100. The default is 100.

Returned results

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "amt": "1.1",
      "billId": "89887685",
      "ccy": "USDT",
      "subAcct": "hahatest1",
      "ts": "1712560959000",
      "type": "0"
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter name	Type	Description
billId	String	Bill ID
ccy	String	Transfer currency
amt	String	Transfer amount
type	String	Bill type
subAcct	String	Sub-account name
ts	String	Bill ID creation time, Unix timestamp in millisecond format, e.g. <code>1597026383085</code>

## GET HISTORY OF MANAGED SUB-ACCOUNT TRANSFER

Only applicable to the trading team's master account to getting transfer records of managed sub accounts entrusted to oneself.

**RATE LIMIT: 6 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/asset/subaccount/managed-subaccount-bills`

Request sample

## REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	No	Currency, e.g <code>BTC</code>
type	String	No	Transfer type <code>0</code> : Transfers from master account to sub-account <code>1</code> : Transfers from sub-account to master account
subAcct	String	No	Sub-account name
subUid	String	No	Sub-account UID
after	String	No	Query the data prior to the requested bill ID creation time (exclude), Unix timestamp in millisecond format, e.g. <code>1597026383085</code>
before	String	No	Query the data after the requested bill ID creation time (exclude), Unix timestamp in millisecond format, e.g. <code>597026383085</code>
limit	String	No	Number of results per request. The maximum is 100. The default is 100.

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "billId": "12344",
      "type": "1",
      "ccy": "BTC",
      "amt": "2",
      "subAcct": "test-1",
      "subUid": "xxxxxx",
      "ts": "1597026383085"
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter name	Type	Description
billId	String	Bill ID
ccy	String	Transfer currency
amt	String	Transfer amount
type	String	Bill type
subAcct	String	Sub-account name
subUid	String	Sub-account UID
ts	String	Bill ID creation time, Unix timestamp in millisecond format, e.g. 1597026383085

## MASTER ACCOUNTS MANAGE THE TRANSFERS BETWEEN SUB-ACCOUNTS

Applies to master accounts only.

Only API keys with **Trade** privilege can call this endpoint.

**RATE LIMIT: 1 REQUEST PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

## HTTP REQUEST

`POST /api/v5/asset/subaccount/transfer`

Request sample

```
import okx.SubAccount as SubAccount

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

subAccountAPI = SubAccount.SubAccountAPI(apikey, secretkey, passphrase, False, flag)

# Master accounts manage the transfers between sub-accounts
result = subAccountAPI.subAccount_transfer(
  ccy="USDT",
  amt="10",
  froms="6",
  to="6",
  fromSubAccount="test-1",
  toSubAccount="test-2"
)
```

```
)  
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	Yes	Currency
amt	String	Yes	Transfer amount
from	String	Yes	Account type of transfer from sub-account 6: Funding Account 18: Trading account
to	String	Yes	Account type of transfer to sub-account 6: Funding Account 18: Trading account
fromSubAccount	String	Yes	Sub-account name of the account that transfers funds out.
toSubAccount	String	Yes	Sub-account name of the account that transfers funds in.
loanTrans	Boolean	No	Whether or not borrowed coins can be transferred out under <a href="#">Multi-currency margin</a> / <a href="#">Portfolio margin</a> The default is <code>false</code>
omitPosRisk	String	No	Ignore position risk Default is <code>false</code> Applicable to <a href="#">Portfolio margin</a>

## Returned results

```
{  
  "code": "0",  
  "msg": "",  
  "data": [  
    {  
      "transId": "12345",  
    }  
  ]  
}
```

## RESPONSE PARAMETERS

Parameter name	Type	Description
transId	String	Transfer ID

## SET PERMISSION OF TRANSFER OUT

Set permission of transfer out for sub-account (only applicable to master account API key). Sub-account can transfer out to master account by default.

## RATE LIMIT: 1 REQUEST PER SECOND

## RATE LIMIT RULE: USER ID

## PERMISSION: TRADE

## HTTP REQUEST

```
POST /api/v5/users/subaccount/set-transfer-out
```

## Request Example

```
import okx.SubAccount as SubAccount  
  
# API initialization
```

```

apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

subAccountAPI = SubAccount.SubAccountAPI(apikey, secretkey, passphrase, False, flag)

# Set permission of transfer out for sub-account
result = subAccountAPI.set_permission_transfer_out(
    subAcct="hahawang1",
    canTransOut=False
)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
subAcct	String	Yes	Name of the sub-account. Single sub-account or multiple sub-account (no more than 20) separated with comma.
canTransOut	Boolean	No	Whether the sub-account has the right to transfer out. The default is <code>true</code> . <code>false</code> : cannot transfer out <code>true</code> : can transfer out

## Returned result

```

{
  "code": "0",
  "msg": "",
  "data": [
    {
      "subAcct": "Test001",
      "canTransOut": true
    },
    {
      "subAcct": "Test002",
      "canTransOut": true
    }
  ]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
subAcct	String	Name of the sub-account
canTransOut	Boolean	Whether the sub-account has the right to transfer out. <code>false</code> : cannot transfer out <code>true</code> : can transfer out

## GET CUSTODY TRADING SUB-ACCOUNT LIST

The trading team uses this interface to view the list of sub-accounts currently under escrow

**RATE LIMIT: 1 REQUEST PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

## HTTP REQUEST

`GET /api/v5/users/entrust-subaccount-list`

Request sample

```

import okx.SubAccount as SubAccount

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "1" # Production trading: 0, Demo trading: 1

subAccountAPI = SubAccount.SubAccountAPI(apikey, secretkey, passphrase, False, flag)

# Get custody trading sub-account list
result = subAccountAPI.get_entrust_subaccount_list()
print(result)

```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
subAcct	String	No	Sub-account name

Returned results

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "subAcct": "test-1"
    },
    {
      "subAcct": "test-2"
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter name	Type	Description
subAcct	String	Sub-account name

# Financial Product

## On-chain earn

Only the assets in the funding account can be used for purchase. More details

**GET / OFFERS**

**RATE LIMIT: 3 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

```
GET /api/v5/finance/staking-defi/offers
```

Request Example

```

import okx.Finance.StakingDefi as StakingDefi

```

```

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

StakingAPI = StakingDefi.StakingDefiAPI(apikey, secretkey, passphrase, False, flag)

result = StakingAPI.get_offers(ccy="USDT")
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
productId	String	No	Product ID
protocolType	String	No	Protocol type defi: on-chain earn
ccy	String	No	Investment currency, e.g. <a href="#">BTC</a>

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "ccy": "DOT",
      "productId": "101",
      "protocol": "Polkadot",
      "protocolType": "defi",
      "term": "0",
      "apy": "0.1767",
      "earlyRedeem": false,
      "state": "purchasable",
      "investData": [
        {
          "bal": "0",
          "ccy": "DOT",
          "maxAmt": "0",
          "minAmt": "2"
        }
      ],
      "earningData": [
        {
          "ccy": "DOT",
          "earningType": "0"
        }
      ],
      "fastRedemptionDailyLimit": "",
      "redeemPeriod": [
        "28D",
        "28D"
      ]
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency type, e.g. <a href="#">BTC</a>
productId	String	Product ID

Parameter	Type	Description
protocol	String	Protocol
protocolType	String	Protocol type defi: on-chain earn
term	String	Protocol term It will return the days of fixed term and will return <code>0</code> for flexible product
apy	String	Estimated annualization If the annualization is 7% , this field is 0.07
earlyRedeem	Boolean	Whether the protocol supports early redemption
investData	Array of objects	Current target currency information available for investment
> ccy	String	Investment currency, e.g. <code>BTC</code>
> bal	String	Available balance to invest
> minAmt	String	Minimum subscription amount
> maxAmt	String	Maximum available subscription amount
earningData	Array of objects	Earning data
> ccy	String	Earning currency, e.g. <code>BTC</code>
> earningType	String	Earning type <code>0</code> : Estimated earning <code>1</code> : Cumulative earning
state	String	Product state <code>purchasable</code> : Purchasable <code>sold_out</code> : Sold out <code>Stop</code> : Suspension of subscription
redeemPeriod	Array of strings	Redemption Period, format in [min time,max time] <code>H</code> : Hour, <code>D</code> : Day e.g. <code>["1H","24H"]</code> represents redemption period is between 1 Hour and 24 Hours. <code>["14D","14D"]</code> represents redemption period is 14 days.
fastRedemptionDailyLimit	String	Fast redemption daily limit If fast redemption is not supported, it will return <code>0</code> .

## POST / PURCHASE

**RATE LIMIT: 2 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

## HTTP REQUEST

POST /api/v5/finance/staking-defi/purchase

### Request Example

```
import okx.Finance.StakingDefi as StakingDefi

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"
```

```

flag = "0" # Production trading:0 , demo trading:1

StakingAPI = StakingDefi.StakingDefiAPI(apikey, secretkey, passphrase, False, flag)

result = StakingAPI.purchase(
    productId = "4005",
    investData = [
        {"ccy": "USDT",
        "amt": "100"
    }]
)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
productId	String	Yes	Product ID
investData	Array of objects	Yes	Investment data
> ccy	String	Yes	Investment currency, e.g. <code>BTC</code>
> amt	String	Yes	Investment amount
term	String	Conditional	Investment term Investment term must be specified for fixed-term product
tag	String	No	Order tag A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters.

## Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ordId": "754147",
      "tag": ""
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ordId	String	Order ID
tag	String	Order tag

## POST / REDEEM

**RATE LIMIT: 2 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

```
POST /api/v5/finance/staking-defi/redeem
```

## Request Example

```
import okx.Finance.StakingDefi as StakingDefi
```

```

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

StakingAPI = StakingDefi.StakingDefiAPI(apikey, secretkey, passphrase, False, flag)

result = StakingAPI.redeem(
    ordId = "1234",
    protocolType = "defi"
)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
ordId	String	Yes	Order ID
protocolType	String	Yes	Protocol type defi: on-chain earn
allowEarlyRedeem	Boolean	No	Whether allows early redemption Default is false

## Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ordId": "754147",
      "tag": ""
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ordId	String	Order ID
tag	String	Order tag

## POST / CANCEL PURCHASES/REDEMPTIONS

After cancelling, returning funds will go to the funding account.

## RATE LIMIT: 2 REQUESTS PER SECOND

## RATE LIMIT RULE: USER ID

## PERMISSION: TRADE

## HTTP REQUEST

```
POST /api/v5/finance/staking-defi/cancel
```

## Request Example

```

import okx.Finance.StakingDefi as StakingDefi

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

StakingAPI = StakingDefi.StakingDefiAPI(apikey, secretkey, passphrase, False, flag)

result = StakingAPI.cancel(
    ordId = "1234",
    protocolType = "defi"
)
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
ordId	String	Yes	Order ID
protocolType	String	Yes	Protocol type defi: on-chain earn

## Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ordId": "754147",
      "tag": ""
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ordId	String	Order ID
tag	String	Order tag

## GET / ACTIVE ORDERS

**RATE LIMIT: 3 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

## HTTP REQUEST

```
GET /api/v5/finance/staking-defi/orders-active
```

## Request Example

```

import okx.Finance.StakingDefi as StakingDefi

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

StakingAPI = StakingDefi.StakingDefiAPI(apikey, secretkey, passphrase, False, flag)

```

```
result = StakingAPI.get_activity_orders()
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
productId	String	No	Product ID
protocolType	String	No	Protocol type defi: on-chain earn
ccy	String	No	Investment currency, e.g. BTC
state	String	No	Order state 8: Pending 13: Cancelling 9: Onchain 1: Earning 2: Redeeming

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "ordId": "2413499",
      "ccy": "DOT",
      "productId": "101",
      "state": "1",
      "protocol": "Polkadot",
      "protocolType": "defi",
      "term": "6",
      "apy": "0.1014",
      "investData": [
        {
          "ccy": "DOT",
          "amt": "2"
        }
      ],
      "earningData": [
        {
          "ccy": "DOT",
          "earningType": "0",
          "earnings": "0.10615025"
        }
      ],
      "purchasedTime": "1729839328000",
      "tag": "",
      "estSettlementTime": "",
      "cancelRedemptionDeadline": "",
      "fastRedemptionData": []
    },
    {
      "ordId": "2213257",
      "ccy": "USDT",
      "productId": "4005",
      "state": "1",
      "protocol": "On-Chain Defi",
      "protocolType": "defi",
      "term": "0",
      "apy": "0.0323",
      "investData": [
        {
          "ccy": "USDT",
          "amt": "1"
        }
      ],
      "earningData": [
        {
          "ccy": "USDT",
          "earningType": "0",
          "earnings": "0.0323"
        }
      ],
      "purchasedTime": "1729839328000",
      "tag": "",
      "estSettlementTime": "",
      "cancelRedemptionDeadline": "",
      "fastRedemptionData": []
    }
  ]
}
```

```

    "ccy": "USDT",
    "earningType": "0",
    "earnings": "0.02886582"
  },
  {
    "ccy": "COMP",
    "earningType": "1",
    "earnings": "0.0000627"
  }
],
"purchasedTime": "172534579000",
"tag": "",
"estSettlementTime": "",
"cancelRedemptionDeadline": "",
"fastRedemptionData": []
},
{
  "ordId": "2210943",
  "ccy": "USDT",
  "productId": "4005",
  "state": "1",
  "protocol": "On-Chain Defi",
  "protocolType": "defi",
  "term": "0",
  "apy": "0.0323",
  "investData": [
    {
      "ccy": "USDT",
      "amt": "1"
    }
  ],
  "earningData": [
    {
      "ccy": "USDT",
      "earningType": "0",
      "earnings": "0.02891823"
    },
    {
      "ccy": "COMP",
      "earningType": "1",
      "earnings": "0.0000632"
    }
  ],
  "purchasedTime": "1725280801000",
  "tag": "",
  "estSettlementTime": "",
  "cancelRedemptionDeadline": "",
  "fastRedemptionData": []
}
],
"msg": ""
}

```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency, e.g. <code>BTC</code>
ordId	String	Order ID
productId	String	Product ID
state	String	Order state <div style="display: flex; justify-content: space-between; align-items: center;"> <span>(8): Pending</span> <span>(13): Cancelling</span> </div> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>(9): Onchain</span> <span>(1): Earning</span> </div> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>(2): Redeeming</span> </div>
protocol	String	Protocol

Parameter	Type	Description
protocolType	String	Protocol type [defi]: on-chain earn
term	String	Protocol term It will return the days of fixed term and will return 0 for flexible product
apy	String	Estimated APY If the estimated APY is 7% , this field is 0.07 Retain to 4 decimal places (truncated)
investData	Array of objects	Investment data
> ccy	String	Investment currency, e.g. [BTC]
> amt	String	Invested amount
earningData	Array of objects	Earning data
> ccy	String	Earning currency, e.g. [BTC]
> earningType	String	Earning type 0: Estimated earning 1: Cumulative earning
> earnings	String	Earning amount
fastRedemptionData	Array of objects	Fast redemption data
> ccy	String	Currency, e.g. [BTC]
> redeemingAmt	String	Redeeming amount
purchasedTime	String	Order purchased time, Unix timestamp format in milliseconds, e.g. [1597026383085]
estSettlementTime	String	Estimated redemption settlement time
cancelRedemptionDeadline	String	Deadline for cancellation of redemption application
tag	String	Order tag

## GET / ORDER HISTORY

RATE LIMIT: 3 REQUESTS PER SECOND

RATE LIMIT RULE: USER ID

PERMISSION: READ

## HTTP REQUEST

GET /api/v5/finance/staking-defi/orders-history

### Request Example

```

import okx.Finance.StakingDefi as StakingDefi

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

StakingAPI = StakingDefi.StakingDefiAPI(apikey, secretkey, passphrase, False, flag)

```

```
result = StakingAPI.get_orders_history()
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
productId	String	No	Product ID
protocolType	String	No	Protocol type defi: on-chain earn
ccy	String	No	Investment currency, e.g. <a href="#">BTC</a>
after	String	No	Pagination of data to return records earlier than the requested ID. The value passed is the corresponding <a href="#">ordId</a>
before	String	No	Pagination of data to return records newer than the requested ID. The value passed is the corresponding <a href="#">ordId</a>
limit	String	No	Number of results per request. The default is <a href="#">100</a> . The maximum is <a href="#">100</a> .

## Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ordId": "1579252",
      "ccy": "DOT",
      "productId": "101",
      "state": "3",
      "protocol": "Polkadot",
      "protocolType": "defi",
      "term": "0",
      "apy": "0.1704",
      "investData": [
        {
          "ccy": "DOT",
          "amt": "2"
        }
      ],
      "earningData": [
        {
          "ccy": "DOT",
          "earningType": "0",
          "realizedEarnings": "0"
        }
      ],
      "purchasedTime": "1712908001000",
      "redeemedTime": "1712914294000",
      "tag": ""
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency, e.g. <a href="#">BTC</a>
ordId	String	Order ID
productId	String	Product ID

Parameter	Type	Description
state	String	Order state ③: Completed (including canceled and redeemed)
protocol	String	Protocol
protocolType	String	Protocol type ④: on-chain earn
term	String	Protocol term It will return the days of fixed term and will return ④ for flexible product
apy	String	Estimated APY If the estimated APY is 7% , this field is ④ Retain to 4 decimal places (truncated)
investData	Array of objects	Investment data
> ccy	String	Investment currency, e.g. ④
> amt	String	Invested amount
earningData	Array of objects	Earning data
> ccy	String	Earning currency, e.g. ④
> earningType	String	Earning type ④: Estimated earning ⑤: Cumulative earning
> realizedEarnings	String	Cumulative earning of redeemed orders This field is just valid when the order is in redemption state
purchasedTime	String	Order purchased time, Unix timestamp format in milliseconds, e.g. ④
redeemedTime	String	Order redeemed time, Unix timestamp format in milliseconds, e.g. ④
tag	String	Order tag

## ETH staking

ETH Staking, also known as Ethereum Staking, is the process of participating in the Ethereum blockchain's Proof-of-Stake (PoS) consensus mechanism. Stake to receive BETH for liquidity at 1:1 ratio and earn daily BETH rewards

Learn more about ETH Staking

### GET / PRODUCT INFO

**RATE LIMIT: 3 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

### HTTP REQUEST

GET /api/v5/finance/staking-defi/eth/product-info

Request Example

```
import okx.Finance.EthStaking as EthStaking
```

```
# API initialization
```

```
apikey = "YOUR_API_KEY"
```

```

secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

StackingAPI = EthStaking.EthStakingAPI(apikey, secretkey, passphrase, False, flag)

result = StackingAPI.eth_product_info()
print(result)

```

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "fastRedemptionDailyLimit": "100"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
fastRedemptionDailyLimit	String	Fast redemption daily limit The master account and sub-accounts share the same limit

## POST / PURCHASE

Staking ETH for BETH

Only the assets in the funding account can be used.

**RATE LIMIT: 2 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

```
POST /api/v5/finance/staking-defi/eth/purchase
```

## Request Example

```

import okx.Finance.EthStaking as EthStaking

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

StackingAPI = EthStaking.EthStakingAPI(apikey, secretkey, passphrase, False, flag)

result = StackingAPI.eth_purchase(amt="1")
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
amt	String	Yes	Investment amount

## Response Example

```
{  
  "code": "0",  
  "msg": "",  
  "data": [  
  ]  
}
```

## RESPONSE PARAMETERS

code = 0 means your request has been successfully handled.

## POST / REDEEM

Only the assets in the funding account can be used. If your BETH is in your trading account, you can make funding transfer first.

### RATE LIMIT: 2 REQUESTS PER SECOND

### RATE LIMIT RULE: USER ID

### PERMISSION: TRADE

### HTTP REQUEST

```
POST /api/v5/finance/staking-defi/eth/redeem
```

#### Request Example

```
import okx.Finance.EthStaking as EthStaking  
  
# API initialization  
apikey = "YOUR_API_KEY"  
secretkey = "YOUR_SECRET_KEY"  
passphrase = "YOUR_PASSPHRASE"  
  
flag = "0" # Production trading:0 , demo trading:1  
  
StackingAPI = EthStaking.EthStakingAPI(apikey, secretkey, passphrase, False, flag)  
  
result = StackingAPI.eth_redeem(amt="1")  
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
amt	String	Yes	Redeeming amount

#### Response Example

```
{  
  "code": "0",  
  "msg": "",  
  "data": [  
  ]  
}
```

## RESPONSE PARAMETERS

code = 0 means your request has been successfully handled.

## POST / CANCEL REDEEM

### RATE LIMIT: 2 REQUESTS PER SECOND

### RATE LIMIT RULE: USER ID

### PERMISSION: TRADE

### HTTP REQUEST

```
POST /api/v5/finance/staking-defi/eth/cancel-redeem
```

## Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
ordId	String	Yes	Order ID

## Response Example

```
{  
  "code": "0",  
  "data": [  
    {  
      "ordId": "1234567890"  
    }  
  ],  
  "msg": ""  
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
ordId	String	Order ID

## GET / BALANCE

The balance is a snapshot summarized all BETH assets (including assets in redeeming) in account.

### RATE LIMIT: 6 REQUESTS PER SECOND

### RATE LIMIT RULE: USER ID

### PERMISSION: READ

### HTTP REQUEST

```
GET /api/v5/finance/staking-defi/eth/balance
```

## Request Example

```
import okx.Finance.EthStaking as EthStaking  
  
# API initialization  
apikey = "YOUR_API_KEY"  
secretkey = "YOUR_SECRET_KEY"  
passphrase = "YOUR_PASSPHRASE"  
  
flag = "0" # Production trading:0 , demo trading:1  
  
StackingAPI = EthStaking.EthStakingAPI(apikey, secretkey, passphrase, False, flag)  
  
result = StackingAPI.eth_balance()  
print(result)
```

### REQUEST PARAMETERS

None

## Response Example

```
{  
  "code": "0",  
  "data": [  
    {  
      "amt": "0.63926191",  
      "ccy": "BETH",  
      "latestInterestAccrual": "0.00006549",  
      "totalInterestAccrual": "0.01490596",  
      "ts": "1699257600000"  
    }  
  ]  
}
```

```

},
"msg": ""
}
]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency, e.g. <code>BETH</code>
amt	String	Currency amount
latestInterestAccrual	String	Latest interest accrual
totalInterestAccrual	String	Total interest accrual
ts	String	Query data time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## GET / PURCHASE&REDEEM HISTORY

**RATE LIMIT: 6 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

### HTTP REQUEST

```
GET /api/v5/finance/staking-defi/eth/purchase-redeem-history
```

#### Request Example

```

import okx.Finance.EthStaking as EthStaking

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

StackingAPI = EthStaking.EthStakingAPI(apikey, secretkey, passphrase, False, flag)

result = StackingAPI.eth_purchase_redeem_history()
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
type	String	No	Type <code>purchase</code> <code>redeem</code>
status	String	No	Status <code>pending</code> <code>success</code> <code>failed</code> <code>cancelled</code>
after	String	No	Pagination of data to return records earlier than the <code>requestTime</code> . The value passed is the corresponding <code>times</code> <code>stamp</code>
before	String	No	Pagination of data to return records newer than the <code>requestTime</code> . The value passed is the corresponding <code>times</code> <code>stamp</code>
limit	String	No	Number of results per request. The default is <code>100</code> . The maximum is <code>100</code> .

## Response Example

```
{  
  "code": "0",  
  "data": [  
    {  
      "amt": "0.62666630",  
      "completedTime": "1683413171000",  
      "estCompletedTime": "",  
      "redeemingAmt": "",  
      "requestTime": "1683413171000",  
      "status": "success",  
      "type": "purchase"  
    }  
  ],  
  "msg": ""  
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
type	String	Type <code>purchase</code> <code>redeem</code>
amt	String	Purchase/Redeem amount
redeemingAmt	String	Redeeming amount
status	String	Status <code>pending</code> <code>success</code> <code>failed</code> <code>cancelled</code>
ordId	String	Order ID
requestTime	String	Request time of make purchase/redeem, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
completedTime	String	Completed time of redeem settlement, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
estCompletedTime	String	Estimated completed time of redeem settlement, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## GET / APY HISTORY (PUBLIC)

Public endpoints don't need authorization.

### RATE LIMIT: 6 REQUESTS PER SECOND

### RATE LIMIT RULE: IP

### HTTP REQUEST

```
GET /api/v5/finance/staking-defi/eth/apy-history
```

## Request Example

```
import okx.Finance.EthStaking as EthStaking  
  
flag = "0" # Production trading:0 , demo trading:1  
  
StackingAPI = EthStaking.EthStakingAPI(flag=flag)  
  
result = StackingAPI.eth_apy_history(days="7")  
print(result)
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
days	String	Yes	Get the days of APY(Annual percentage yield) history record in the past No more than 365 days

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "rate": "0.02690000",
      "ts": "1734195600000"
    },
    {
      "rate": "0.02840000",
      "ts": "1734109200000"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
rate	String	APY(Annual percentage yield), e.g. <code>0.01</code> represents <code>1%</code>
ts	String	Data time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## SOL staking

By staking SOL tokens and delegating them to validators on the Solana network, you can receive equivalent OKSOL and earn extra OKSOL rewards.

Stake SOL on Solana to receive OKSOL at a 1:1 ratio for liquidity

Learn more about OKSOL Staking

#### GET / PRODUCT INFO

**RATE LIMIT: 3 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

#### HTTP REQUEST

```
GET /api/v5/finance/staking-defi/sol/product-info
```

#### Request Example

#### Response Example

```
{
  "code": "0",
  "data": {
    "fastRedemptionAvail": "240",
    "fastRedemptionDailyLimit": "240"
  },
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
fastRedemptionDailyLimit	String	Fast redemption daily limit The master account and sub-accounts share the same limit
fastRedemptionAvail	String	Currently fast redemption max available amount

#### POST / PURCHASE

Staking SOL for OKSOL

Only the assets in the funding account can be used.

**RATE LIMIT: 2 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

`POST /api/v5/finance/staking-defi/sol/purchase`

Request Example

```
import okx.Finance.SolStaking as SolStaking

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

StackingAPI = SolStaking.SolStakingAPI(apikey, secretkey, passphrase, False, flag)

result = StackingAPI.sol_purchase(amt="1")
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
amt	String	Yes	Investment amount

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
  ]
}
```

#### RESPONSE PARAMETERS

code = 0 means your request has been successfully handled.

#### POST / REDEEM

Only the assets in the funding account can be used. If your OKSOL is in your trading account, you can make funding transfer first.

**RATE LIMIT: 2 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

`POST /api/v5/finance/staking-defi/sol/redeem`

## Request Example

```
import okx.Finance.SolStaking as SolStaking

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

StackingAPI = SolStaking.SolStakingAPI(apikey, secretkey, passphrase, False, flag)

result = StackingAPI.sol_redeem(amt="1")
print(result)
```

### REQUEST PARAMETERS

Parameter	Type	Required	Description
amt	String	Yes	Redeeming amount

## Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
  ]
}
```

### RESPONSE PARAMETERS

code = 0 means your request has been successfully handled.

## POST / CANCEL REDEEM

**RATE LIMIT: 2 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

```
POST /api/v5/finance/staking-defi/sol/cancel-redeem
```

## Request Example

### REQUEST PARAMETERS

Parameter	Type	Required	Description
ordId	String	Yes	Order ID

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "ordId": "1234567890"
    }
  ],
  "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
ordId	String	Order ID

#### GET / BALANCE

The balance is summarized all OKSOL assets (including assets in redeeming) in account.

**RATE LIMIT: 6 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

#### HTTP REQUEST

GET /api/v5/finance/staking-defi/sol/balance

#### Request Example

```
import okx.Finance.SolStaking as SolStaking

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

StackingAPI = SolStaking.SolStakingAPI(apikey, secretkey, passphrase, False, flag)

result = StackingAPI.sol_balance()
print(result)
```

#### REQUEST PARAMETERS

None

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "amt": "0.01100012",
      "ccy": "OKSOL",
      "latestInterestAccrual": "0.00000012",
      "totalInterestAccrual": "0.00000012"
    }
  ],
  "msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency, e.g. <code>OKSOL</code>
amt	String	Currency amount
latestInterestAccrual	String	Latest interest accrual
totalInterestAccrual	String	Total interest accrual

#### GET / PURCHASE&REDEEM HISTORY

**RATE LIMIT: 6 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

## HTTP REQUEST

GET /api/v5/finance/staking-defi/sol/purchase-redeem-history

## Request Example

```

import okx.Finance.SolStaking as SolStaking

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

StackingAPI = SolStaking.SolStakingAPI(apikey, secretkey, passphrase, False, flag)

result = StackingAPI.sol_purchase_redeem_history()
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
type	String	No	Type <input type="button" value="purchase"/> <input type="button" value="redeem"/>
status	String	No	Status <input type="button" value="pending"/> <input type="button" value="success"/> <input type="button" value="failed"/> <input type="button" value="cancelled"/>
after	String	No	Pagination of data to return records earlier than the <input type="button" value="requestTime"/> . The value passed is the corresponding <input type="button" value="times"/> <input type="button" value="stamp"/>
before	String	No	Pagination of data to return records newer than the <input type="button" value="requestTime"/> . The value passed is the corresponding <input type="button" value="times"/> <input type="button" value="stamp"/>
limit	String	No	Number of results per request. The default is <input type="button" value="100"/> . The maximum is <input type="button" value="100"/> .

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "amt": "0.62666630",
      "completedTime": "1683413171000",
      "estCompletedTime": "",
      "redeemingAmt": "",
      "requestTime": "1683413171000",
      "status": "success",
      "type": "purchase"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
type	String	Type <input type="button" value="purchase"/>

Parameter	Type	Description
		redeem
amt	String	Purchase/Redeem amount
redeemingAmt	String	Redeeming amount
		Status
		pending
		success
		failed
		cancelled
ordId	String	Order ID
requestTime	String	Request time of make purchase/redeem, Unix timestamp format in milliseconds, e.g. 1597026383085
completedTime	String	Completed time of redeem settlement, Unix timestamp format in milliseconds, e.g. 1597026383085
estCompletedTime	String	Estimated completed time of redeem settlement, Unix timestamp format in milliseconds, e.g. 1597026383085

#### GET / APY HISTORY (PUBLIC)

Public endpoints don't need authorization.

#### RATE LIMIT: 6 REQUESTS PER SECOND

#### RATE LIMIT RULE: IP

#### HTTP REQUEST

GET /api/v5/finance/staking-defi/sol/apy-history

#### Request Example

```
import okx.Finance.SolStaking as SolStaking

flag = "0" # Production trading:0 , demo trading:1

StackingAPI = SolStaking.SolStakingAPI(flag=flag)

result = StackingAPI.sol_apy_history(days="7")
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
days	String	Yes	Get the days of APY(Annual percentage yield) history record in the past No more than 365 days

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "rate": "0.11280000",
      "ts": "1734192000000"
    },
    {
      "rate": "0.11270000",
      "ts": "1734105600000"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
rate	String	APY(Annual percentage yield), e.g. <code>0.01</code> represents <code>1%</code>
ts	String	Data time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## Simple earn flexible

Simple earn flexible (saving) is earned by lending to leveraged trading users in the lending market. [learn more](#)

### GET / SAVING BALANCE

**RATE LIMIT: 6 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/finance/savings/balance`

Request Example

```
import okx.Finance.Savings as Savings

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

SavingsAPI = Savings.SavingsAPI(apikey, secretkey, passphrase, False, flag)

result = SavingsAPI.get_saving_balance(ccy="USDT")
print(result)
```

### REQUEST PARAMETERS

Parameters	Types	Required	Description
ccy	String	No	Currency, e.g. <code>BTC</code>

Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "earnings": "0.0010737388791526",
      "redemptAmt": "",
      "rate": "0.0100000000000000",
      "ccy": "USDT",
      "amt": "11.0010737453457821",
      "loanAmt": "11.0010630707982819",
      "pendingAmt": "0.0000106745475002"
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency
amt	String	Currency amount
earnings	String	Currency earnings
rate	String	Minimum annual lending rate configured by users
loanAmt	String	Lending amount
pendingAmt	String	Pending amount
redemptAmt	String	Redempting amount (Deprecated)

#### POST / SAVINGS PURCHASE/REDEMPTION

Only the assets in the funding account can be used for saving.

**RATE LIMIT: 6 REQUESTS PER SECOND**

**RATE LIMIT RULE: USER ID**

**PERMISSION: TRADE**

**HTTP REQUEST**

POST /api/v5/finance/savings/purchase-redempt

#### Request Example

```
import okx.Finance.Savings as Savings

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

SavingsAPI = Savings.SavingsAPI(apikey, secretkey, passphrase, False, flag)

result = SavingsAPI.savings_purchase_redemption(ccy='USDT',amt="0.1",side="purchase",rate="1")
print(result)
```

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	Yes	Currency, e.g. <code>BTC</code>
amt	String	Yes	Purchase/redemption amount
side	String	Yes	Action type. <code>purchase</code> : purchase saving shares <code>redempt</code> : redeem saving shares
rate	String	Conditional	Annual purchase rate, e.g. <code>0.1</code> represents <code>10%</code> Only applicable to purchase saving shares The interest rate of the new subscription will cover the interest rate of the last subscription The rate value range is between 1% and 365%

#### Response Example

```
{
  "code": "0",
  "msg": ""}
```

```

"data": [
  {
    "ccy": "BTC",
    "amt": "1",
    "side": "purchase",
    "rate": "0.01"
  }
]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency
amt	String	Purchase/Redemption amount
side	String	Action type
rate	String	Annual purchase rate, e.g. <code>0.1</code> represents <code>10%</code>

## POST / SET LENDING RATE

RATE LIMIT: 6 REQUESTS PER SECOND

RATE LIMIT RULE: USER ID

PERMISSION: TRADE

HTTP REQUEST

`POST /api/v5/finance/savings/set-lending-rate`

### Request Example

```

import okx.Finance.Savings as Savings

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

SavingsAPI = Savings.SavingsAPI(apikey, secretkey, passphrase, False, flag)

result = SavingsAPI.set_lending_rate(ccy='USDT',rate="1")
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
ccy	String	Yes	Currency, e.g. <code>BTC</code>
rate	String	Yes	Annual lending rate The rate value range is between 1% and 365%

### Response Example

```

{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ccy": "BTC",
      "rate": "0.02"
    }
  ]
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency, e.g. <code>BTC</code>
rate	String	Annual lending rate

## GET / LENDING HISTORY

Return data in the past month.

### RATE LIMIT: 6 REQUESTS PER SECOND

### RATE LIMIT RULE: USER ID

### PERMISSION: READ

### HTTP REQUEST

```
GET /api/v5/finance/savings/lending-history
```

#### Request Example

```
import okx.Finance.Savings as Savings

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

SavingsAPI = Savings.SavingsAPI(apikey, secretkey, passphrase, False, flag)

result = SavingsAPI.get_lending_history()
print(result)
```

## REQUEST PARAMETERS

Parameters	Types	Required	Description
ccy	String	No	Currency, e.g. <code>BTC</code>
after	String	No	Pagination of data to return records earlier than the requested <code>ts</code> , Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
before	String	No	Pagination of data to return records newer than the requested <code>ts</code> , Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. The maximum is <code>100</code> . The default is <code>100</code> .

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ccy": "BTC",
      "amt": "0.01",
      "earnings": "0.001",
      "rate": "0.01",
      "ts": "1597026383085"
    },
    {
      "ccy": "ETH",
      "amt": "0.2",
      "earnings": "0.001",
      "rate": "0.01",
      "ts": "1597026383085"
    }
  ]
}
```

]  
}

## RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency, e.g. <code>BTC</code>
amt	String	Lending amount
earnings	String	Currency earnings
rate	String	Lending annual interest rate
ts	String	Lending time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## GET / PUBLIC BORROW INFO (PUBLIC)

Authentication is not required for this public endpoint.

**RATE LIMIT: 6 REQUESTS PER SECOND**

**RATE LIMIT RULE: IP**

### HTTP REQUEST

`GET /api/v5/finance/savings/lending-rate-summary`

#### Request Example

```
import okx.Finance.Savings as Savings

flag = "0" # Production trading:0 , demo trading:1

SavingsAPI = Savings.SavingsAPI(flag=flag)

result = SavingsAPI.get_public_borrow_info()
print(result)
```

## REQUEST PARAMETERS

Parameters	Types	Required	Description
ccy	String	No	Currency, e.g. <code>BTC</code>

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ccy": "BTC",
      "avgAmt": "10000",
      "avgAmtUsd": "1000000000",
      "avgRate": "0.03",
      "preRate": "0.02",
      "estRate": "0.01"
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency, e.g. <code>BTC</code>
avgAmt	String	24H average borrowing amount(deprecated)

Parameter	Type	Description
avgAmtUsd	String	24-hours average borrowing amount in <code>USD</code> value (deprecated)
avgRate	String	24-hours average annual borrowing rate
preRate	String	Last annual borrowing interest rate
estRate	String	Next estimate annual borrowing interest rate

#### GET / PUBLIC BORROW HISTORY (PUBLIC)

Authentication is not required for this public endpoint.

Only returned records after December 14, 2021.

**RATE LIMIT: 6 REQUESTS PER SECOND**

**RATE LIMIT RULE: IP**

#### HTTP REQUEST

`GET /api/v5/finance/savings/lending-rate-history`

#### Request Example

```
import okx.Finance.Savings as Savings

flag = "0" # Production trading:0 , demo trading:1

SavingsAPI = Savings.SavingsAPI(flag=flag)

result = SavingsAPI.get_public_borrow_history()
print(result)
```

#### REQUEST PARAMETERS

Parameters	Types	Required	Description
ccy	String	No	Currency, e.g. <code>BTC</code>
after	String	No	Pagination of data to return records earlier than the requested <code>ts</code> , Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
before	String	No	Pagination of data to return records newer than the requested <code>ts</code> , Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
limit	String	No	Number of results per request. The maximum is <code>100</code> . The default is <code>100</code> . If <code>ccy</code> is not specified, all data under the same <code>ts</code> will be returned, not limited by <code>limit</code>

#### Response Example

```
{
  "code": "0",
  "msg": "",
  "data": [
    {
      "ccy": "BTC",
      "amt": "0.01",
      "rate": "0.001",
      "ts": "1597026383085"
    }
  ]
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
ccy	String	Currency, e.g. <code>BTC</code>

Parameter	Type	Description
amt	String	Lending amount(deprecated)
rate	String	Annual borrowing interest rate
ts	String	Time, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## Flexible loan

OKX Flexible Loan is a high-end loan product that allows users to increase cash flow without selling off their crypto. More details

### GET / BORROWABLE CURRENCIES

Get borrowable currencies

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

### HTTP REQUEST

`GET /api/v5/finance/flexible-loan/borrow-currencies`

#### Request Example

```
from okx.Finance import FlexibleLoan

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

flexibleLoanAPI = FlexibleLoan.FlexibleLoanAPI(apikey, secretkey, passphrase, False, flag)
result = flexibleLoanAPI.borrow_currencies()
print(result)
```

#### Response Example

```
{
  "code": "0",
  "data": [
    {
      "borrowCcy": "USDT"
    },
    {
      "borrowCcy": "USDC"
    }
  ],
  "msg": ""
}
```

### RESPONSE PARAMETERS

Parameter	Type	Description
borrowCcy	String	Borrowable currency, e.g. <code>BTC</code>

### GET / COLLATERAL ASSETS

Get collateral assets in funding account.

## RATE LIMIT RULE: USER ID

## PERMISSION: READ

## HTTP REQUEST

GET /api/v5/finance/flexible-loan/collateral-assets

## Request Example

```
from okx.Finance import FlexibleLoan

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

flexibleLoanAPI = FlexibleLoan.FlexibleLoanAPI(apikey, secretkey, passphrase, False, flag)
result = flexibleLoanAPI.collateral_assets()
print(result)
```

## REQUEST PARAMETERS

Parameters	Types	Required	Description
ccy	String	No	Collateral currency, e.g. <a href="#">BTC</a>

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "assets": [
        {
          "amt": "1.7921483143067599",
          "ccy": "BTC",
          "notionalUsd": "158292.621793314105231"
        },
        {
          "amt": "1.9400755578876945",
          "ccy": "ETH",
          "notionalUsd": "6325.6652712507628946"
        },
        {
          "amt": "63.9795959720319628",
          "ccy": "USDT",
          "notionalUsd": "64.3650372635940345"
        }
      ]
    },
    {
      "msg": ""
    }
  ]
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
assets	Array of objects	Collateral assets data
> ccy	String	Currency, e.g. <a href="#">BTC</a>
> amt	String	Available amount
> notionalUsd	String	Notional value in <a href="#">USD</a>

## POST / MAXIMUM LOAN AMOUNT

RATE LIMIT: 5 REQUESTS PER 2 SECONDS

RATE LIMIT RULE: USER ID

PERMISSION: READ

HTTP REQUEST

POST /api/v5/finance/flexible-loan/max-loan

Request Example

```
from okx.Finance import FlexibleLoan

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

flexibleLoanAPI = FlexibleLoan.FlexibleLoanAPI(apikey, secretkey, passphrase, False, flag)
result = flexibleLoanAPI.max_loan(borrowCcy="USDT")
print(result)
```

## REQUEST PARAMETERS

Parameters	Types	Required	Description
borrowCcy	String	Yes	Currency to borrow, e.g. <code>USDT</code>
supCollateral	Array of objects	No	Supplementary collateral assets
> ccy	String	Yes	Currency, e.g. <code>BTC</code>
> amt	String	Yes	Amount

Response Example

```
{
  "code": "0",
  "data": [
    {
      "borrowCcy": "USDT",
      "maxLoan": "0.01113",
      "notionalUsd": "0.01113356",
      "remainingQuota": "3395000"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
borrowCcy	String	Currency to borrow, e.g. <code>USDT</code>
maxLoan	String	Maximum available loan
notionalUsd	String	Maximum available loan notional value, unit in <code>USD</code>
remainingQuota	String	Remaining quota, unit in <code>borrowCcy</code>

## GET / MAXIMUM COLLATERAL REDEEM AMOUNT

RATE LIMIT: 5 REQUESTS PER 2 SECONDS

**RATE LIMIT RULE: USER ID****PERMISSION: READ****HTTP REQUEST****GET /api/v5/finance/flexible-loan/max-collateral-redeem-amount****Request Example**

```

from okx.Finance import FlexibleLoan

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

flexibleLoanAPI = FlexibleLoan.FlexibleLoanAPI(apikey, secretkey, passphrase, False, flag)
result = flexibleLoanAPI.max_collateral_redeem_amount("USDT")
print(result)

```

**REQUEST PARAMETERS**

Parameters	Types	Required	Description
ccy	String	Yes	Collateral currency, e.g. <code>USDT</code>

**Response Example**

```

{
  "code": "0",
  "data": [
    {
      "ccy": "USDT",
      "maxRedeemAmt": "1"
    }
  ],
  "msg": ""
}

```

**RESPONSE PARAMETERS**

Parameter	Type	Description
ccy	String	Collateral currency, e.g. <code>USDT</code>
maxRedeemAmt	String	Maximum collateral redeem amount

**POST / ADJUST COLLATERAL****RATE LIMIT: 5 REQUESTS PER 2 SECONDS****RATE LIMIT RULE: USER ID****PERMISSION: TRADE****HTTP REQUEST****POST /api/v5/finance/flexible-loan/adjust-collateral****Request Example**

```

from okx.Finance import FlexibleLoan

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

```

```
flag = "0" # Production trading:0 , demo trading:1

flexibleLoanAPI = FlexibleLoan.FlexibleLoanAPI(apikey, secretkey, passphrase, False, flag)
result = flexibleLoanAPI.adjust_collateral(type="add", collateralCcy="USDT", collateralAmt="1")
print(result)
```

## REQUEST PARAMETERS

Parameters	Types	Required	Description
type	String	Yes	Operation type [ <b>add</b> ]: Add collateral [ <b>reduce</b> ]: Reduce collateral
collateralCcy	String	Yes	Collateral currency, e.g. <b>BTC</b>
collateralAmt	String	Yes	Collateral amount

## Response Example

```
{
  "code": "0",
  "data": [
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

code = **0** means your request has been accepted (It doesn't mean the request has been successfully handled.)

## GET / LOAN INFO

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

## HTTP REQUEST

```
GET /api/v5/finance/flexible-loan/loan-info
```

## Request Example

```
from okx.Finance import FlexibleLoan

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

flexibleLoanAPI = FlexibleLoan.FlexibleLoanAPI(apikey, secretkey, passphrase, False, flag)
result = flexibleLoanAPI.loan_info()
print(result)
```

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "collateralData": [
        {
          "amt": "0.0000097",
          "ccy": "COMP"
        }
      ]
    }
  ]
}
```

```

        "amt": "0.78",
        "ccy": "STX"
    },
    {
        "amt": "0.001",
        "ccy": "DOT"
    },
    {
        "amt": "0.05357864",
        "ccy": "LUNA"
    }
],
"collateralNotionalUsd": "1.5078763",
"curLTV": "0.5742",
"liqLTV": "0.8374",
"loanData": [
    {
        "amt": "0.86590608",
        "ccy": "USDC"
    }
],
"loanNotionalUsd": "0.8661285",
"marginCallLTV": "0.7374",
"riskWarningData": {
    "instId": "",
    "liqPx": ""
}
},
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
loanNotionalUsd	String	Loan value in <a href="#">USD</a>
loanData	Array of objects	Loan data
> ccy	String	Loan currency, e.g. <a href="#">USDT</a>
> amt	String	Loan amount
collateralNotionalUsd	String	Collateral value in <a href="#">USD</a>
collateralData	Array of objects	Collateral data
> ccy	String	Collateral currency, e.g. <a href="#">BTC</a>
> amt	String	Collateral amount
riskWarningData	Object	Risk warning data
> instId	String	<p>Liquidation instrument ID, e.g. <a href="#">BTC-USDT</a></p> <p>This field is only valid when there is only one type of collateral and one type of borrowed currency. In other cases, it returns "".</p>
> liqPx	String	<p>Liquidation price</p> <p>The unit of the liquidation price is the quote currency of the instrument, e.g. <a href="#">USDT</a> in <a href="#">BTC-USDT</a></p> <p>This field is only valid when there is only one type of collateral and one type of borrowed currency. In other cases, it returns "".</p>
curLTV	String	<p>Current LTV, e.g. <a href="#">0.1</a> represents <a href="#">10%</a></p> <p>Note: LTV = Loan to Value</p>

Parameter	Type	Description
marginCallLTV	String	Margin call LTV, e.g. <code>0.1</code> represents <code>10%</code> If your loan hits the margin call LTV, our system will automatically warn you that your loan is getting close to forced liquidation.
liqLTV	String	Liquidation LTV, e.g. <code>0.1</code> represents <code>10%</code> If your loan reaches liquidation LTV, it'll trigger forced liquidation. When this happens, you'll lose access to your collateral and any repayments made.

## GET / LOAN HISTORY

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

## RATE LIMIT RULE: USER ID

## PERMISSION: READ

## HTTP REQUEST

GET /api/v5/finance/flexible-loan/loan-history

## Request Example

```
from okx.Finance import FlexibleLoan

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

flexibleLoanAPI = FlexibleLoan.FlexibleLoanAPI(apikey, secretkey, passphrase, False, flag)
result = flexibleLoanAPI.loan_history()
print(result)
```

## REQUEST PARAMETERS

Parameters	Types	Required	Description
			Action type
			<code>borrowed</code>
			<code>repaid</code>
			<code>collateral_locked</code>
			<code>collateral_released</code>
			<code>forced_repayment_buy</code>
			<code>forced_repayment_sell</code>
			<code>forced_liquidation</code>
			<code>partial_liquidation</code>
			<code>sell_collateral</code>
			<code>buy_transition_coin</code>
			<code>sell_transition_coin</code>
			<code>buy_borrowed_coin</code>
type	String	No	
after	String	No	Pagination of data to return records earlier than the requested <code>refId</code> (not include)
before	String	No	Pagination of data to return records newer than the requested <code>refId</code> (not include)
limit	String	No	Number of results per request. The maximum is <code>100</code> . The default is <code>100</code> .

### Response Example

```
{  
  "code": "0",  
  "data": [  
    {  
      "amt": "-0.001"  
    }  
  ]  
}
```

```

    "ccy": "DOT",
    "refId": "17316594851045086",
    "ts": "1731659485000",
    "type": "collateral_locked"
  },
],
"msg": ""
}

```

## RESPONSE PARAMETERS

Parameter	Type	Description
refId	String	Reference ID
type	String	Action type
ccy	String	Currency, e.g. <code>BTC</code>
amt	String	Amount
ts	String	Timestamp for the action, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

## GET / ACCRUED INTEREST

Retrieves the interest accrual history for flexible loans over the past 30 days.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**PERMISSION: READ**

### HTTP REQUEST

```
GET /api/v5/finance/flexible-loan/interest-accrued
```

#### Request Example

```

from okx.Finance import FlexibleLoan

# API initialization
apikey = "YOUR_API_KEY"
secretkey = "YOUR_SECRET_KEY"
passphrase = "YOUR_PASSPHRASE"

flag = "0" # Production trading:0 , demo trading:1

flexibleLoanAPI = FlexibleLoan.FlexibleLoanAPI(apikey, secretkey, passphrase, False, flag)
result = flexibleLoanAPI.interest_accrued()
print(result)

```

## REQUEST PARAMETERS

Parameters	Types	Required	Description
ccy	String	No	Loan currency, e.g. <code>BTC</code>
after	String	No	Pagination of data to return records earlier than the requested <code>refId</code> (not include)
before	String	No	Pagination of data to return records newer than the requested <code>refId</code> (not include)
limit	String	No	Number of results per request. The maximum is <code>100</code> . The default is <code>100</code> .

### 返回结果

```

{
  "code": "0",
  "data": [

```

```
{
  "ccy": "USDC",
  "interest": "0.00004054",
  "interestRate": "0.41",
  "loan": "0.86599309",
  "refId": "17319133035195744",
  "ts": "1731913200000"
}
],
"msg": ""
}
```

#### RESPONSE PARAMETERS

Parameter	Type	Description
refId	String	Reference ID
ccy	String	Loan currency, e.g. <code>BTC</code>
loan	String	Loan when calculated interest
interest	String	Interest
interestRate	String	APY, e.g. <code>0.01</code> represents <code>1%</code>
ts	String	Timestamp to calculated interest, Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>

# Affiliate

The Affiliate API offers affiliate users a flexible function to query the invitee information. Simply enter the UID of your direct invitee to access their relevant information, empowering your affiliate business growth and day-to-day business operation. If you have additional data requirements regarding the Affiliate API, please don't hesitate to contact your BD. We will reach out to you through your BD to provide more comprehensive API support.

## REST API

### GET THE INVITEE'S DETAIL

**RATE LIMIT: 20 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID**

**HTTP REQUEST**

`GET /api/v5/affiliate/invitee/detail`

Request sample

#### REQUEST PARAMETERS

Parameter	Type	Required	Description
uid	String	Yes	UID of the invitee. Only applicable to the UID of invitee master account. The data returned covers invitee master account and invitee sub-accounts.

Returned results

```
{
  "msg": "",
```

```

"code": "0",
"data": [
  {
    "accFee": "0",
    "affiliateCode": "HIIIIII",
    "depAmt": "0",
    "firstTradeTime": "",
    "inviteeLevel": "2",
    "inviteeRebateRate": "0.39",
    "joinTime": "1712546713000",
    "kycTime": "",
    "level": "Lv1",
    "region": "Vietnam",
    "totalCommission": "0",
    "volMonth": "0"
  }
]
}

```

## RESPONSE PARAMETERS

Parameter name	Type	Description
inviteeLevel	String	Invitee's relative level to the affiliate If the user is a invitee, the level will be 2.
joinTime	String	Timestamp that the rebate relationship is established, Unix timestamp in millisecond format, e.g. 1597026383085
inviteeRebateRate	String	Self rebate rate of the invitee (in decimal), e.g. 0.01 represents 10%
totalCommission	String	Total commission earned from the invitee, unit in usdt
firstTradeTime	String	Timestamp that the first trade is completed after the latest rebate relationship is established with the parent affiliate Unix timestamp in millisecond format, e.g. 1597026383085 If user has not traded, "" will be returned
level	String	Invitee trading fee level, e.g. Lv1
depAmt	String	Accumulated amount of deposit in USDT If user has not deposited, 0 will be returned
volMonth	String	Accumulated Trading volume in the current month in USDT If user has not traded, 0 will be returned
accFee	String	Accumulated Amount of trading fee in USDT If there is no any fee, 0 will be returned
kycTime	String	KYC2 verification time. Unix timestamp in millisecond format and the precision is in day If user has not passed KYC2, "" will be returned
region	String	User country or region. e.g. "United Kingdom"
affiliateCode	String	Affiliate invite code that the invitee registered/recalled via

## Status

### GET / Status

Get event status of system upgrade.

Planned system maintenance that may result in short interruption (lasting less than 5 seconds) or websocket disconnection (users can immediately reconnect) will not be announced. The maintenance will only be performed during times of low market volatility.

## HTTP REQUEST

GET /api/v5/system/status

## Request Example

```

import okx.Status as Status

flag = "0" # Production trading: 0, Demo trading: 1
statusAPI = Status.StatusAPI(
    domain="https://www.okx.com",
    flag=flag,
)

# Get event status of system upgrade
result = statusAPI.status()
print(result)

```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
state	String	No	<p>System maintenance status</p> <ul style="list-style-type: none"> <li><code>scheduled</code>: waiting</li> <li><code>ongoing</code>: processing</li> <li><code>pre_open</code>: pre_open</li> <li><code>completed</code>: completed</li> <li><code>canceled</code>: canceled</li> </ul> <p>Generally, <code>pre_open</code> last about 10 minutes. There will be <code>pre_open</code> when the time of upgrade is too long. If this parameter is not filled, the data with status <code>scheduled</code>, <code>ongoing</code> and <code>pre_open</code> will be returned by default</p>

## Response Example

```
{
  "code": "0",
  "data": [
    {
      "begin": "1672823400000",
      "end": "1672823520000",
      "href": "",
      "preOpenBegin": "",
      "scheDesc": "",
      "serviceType": "8",
      "state": "completed",
      "maintType": "1",
      "env": "1",
      "system": "unified",
      "title": "Trading account system upgrade (in batches of accounts)"
    }
  ],
  "msg": ""
}
```

## RESPONSE PARAMETERS

Parameter	Type	Description
title	String	The title of system maintenance instructions
state	String	System maintenance status
begin	String	Begin time of system maintenance, Unix timestamp format in milliseconds, e.g. <code>1617788463867</code>
end	String	Time of resuming trading totally. Unix timestamp format in milliseconds, e.g. <code>1617788463867</code> . It is expected end time before <code>completed</code> , changed to actual end time after <code>completed</code> .

Parameter	Type	Description
preOpenBegin	String	The time of pre_open. Canceling orders, placing Post Only orders, and transferring funds to trading accounts are back after <code>preOpenBegin</code> .
href	String	Hyperlink for system maintenance details, if there is no return value, the default value will be empty. e.g. ""
serviceType	String	Service type <input type="radio"/> 0: WebSocket <input type="radio"/> 5: Trading service <input type="radio"/> 6: Block trading <input type="radio"/> 7: Trading bot <input type="radio"/> 8: Trading service (in batches of accounts) <input type="radio"/> 9: Trading service (in batches of products) <input checked="" type="radio"/> 10: Spread trading <input type="radio"/> 11: Copy trading <input type="radio"/> 99: Others (e.g. Suspend partial instruments)
system	String	System <input type="radio"/> unified: Trading account
scheDesc	String	Rescheduled description, e.g. <code>Rescheduled from 2021-01-26T16:30:00.000Z to 2021-01-28T16:30:00.000Z</code>
maintType	String	Maintenance type <input type="radio"/> 1: Scheduled maintenance <input type="radio"/> 2: Unscheduled maintenance <input type="radio"/> 3: System disruption
env	String	Environment <input type="radio"/> 1: Production Trading <input type="radio"/> 2: Demo Trading

## WS / Status channel

Get the status of system maintenance and push when rescheduling and the system maintenance status and end time changes. First subscription: "Push the latest change data"; every time there is a state change, push the changed content.

Planned system maintenance that may result in short interruption (lasting less than 5 seconds) or websocket disconnection (users can immediately reconnect) will not be announced. The maintenance will only be performed during times of low market volatility.

### URL PATH

/ws/v5/public

#### Request Example

```

import asyncio
from okx.websocket.WsPublicAsync import WsPublicAsync

def callbackFunc(message):
    print(message)

async def main():
    ws = WsPublicAsync(url="wss://wspap.okx.com:8443/ws/v5/public")
    await ws.start()
    args = [
        {"channel": "status"}
    ]
    await ws.subscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

    await ws.unsubscribe(args, callback=callbackFunc)
    await asyncio.sleep(10)

```

```
asyncio.run(main())
```

## REQUEST PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message Provided by client. It will be returned in response message for identifying the corresponding request. A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters.
op	String	Yes	<code>subscribe</code> <code>unsubscribe</code>
args	Array of objects	Yes	List of subscribed channels
> channel	String	Yes	Channel name <code>status</code>

### Successful Response Example

```
{
  "id": "1512",
  "event": "subscribe",
  "arg": {
    "channel": "status"
  },
  "connId": "a4d3ae55"
}
```

### Failure Response Example

```
{
  "id": "1512",
  "event": "error",
  "code": "60012",
  "msg": "Invalid request: {\\"op\\": \\"subscribe\\\", \\"args\\\": [{ \\"channel\\\" : \\"status\\\" }]}",
  "connId": "a4d3ae55"
}
```

## RESPONSE PARAMETERS

Parameter	Type	Required	Description
id	String	No	Unique identifier of the message
event	String	Yes	<code>subscribe</code> <code>unsubscribe</code> <code>error</code>
arg	Object	No	Subscribed channel
> channel	String	Yes	Channel name <code>status</code>
code	String	No	Error code
msg	String	No	Error message
connId	String	Yes	WebSocket connection ID

### Push Data Example

```
{
  "arg": {
    "channel": "status"
  },
}
```

```

"data": [
  {
    "begin": "1672823400000",
    "end": "1672825980000",
    "href": "",
    "preOpenBegin": "",
    "scheDesc": "",
    "serviceType": "0",
    "state": "completed",
    "system": "unified",
    "maintType": "1",
    "env": "1",
    "title": "Trading account WebSocket system upgrade",
    "ts": "1672826038470"
  }
]
}

```

#### PUSH DATA PARAMETERS

Parameter	Type	Description
arg	Object	Successfully subscribed channel
> channel	String	Channel name
data	Array of objects	Subscribed data
> title	String	The title of system maintenance instructions
> state	String	System maintenance status, <code>scheduled</code> : waiting; <code>ongoing</code> : processing; <code>pre_open</code> : pre_open; <code>completed</code> : completed; <code>canceled</code> : canceled. Generally, <code>pre_open</code> last about 10 minutes. There will be <code>pre_open</code> when the time of upgrade is too long.
> begin	String	Start time of system maintenance, Unix timestamp format in milliseconds, e.g. <code>1617788463867</code>
> end	String	Time of resuming trading totally. Unix timestamp format in milliseconds, e.g. <code>1617788463867</code> . It is expected end time before <code>completed</code> , changed to actual end time after <code>completed</code> .
> preOpenBegin	String	The time of pre_open. Canceling orders, placing Post Only orders, and transferring funds to trading accounts are back after <code>preOpenBegin</code> .
> href	String	Hyperlink for system maintenance details, if there is no return value, the default value will be empty. e.g. ""
> serviceType	String	Service type, <code>0</code> : WebSocket ; <code>5</code> : Trading service; <code>6</code> : Block trading; <code>7</code> : Trading bot; <code>8</code> : Trading service (in batches of accounts); <code>9</code> : Trading service (in batches of products); <code>10</code> : Spread trading; <code>11</code> : Copy trading; <code>99</code> : Others (e.g. Suspend partial instruments)
> system	String	System, <code>unified</code> : Trading account
> scheDesc	String	Rescheduled description, e.g. <code>Rescheduled from 2021-01-26T16:30:00.000Z to 2021-01-28T16:30:00.000Z</code>
> maintType	String	Maintenance type <code>1</code> : Scheduled maintenance; <code>2</code> : Unscheduled maintenance; <code>3</code> : System disruption
> env	String	Environment. <code>1</code> : Production Trading, <code>2</code> : Demo Trading
> ts	String	Push time due to change event, Unix timestamp format in milliseconds, e.g. <code>1617788463867</code>

# Announcement

# GET / Announcements

Get announcements, the response is sorted by `pTime` and `businessPTime` with the most recent first. The sort will not be affected if the announcement is updated. Every page has 20 records

Authentication is optional for this endpoint.

It will be regarded as private endpoint and authentication is required if OK-ACCESS-KEY in HTTP header is delivered.

It will be regarded as public endpoint and authentication isn't required if OK-ACCESS-KEY in HTTP header isn't delivered.

There are differences between public endpoint and private endpoint.

For public endpoint, the response is restricted based on your request IP.

For private endpoint, the response is restricted based on your country of residence.

**RATE LIMIT: 5 REQUESTS PER 2 SECONDS**

**RATE LIMIT RULE: USER ID(PRIVATE) OR IP(PUBLIC)**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/support/announcements`

Request Example

**REQUEST PARAMETERS**

Parameter	Type	Required	Description
annType	String	No	Announcement type. Delivering the <code>annType</code> from "GET / Announcement types" Returning all when it is not posted
page	String	No	Page for pagination. The default is 1

Response Example

```
{
  "code": "0",
  "data": [
    {
      "details": [
        {
          "annType": "announcements-new-listings",
          "title": "OKX to list Virtuals Protocol (VIRTUAL) for spot trading",
          "url": "https://www.okx.com/help/okx-to-list-virtuals-protocol-virtual-for-spot-trading",
          "pTime": "1761620404821",
          "businessPTime": "1761620400000"
        },
        {
          "annType": "announcements-web3",
          "title": "Completion of X Layer Mainnet Upgrade",
          "url": "https://www.okx.com/help/completion-of-x-layer-mainnet-upgrade",
          "pTime": "1761582756071",
          "businessPTime": "1761580800000"
        },
        ...
      ],
      "totalPage": "123"
    }
  ],
  "msg": ""
}
```

**RESPONSE PARAMETERS**

Parameter	Type	Description
totalPage	String	Total number of pages
details	Array of objects	List of announcements
> title	String	Announcement title
> annType	String	Announcement type
> businessPTime	String	The time displayed on the announcement page for user reference. Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> pTime	String	The actual time the announcement was first published. Unix timestamp format in milliseconds, e.g. <code>1597026383085</code>
> url	String	Announcement url

## GET / Announcement types

Authentication is not required for this public endpoint.

The response is restricted based on your request IP.

**RATE LIMIT: 1 REQUEST PER 2 SECONDS**

**RATE LIMIT RULE: IP**

**PERMISSION: READ**

**HTTP REQUEST**

`GET /api/v5/support/announcement-types`

Request Example

**REQUEST PARAMETERS**

None

Response Example

```
{
  "code": "0",
  "data": [
    {
      "annType": "announcements-new-listings",
      "annTypeDesc": "New listings"
    },
    {
      "annType": "announcements-delistings",
      "annTypeDesc": "Delistings"
    }
  ],
  "msg": ""
}
```

**RESPONSE PARAMETERS**

Parameter	Type	Description
annType	String	Announcement type
annTypeDesc	String	Announcement type description

# Error Code

Here is the REST API Error Code

## REST API

REST API Error Code is from 50000 to 59999.

### PUBLIC

Error Code from 50000 to 53999

### GENERAL CLASS

Error Code	HTTP Status Code	Error Message
0	200	
1	200	Operation failed.
2	200	Bulk operation partially succeeded.
50000	400	Body for POST request cannot be empty.
50001	503	Service temporarily unavailable. Please try again later.
50002	400	JSON syntax error
50004	400	API endpoint request timeout. (does not mean that the request was successful or failed, please check the request result).
50005	410	API endpoint is inactive or unavailable.
50006	400	Invalid Content-Type. Please use "application/JSON".
50007	200	User blocked.
50008	200	User doesn't exist.
50009	200	Account is frozen due to stop-out.
50010	200	User ID cannot be empty.
50011	200	Rate limit reached. Please refer to API documentation and throttle requests accordingly.
50011	429	Too Many Requests
50012	200	Account status invalid. Check account status
50013	429	Systems are busy. Please try again later.
50014	400	Parameter {param0} can not be empty.
50015	400	Either parameter {param0} or {param1} is required
50016	400	Parameter {param0} does not match parameter {param1}
50017	200	Position frozen and related operations restricted due to auto-deleveraging (ADL). Please try again later.

Error Code	HTTP Status Code	Error Message
50018	200	Currency {param0} is frozen due to ADL. Operation restricted.
50019	200	Account frozen and related operations restricted due to auto-deleveraging (ADL). Please try again later.
50020	200	Position frozen and related operations restricted due to forced liquidation. Please try again later.
50021	200	Currency {param0} is frozen due to liquidation. Operation restricted.
50022	200	Account frozen and related operations restricted due to forced liquidation. Please try again later.
50023	200	Funding fees frozen and related operations are restricted. Please try again later.
50024	200	Parameter {param0} and {param1} can not exist at the same time.
50025	200	Parameter {param0} count exceeds the limit {param1}.
50026	500	System error. Try again later
50027	200	This account is restricted from trading. Please contact customer support for assistance.
50028	200	Unable to place the order. Please contact the customer service for details.
50029	200	Your account has triggered OKX risk control and is temporarily restricted from conducting transactions. Please check your email registered with OKX for contact from our customer support team.
50030	200	You don't have permission to use this API endpoint
50032	200	Your account has been set to prohibit transactions in this currency. Please confirm and try again
50033	200	Instrument blocked. Please verify trading this instrument is allowed under account settings and try again.
50035	403	This endpoint requires that APIKey must be bound to IP
50036	200	The expTime can't be earlier than the current system time. Please adjust the expTime and try again.
50037	200	Order expired.
50038	200	This feature is unavailable in demo trading
50039	200	Parameter "before" isn't supported for timestamp pagination
50040	200	Too frequent operations, please try again later
50041	200	Your user ID hasn't been allowlisted. Please contact customer service for assistance.
50042	200	Repeated request
50044	200	Must select one broker type
50045	200	simPos should be empty because simulated positions cannot be counted when Position Builder is calculating under Spot-Derivatives risk offset mode
50046	200	This feature is temporarily unavailable while we make some improvements to it. Please try again later.
50047	200	{param0} has already settled. To check the relevant candlestick data, please use {param1}
50048	200	Switching risk unit may lead position risk increases and be forced liquidated. Please adjust position size, make sure margin is in a safe status.
50049	200	No information on the position tier. The current instrument doesn't support margin trading.
50050	200	You've already activated options trading. Please don't activate it again.

Error Code	HTTP Status Code	Error Message
50051	200	Due to compliance restrictions in your country or region, you cannot use this feature.
50052	200	Due to local laws and regulations, you cannot trade with your chosen crypto.
50053	200	This feature is only available in demo trading.
50055	200	Reset unsuccessful. Assets can only be reset up to 5 times per day.
50056	200	You have pending orders or open positions with this currency. Please reset after canceling all the pending orders/closing all the open positions.
50057	200	Reset unsuccessful. Try again later.
50058	200	This crypto is not supported in an asset reset.
50059	200	Before you continue, you'll need to complete additional steps as required by your local regulators. Please visit the website or app for more details.
50060	200	For security and compliance purposes, please complete the identity verification process to continue using our services.
50061	200	You've reached the maximum order rate limit for this account.
50062	200	This feature is currently unavailable.
50063	200	You can't activate the credits as they might have expired or are already activated.
50064	200	The borrowing system is unavailable. Try again later.
50067	200	The API doesn't support cross site trading feature
50069	200	Margin ratio verification for this risk unit failed.
50071	200	{param} already exists e.g. clOrdId already exists
50072	200	Isolated margin mode is no longer supported in multi-currency margin mode and portfolio margin mode.
50075	200	The token is restricted in Islamic sub-account
50076	200	The time interval between the start date and end date cannot exceed {param0} days
50077	200	The time interval between the start date and end date cannot exceed {param0} months

#### API CLASS

Error Code	HTTP Status Code	Error Message
50100	400	API frozen, please contact customer service.
50101	401	APIKey does not match current environment.
50102	401	Timestamp request expired.
50103	401	Request header "OK-ACCESS-KEY" cannot be empty.
50104	401	Request header "OK-ACCESS-PASSPHRASE" cannot be empty.
50105	401	Request header "OK-ACCESS-PASSPHRASE" incorrect.
50106	401	Request header "OK-ACCESS-SIGN" cannot be empty.

Error Code	HTTP Status Code	Error Message
50107	401	Request header "OK-ACCESS-TIMESTAMP" cannot be empty.
50108	401	Exchange ID does not exist.
50109	401	Exchange domain does not exist.
50110	401	Your IP {param0} is not included in your API key's IP whitelist.
50111	401	Invalid OK-ACCESS-KEY.
50112	401	Invalid OK-ACCESS-TIMESTAMP.
50113	401	Invalid signature.
50114	401	Invalid authorization.
50115	405	Invalid request method.
50116	200	Fast API is allowed to create only one API key
50118	200	To link the app using your API key, your broker needs to share their IP to be whitelisted
50119	200	API key doesn't exist
50120	200	This API key doesn't have permission to use this function
50121	200	You can't access our services through the IP address ({param0})
50122	200	Order amount must exceed minimum amount

#### TRADE CLASS

Error Code	HTTP Status code	Error Message
51000	400	Parameter {param0} error
51001	200	Instrument ID, Instrument ID code or Spread ID doesn't exist.
51002	200	Instrument ID doesn't match underlying index.
51003	200	Either client order ID or order ID is required.
51004	200	Order failed. For isolated long/short mode of {param0}, the sum of current order size, position quantity in the same direction, and pending orders in the same direction can't be more than {param1}(contracts) which is the maximum position amount under current leverage. Please lower the leverage or use a new sub-account to place the order again (current leverage: {param2}×, current order size: {param3} contracts, position quantity in the same direction: {param4} contracts, pending orders in the same direction: {param5} contracts).
51004	200	Order failed. For cross long/short mode of {param0}, the sum of current order size, position quantity in the long and short directions, and pending orders in the long and short directions can't be more than {param1}(contracts) which is the maximum position amount under current leverage. Please lower the leverage or use a new sub-account to place the order again (current leverage: {param2}×, current order size: {param3} contracts, position quantity in the long and short directions: {param4} contracts, pending orders in the long and short directions: {param5} contracts).
51004	200	Order failed. For cross buy/sell mode of {param0} and instFamily {param1}, the sum of current order size, current instId position quantity in the long and short directions, current instId pending orders in the long and short directions, and other contracts of the same instFamily can't be more than {param2}(contracts) which is the maximum position amount under current leverage. Please lower the leverage or use a new sub-account to place the order again (current leverage: {param3}×, current order size: {param4} contracts, current instId position quantity in the long and short directions: {param5} contracts, current instId pending orders in the long and short directions: {param6} contracts, other contracts of the same instFamily: {param7} contracts).

Error Code	HTTP Status code	Error Message
51004	200	Order failed. For buy/sell mode of {param0}, the sum of current buy order size, position quantity, and pending buy orders can't be more than {param1}(contracts) which is the maximum position amount under current leverage. Please lower the leverage or use a new sub-account to place the order again (current leverage: {param2}×, current buy order size: {param3} contracts, position quantity: {param4} contracts, pending buy orders: {param5} contracts).
51004	200	Order failed. For buy/sell mode of {param0}, the sum of current sell order size, position quantity, and pending sell orders can't be more than {param1}(contracts) which is the maximum position amount under current leverage. Please lower the leverage or use a new sub-account to place the order again (current leverage: {param2}×, current sell order size: {param3} contracts, position quantity: {param4} contracts, pending sell orders: {param5} contracts).
51004	200	Order failed. For cross buy/sell mode of {param0} and instFamily {param1}, the sum of current buy order size, current instId position quantity, current instId pending buy orders, and other contracts of the same instFamily can't be more than {param2}(contracts) which is the maximum position amount under current leverage. Please lower the leverage or use a new sub-account to place the order again (current leverage: {param3}×, current buy order size: {param4} contracts, current instId position quantity: {param5} contracts, current instId pending buy orders: {param6} contracts, other contracts of the same instFamily: {param7} contracts).
51004	200	Order failed. For cross buy/sell mode of {param0} and instFamily {param1}, the sum of current sell order size, current instId position quantity, current instId pending sell orders, and other contracts of the same instFamily can't be more than {param2}(contracts) which is the maximum position amount under current leverage. Please lower the leverage or use a new sub-account to place the order again (current leverage: {param3}×, current sell order size: {param4} contracts, current instId position quantity: {param5} contracts, current instId pending sell orders: {param6} contracts, other contracts of the same instFamily: {param7} contracts).
51004	200	Order amendment failed. For isolated long/short mode of {param0}, the sum of increment order size by amendment, position quantity in the same direction, and pending orders in the same direction can't be more than {param1}(contracts) which is the maximum position amount under current leverage. Please lower the leverage or use a new sub-account to place the order again (current leverage: {param2}×, increment order size by amendment: {param3} contracts, position quantity in the same direction: {param4} contracts, pending orders in the same direction: {param5} contracts).
51004	200	Order amendment failed. For cross long/short mode of {param0}, the sum of increment order size by amendment, position quantity in the long and short directions, and pending orders in the long and short directions can't be more than {param1}(contracts) which is the maximum position amount under current leverage. Please lower the leverage or use a new sub-account to place the order again (current leverage: {param2}×, increment order size by amendment: {param3} contracts, position quantity in the long and short directions: {param4} contracts, pending orders in the same direction: {param5} contracts).
51004	200	Order amendment failed. For cross buy/sell mode of {param0} and instFamily {param1}, the sum of increment order size by amendment, current instId position quantity in the long and short directions, current instId pending orders in the long and short directions, and other contracts of the same instFamily can't be more than {param2}(contracts) which is the maximum position amount under current leverage. Please lower the leverage or use a new sub-account to place the order again (current leverage: {param3}×, increment order size by amendment: {param4} contracts, current instId position quantity in the long and short directions: {param5} contracts, current instId pending orders in the long and short directions: {param6} contracts, other contracts of the same instFamily: {param7} contracts).
51004	200	Order amendment failed. For buy/sell mode of {param0}, the sum of increment order size by amending current buy order, position quantity, and pending buy orders can't be more than {param1}(contracts) which is the maximum position amount under current leverage. Please lower the leverage or use a new sub-account to place the order again (current leverage: {param2}×, increment order size by amending current buy order: {param3} contracts, position quantity: {param4} contracts, pending buy orders: {param5} contracts).
51004	200	Order amendment failed. For buy/sell mode of {param0}, the sum of increment order size by amending current sell order, position quantity, and pending sell orders can't be more than {param1}(contracts) which is the maximum position amount under current leverage. Please lower the leverage or use a new sub-account to place the order again (current leverage: {param2}×, increment order size by amending current sell order: {param3} contracts, position quantity: {param4} contracts, pending sell orders: {param5} contracts).
51004	200	Order amendment failed. For cross buy/sell mode of {param0} and instFamily {param1}, the sum of increment order size by amending current buy order, current instId position quantity, current instId pending buy orders, and other contracts of the same instFamily can't be more than {param2}(contracts) which is the maximum position amount under current leverage. Please lower the leverage or use a new sub-account to place the order again (current leverage: {param3}×, increment order size by amending current buy order: {param4} contracts, current instId position quantity: {param5} contracts, current instId pending buy orders: {param6} contracts, other contracts of the same instFamily: {param7} contracts).

Error Code	HTTP Status code	Error Message
51004	200	Order amendment failed. For cross buy/sell mode of {param0} and instFamily {param1}, the sum of increment order size by amending current sell order, current instId position quantity, current instId pending sell orders, and other contracts of the same instFamily can't be more than {param2}(contracts) which is the maximum position amount under current leverage. Please lower the leverage or use a new sub-account to place the order again (current leverage: {param3}x, increment order size by amending current sell order: {param4} contracts, current instId position quantity: {param5} contracts, current instId pending sell orders: {param6} contracts, other contracts of the same instFamily: {param7} contracts).
51005	200	Your order amount exceeds the max order amount.
51006	200	Order price is not within the price limit (max buy price: {param0} , min sell price: {param1} )
51007	200	Order failed. Please place orders of at least 1 contract or more.
51008	200	Order failed. Insufficient {param0} balance in account
51008	200	Order failed. Insufficient {param0} margin in account
51008	200	Order failed. Insufficient {param0} balance in account, and Auto Borrow is not enabled
51008	200	Order failed. Insufficient {param0} margin in account and auto-borrow is not enabled (Portfolio margin mode can try IOC orders to lower the risks)
51008	200	Insufficient {param0} available as your borrowing amount exceeds tier limit. Lower leverage appropriately. New and pending limit orders need borrowings {param1}, remaining quota {param2}, total limit {param3}, in use {param4}.
51008	200	Order failed. Exceeds {param0} borrow limit (Limit of master account plus the allocated VIP quota for the current account) (Existing pending orders and the new order are required to borrow {param1}, Remaining limit {param2}, Limit {param3}, Limit used {param4})
51008	200	Order failed. Insufficient {param0} borrowing quota results in an insufficient amount available to borrow.
51008	200	Order failed. Insufficient {param0} available in loan pool to borrow.
51008	200	Order failed. Insufficient account balance, and the adjusted equity in <span style="border: 1px solid #ccc; padding: 2px;">usd</span> is less than IMR (Portfolio margin mode can try IOC orders to lower the risks)
51008	200	Order failed. The order didn't pass delta verification because if the order were to succeed, the change in adjEq would be smaller than the change in IMR. Increase adjEq or reduce IMR (Portfolio margin mode can try IOC orders to lower the risks)
51008	200	Order failed. Your available margin (in USD) is too low. It excludes crypto ({param0}) exceeding the platform's collateral limit.
51008	200	Order failed. Your available balance is insufficient. It excludes crypto ({param0}) exceeding the platform's collateral limit.
51008	200	Order failed. Your available {param0} balance is insufficient, and your available margin (in USD) is too low for borrowing. This margin excludes crypto ({param1}) exceeding the platform's collateral limit.
51008	200	Order failed. Your main account's {param0} borrowing quota is insufficient.
51009	200	Order blocked. Please contact customer support for assistance.
51010	200	Request unsupported under current account mode
51011	200	Order ID already exists.
51012	200	Token doesn't exist.
51014	200	Index doesn't exist.
51015	200	Instrument ID doesn't match instrument type.
51016	200	Client order ID already exists.

Error Code	HTTP Status code	Error Message
51017	200	Loan amount exceeds borrowing limit.
51018	200	Users with options accounts cannot hold net short positions.
51019	200	No net long positions can be held under cross margin mode in options.
51020	200	Your order should meet or exceed the minimum order amount.
51021	200	The pair or contract is not yet listed
51022	200	Contract suspended.
51023	200	Position doesn't exist.
51024	200	Trading account is blocked.
51024	200	In accordance with the terms of service, we regret to inform you that we cannot provide services for you. If you have any questions, please contact our customer support.
51024	200	According to your request, this account has been frozen. If you have any questions, please contact our customer support.
51024	200	Your account has recently changed some security settings. To protect the security of your funds, this action is not allowed for now. If you have any questions, please contact our customer support.
51024	200	You have withdrawn all assets in the account. To protect your personal information, the account has been permanently frozen. If you have any questions, please contact our customer support.
51024	200	Your identity could not be verified. To protect the security of your funds, this action is not allowed. Please contact our customer support.
51024	200	Your verified age doesn't meet the requirement. To protect the security of your funds, we cannot proceed with your request. Please contact our customer support.
51024	200	In accordance with the terms of service, trading is currently unavailable in your verified country or region. Close all open positions or contact customer support if you have any questions.
51024	200	In accordance with the terms of service, multiple account is not allowed. To protect the security of your funds, this action is not allowed. Please contact our customer support.
51024	200	Your account is in judicial freezing, and this action is not allowed for now. If you have any questions, please contact our customer support.
51024	200	Based on your previous requests, this action is not allowed for now. If you have any questions, please contact our customer support.
51024	200	Your account has disputed deposit orders. To protect the security of your funds, this action is not allowed for now. Please contact our customer support.
51024	200	Unable to proceed. Please resolve your existing P2P disputes first.
51024	200	Your account might have compliance risk. To protect the security of your funds, this action is not allowed for now. Please contact our customer support.
51024	200	Based on your trading requests, this action is not allowed for now. If you have any questions, please contact our customer support.
51024	200	Your account has triggered risk control. This action is not allowed for now. Please contact our customer support.
51024	200	This account is temporarily unavailable. Please contact our customer support.
51024	200	Withdrawal function of this account is temporarily unavailable. Please contact our customer support.

Error Code	HTTP Status code	Error Message
51024	200	Transfer function of this account is temporarily unavailable. Please contact our customer support.
51024	200	You violated the "Fiat Trading Rules" when you were doing fiat trade, so we'll no longer provide fiat trading-related services for you. The deposit and withdrawal of your account and other trading functions will not be affected.
51024	200	Please kindly check your mailbox and reply to emails from the verification team.
51024	200	According to your request, this account has been closed. If you have any questions, please contact our customer support.
51024	200	Your account might have security risk. To protect the security of your funds, this action is not allowed for now. Please contact our customer support.
51024	200	Your account might have security risk. Convert is now unavailable. Please contact our customer support.
51024	200	Unable to proceed due to account restrictions. We've sent an email to your OKX registered email address regarding this matter, or you can contact customer support via Chat with AI chatbot on our support center page.
51024	200	In accordance with the terms of service, trading is currently unavailable in your verified country or region. Cancel all orders or contact customer support if you have any questions.
51024	200	In accordance with the terms of service, trading is not available in your verified country. If you have any questions, please contact our customer support.
51024	200	This product isn't available in your country or region due to local laws and regulations. If you don't reside in this area, you may continue using OKX Exchange products with a valid government-issued ID.
51024	200	Please note that you may not be able to transfer or trade in the first 30 minutes after establishing custody trading sub-accounts. Please kindly wait and try again later.
51024	200	Feature unavailable. Complete Advanced verification to access this feature.
51024	200	You can't trade or deposit now. Update your personal info to restore full account access immediately.
51024	200	Sub-accounts exceeding the limit aren't allowed to open new positions and can only reduce or close existing ones. Please try again with a different account.
51025	200	Order count exceeds the limit.
51026	200	Instrument type doesn't match underlying index.
51027	200	Contract expired.
51028	200	Contract under delivery.
51029	200	Contract is being settled.
51030	200	Funding fee is being settled.
51031	200	This order price is not within the closing price range.
51032	200	Closing all the positions at the market price.
51033	200	The total amount per order for this pair has reached the upper limit.
51034	200	Fill rate exceeds the limit that you've set. Please reset the market maker protection to inactive for new trades.
51035	200	This account doesn't have permission to submit MM quote order.
51036	200	Only options instrument of the PM account supports MMP orders.
51042	200	Under the Portfolio margin account, users can only place market maker protection orders in cross margin mode in Options.

Error Code	HTTP Status code	Error Message
51043	200	This isolated position doesn't exist.
59509	200	Account does not have permission to reset MMP status
51037	200	This account only supports placing IOC orders to reduce account risk.
51038	200	IOC order already exists under the current risk module.
51039	200	Leverage cannot be adjusted for the cross positions of Expiry Futures and Perpetual Futures under the PM account.
51040	200	Cannot adjust margins for long isolated options positions
51041	200	Portfolio margin account only supports the Buy/Sell mode.
51044	200	The order type {param0}, {param1} is not allowed to set stop loss and take profit
51046	200	The take profit trigger price must be higher than the order price
51047	200	The stop loss trigger price must be lower than the order price
51048	200	The take profit trigger price must be lower than the order price
51049	200	The stop loss trigger price must be higher than the order price
51050	200	The take profit trigger price must be higher than the best ask price
51051	200	The stop loss trigger price must be lower than the best ask price
51052	200	The take profit trigger price must be lower than the best bid price
51053	200	The stop loss trigger price must be higher than the best bid price
51054	500	Request timed out. Please try again.
51055	200	Futures Grid is not available in Portfolio Margin mode
51056	200	Action not allowed
51057	200	This bot isn't available in current account mode. Switch mode in Settings > Account mode to continue.
51058	200	No available position for this algo order
51059	200	Strategy for the current state does not support this operation
51063	200	OrdId does not exist
51065	200	algoClOrdId already exists.
51066	200	Market orders unavailable for options trading. Place a limit order to close position.
51068	200	{param0} already exists within algoClOrdId and attachAlgoClOrdId.
51069	200	The option contracts related to current {param0} do not exist
51070	200	You do not meet the requirements for switching to this account mode. Please upgrade the account mode on the OKX website or App
51071	200	You've reached the maximum limit for tag level cancel all after timers.
51072	200	As a spot lead trader, you need to set tdMode to spot_isolated when buying the configured lead trade pairs.

Error Code	HTTP Status code	Error Message
51073	200	As a spot lead trader, you need to use '/copytrading/close-subposition' for selling assets through lead trades
51074	200	Only the tdMode for lead trade pairs configured by spot lead traders can be set to 'spot_isolated'
51075	200	Order modification failed. You can only modify the price of sell orders in spot copy trading.
51076	200	TP/SL orders in Split TPs only support one-way TP/SL. You can't use slTriggerPx&slOrdPx and tpTriggerPx&tpOrdPx at the same time.
51077	200	Setting multiple TP and cost-price SL orders isn't supported for spot and margin trading.
51078	200	You are a lead trader. Split TPs are not supported.
51079	200	The number of TP orders with Split TPs attached in a same order cannot exceed {param0}
51080	200	Take-profit trigger price types (tpTriggerPxType) must be the same in an order with Split TPs attached
51081	200	Take-profit trigger prices (tpTriggerPx) cannot be the same in an order with Split TPs attached
51082	200	TP trigger prices (tpOrdPx) in one order with multiple TPs must be market prices.
51083	200	The total size of TP orders with Split TPs attached in a same order should equal the size of this order
51084	200	The number of SL orders with Split TPs attached in a same order cannot exceed {param0}
51085	200	The number of TP orders cannot be less than 2 when cost-price SL is enabled (amendPxOnTriggerType set as 1) for Split TPs
51086	200	The number of orders with Split TPs attached in a same order cannot exceed {param0}
51538	200	You need to use attachAlgoOrds if you used attachAlgoOrds when placing an order. attachAlgoOrds is not supported if you did not use attachAlgoOrds when placing this order.
51539	200	attachAlgoid or attachAlgoCIOrid cannot be identical when modifying any TP/SL within your split TPs order
51527	200	Order modification failed. At least 1 of the attached TP/SL orders does not exist.
51087	200	Listing canceled for this crypto
51088	200	You can only place 1 TP/SL order to close an entire position
51089	200	The size of the TP order among split TPs attached cannot be empty
51090	200	You can't modify the amount of an SL order placed with a TP limit order.
51091	200	All TP orders in one order must be of the same type.
51092	200	TP order prices (tpOrdPx) in one order must be different.
51093	200	TP limit order prices (tpOrdPx) in one order can't be -1 (market price).
51094	200	You can't place TP limit orders in spot, margin, or options trading.
51095	200	To place TP limit orders at this endpoint, you must place an SL order at the same time.
51096	200	cxlOnClosePos needs to be true to place a TP limit order
51098	200	You can't add a new TP order to an SL order placed with a TP limit order.
51099	200	You can't place TP limit orders as a lead trader.
51178	200	tpTriggerPx&tpOrdPx or slTriggerPx&slOrdPx can't be empty when using attachAlgoCIOrid.

Error Code	HTTP Status code	Error Message
51100	200	Unable to place order. Take profit/Stop loss conditions cannot be added to reduce-only orders.
51101	200	Order failed, the sz of the current order can't be more than {param0} (contracts).
51102	200	Order failed, the number of pending orders for this instId can't be more than {param0} (orders)
51103	200	Order failed, the number of pending orders across all instIds under the {param0} current instFamily can't be more than {param1} (orders)
51104	200	Order failed, the aggregated contract quantity for all pending orders across all instIds under the {param0} current instFamily can't be more than {param1} (contracts)
51105	200	Order failed, the maximal sum of position quantity and pending orders quantity with the same direction for current instId can't be more than {param0} (contracts)
51106	200	Order failed, the maximal sum of position quantity and pending orders quantity with the same direction across all instIds under the {param0} current instFamily can't be more than {param1} (contracts)
51107	200	Order failed, the maximal sum of position quantity and pending orders quantity in both directions across all instIds under the {param0} current instFamily can't be more than {param1} (contracts)
51108	200	Positions exceed the limit for closing out with the market price.
51109	200	No available offers.
51110	200	You can only place a limit order after Call Auction has started.
51111	200	Maximum {param0} orders can be placed in bulk.
51112	200	Close order size exceeds available size for this position.
51113	429	Market-price liquidation requests too frequent.
51115	429	Cancel all pending close-orders before liquidation.
51116	200	Order price or trigger price exceeds {param0}.
51117	200	Pending close-orders count exceeds limit.
51120	200	Order amount is less than {param0}, please try again.
51121	200	Order quantity must be a multiple of the lot size.
51122	200	Order price should higher than the min price {param0}
51123	200	Min price increment is null.
51124	200	You can only place limit orders during call auction.
51125	200	Currently there are pending reduce + reverse position orders in margin trading. Please cancel all pending reduce + reverse position orders and continue.
51126	200	Currently there are pending reduce only orders in margin trading. Please cancel all pending reduce only orders and continue.
51127	200	Available balance is 0.
51128	200	Multi-currency margin accounts cannot do cross-margin trading.
51129	200	The value of the position and buy order has reached the position limit. No further buying is allowed.
51130	200	Fixed margin currency error.

Error Code	HTTP Status code	Error Message
51131	200	Insufficient balance.
51132	200	Your position amount is negative and less than the minimum trading amount.
51133	200	Reduce-only feature is unavailable for spot transactions in multi-currency margin accounts.
51134	200	Closing failed. Please check your margin holdings and pending orders. Turn off the Reduce-only to continue.
51135	200	Your closing price has triggered the limit price, and the max buy price is {param0}.
51136	200	Your closing price has triggered the limit price, and the min sell price is {param0}.
51137	200	The highest price limit for buy orders is {param0}.
51138	200	The lowest price limit for sell orders is {param0}.
51139	200	Reduce-only feature is unavailable for the spot transactions by spot mode.
51140	200	Purchase failed due to insufficient sell orders, please try later.
51142	200	There is no valid quotation in the market, and the order cannot be filled in USDT mode, please try to switch to currency mode
51143	200	Insufficient conversion amount
51144	200	Please use {param0} for closing.
51147	200	To trade options, make sure you have more than 10,000 USD worth of assets in your trading account first, then activate options trading
51148	200	Failed to place order. The new order may execute an opposite trading direction of your existing reduce-only positions. Cancel or edit pending orders to continue order
51149	500	Order timed out. Please try again. (does not mean that the request was successful or failed, please check the request result)
51150	200	The precision of the number of trades or the price exceeds the limit.
51152	200	Unable to place an order that mixes automatic buy with automatic repayment or manual operation in Quick margin mode.
51153	200	Unable to borrow manually in Quick margin mode. The amount you entered exceeds the upper limit.
51154	200	Unable to repay manually in Quick margin mode. The amount you entered exceeds your available balance.
51155	200	Trading of this pair or contract is restricted due to local compliance requirements.
51158	200	Manual transfer unavailable. To proceed, please switch to Quick margin mode (isoMode = quick_margin)
51164	200	As lead trader, you can't switch to portfolio margin mode.
51169	200	Order failed because you don't have any positions in this direction for this contract to reduce or close.
51170	200	Failed to place order. A reduce-only order can't be the same trading direction as your existing positions.
51171	200	Failed to edit order. The edited order may execute an opposite trading direction of your existing reduce-only positions. Cancel or edit pending orders to continue.
51173	200	Unable to close all at market price. Your current positions don't have any liabilities.
51174	200	Order failed, number of pending orders for {param0} exceed the limit of {param1}.
51175	200	Parameters {param0} {param1} and {param2} cannot be empty at the same time

Error Code	HTTP Status code	Error Message
51176	200	Only one parameter can be filled among Parameters {param0} {param1} and {param2}
51177	200	Unavailable to amend {param1} because the price type of the current options order is {param0}
51179	200	Unavailable to place options orders using {param0} in simple mode
51180	200	The range of {param0} should be ({param1}~{param2})
51181	200	ordType must be limit while placing {param0} orders
51182	200	The total number of pending orders under price types pxUsd and pxVol for the current account cannot exceed {param0}
51185	200	The maximum value allowed per order is {maxOrderValue} USD
51186	200	Order failed. The leverage for {param0} in your current margin mode is {param1}x, which exceeds the platform limit of {param2}x.
51187	200	Order failed. For {param0} {param1} in your current margin mode, the sum of your current order amount, position sizes, and open orders is {param2} contracts, which exceeds the platform limit of {param3} contracts. Reduce your order amount, cancel orders, or close positions.
51192	200	The {param0} price corresponding to the IV level you entered is lower than the minimum allowed selling price of {param1} {param2}. Enter a higher IV level.
51193	200	The {param0} price corresponding to the IV level you entered is higher than the maximum allowed buying price of {param1} {param2}. Enter a lower IV level.
51194	200	The {param0} price corresponding to the USD price you entered is lower than the minimum allowed selling price of {param1} {param2}. Enter a higher USD price.
51195	200	The {param0} price corresponding to the USD price you entered is higher than the maximum allowed buying price of {param1} {param2}. Enter a lower USD price.
51196	200	You can only place limit orders during the pre-quote phase.
51197	200	You can only place limit orders after the pre-quote phase begins.
51201	200	The value of a market order can't exceed {param0}.
51202	200	Market order amount exceeds the maximum amount.
51203	200	Order amount exceeds the limit {param0}.
51204	200	The price for the limit order cannot be empty.
51205	200	Reduce Only is not available.
51206	200	Please cancel the Reduce Only order before placing the current {param0} order to avoid opening a reverse position.
51207	200	Trading amount exceeds the limit, and can't be all closed at the market price. You can try closing the position manually in batches.
51209	200	Order failed because your current status only allows placing delta-reducing orders. Increase your account equity or place an order that would reduce delta when filled to proceed.
51210	200	Order failed because your current status only allows placing one order per underlying. Cancel other orders in the underlying to proceed.
51211	200	Order failed because your current status only allows placing delta-reducing orders. Increase your account equity or place an order that would reduce delta when filled to proceed.
51212	200	Order failed because your current status doesn't support batch orders. Place a single order to proceed.

Error Code	HTTP Status code	Error Message
51213	200	Order failed because your current status only allows placing one order per underlying. Cancel other orders in the underlying to proceed.
51214	200	Failed to modify order because your current status only allows placing delta-reducing orders. Increase your account equity or place an order that would reduce delta when filled to proceed.
51220	200	Lead and follow bots only support "Sell" or "Close all positions" when bot stops
51221	200	The profit-sharing ratio must be between 0% and 30%
51222	200	Profit sharing isn't supported for this type of bot
51223	200	Only lead bot creators can set profit-sharing ratio
51224	200	Profit sharing isn't supported for this crypto pair
51225	200	Instant trigger isn't available for follow bots
51226	200	Editing parameters isn't available for follow bots
51250	200	Algo order price is out of the available range.
51251	200	Bot order type error occurred when placing iceberg order
51252	200	Algo order amount is out of the available range.
51253	200	Average amount exceeds the limit of per iceberg order.
51254	200	Iceberg average amount error occurred.
51255	200	Limit of per iceberg order: Total amount/1000 < x <= Total amount.
51256	200	Iceberg order price variance error.
51257	200	Trailing stop order callback rate error. The callback rate should be {min}< x <= {max} %.
51258	200	Trailing stop order placement failed. The trigger price of a sell order must be higher than the last transaction price.
51259	200	Trailing stop order placement failed. The trigger price of a buy order must be lower than the last transaction price.
51260	200	Maximum of {param0} pending trailing stop orders can be held at the same time.
51261	200	Each user can hold up to {param0} pending stop orders at the same time.
51262	200	Maximum {param0} pending iceberg orders can be held at the same time.
51263	200	Maximum {param0} pending time-weighted orders can be held at the same time.
51264	200	Average amount exceeds the limit of per time-weighted order.
51265	200	Time-weighted order limit error.
51267	200	Time-weighted order strategy initiative rate error.
51268	200	Time-weighted order strategy initiative range error.
51269	200	Time-weighted order interval error. Interval must be {min}< x <= {max} %.
51270	200	The limit of time-weighted order price variance is 0 < x <= 1%.
51271	200	Sweep ratio must be 0 < x <= 100%.

Error Code	HTTP Status code	Error Message
51272	200	Price variance must be $0 < x \leq 1\%$ .
51273	200	Total amount must be greater than {param0}.
51274	200	Total quantity of time-weighted order must be larger than single order limit.
51275	200	The amount of single stop-market order cannot exceed the upper limit.
51276	200	Prices cannot be specified for stop market orders.
51277	200	TP trigger price cannot be higher than the last price.
51278	200	SL trigger price cannot be lower than the last price.
51279	200	TP trigger price cannot be lower than the last price.
51280	200	SL trigger price cannot be higher than the last price.
51281	200	Trigger order do not support the tgtCcy parameter.
51282	200	The range of Price variance is {param0}~{param1}
51283	200	The range of Time interval is {param0}~{param1}
51284	200	The range of Average amount is {param0}~{param1}
51285	200	The range of Total amount is {param0}~{param1}
51286	200	The total amount should not be less than {param0}
51287	200	This bot doesn't support current instrument
51288	200	Bot is currently stopping. Do not make multiple attempts to stop.
51289	200	Bot configuration does not exist. Please try again later
51290	200	The Bot engine is being upgraded. Please try again later
51291	200	This Bot does not exist or has been stopped
51292	200	This Bot type does not exist
51293	200	This Bot does not exist
51294	200	This Bot cannot be created temporarily. Please try again later
51295	200	Portfolio margin account does not support ordType {param0} in Trading bot mode
51298	200	Trigger orders are not available in the net mode of Expiry Futures and Perpetual Futures
51299	200	Order did not go through. You can hold a maximum of {param0} orders of this type.
51300	200	TP trigger price cannot be higher than the mark price
51302	200	SL trigger price cannot be lower than the mark price
51303	200	TP trigger price cannot be lower than the mark price
51304	200	SL trigger price cannot be higher than the mark price
51305	200	TP trigger price cannot be higher than the index price

Error Code	HTTP Status code	Error Message
51306	200	SL trigger price cannot be lower than the index price
51307	200	TP trigger price cannot be lower than the index price
51308	200	SL trigger price cannot be higher than the index price
51309	200	Cannot create trading bot during call auction
51310	200	Strategic orders with Iceberg and TWAP order type are not supported when margins are self-transferred in isolated mode.
51311	200	Failed to place trailing stop order. Callback rate should be within {min}<x<={max}
51312	200	Failed to place trailing stop order. Order amount should be within {min}<x<={max}
51313	200	Manual transfer in isolated mode does not support bot trading
51317	200	Trigger orders are not available by margin
51327	200	closeFraction is only available for Expiry Futures and Perpetual Futures
51328	200	closeFraction is only available for reduceOnly orders
51329	200	closeFraction is only available in NET mode
51330	200	closeFraction is only available for stop market orders
51331	200	closeFraction is only available for close position orders
51332	200	closeFraction is not applicable to Portfolio Margin
51333	200	Close position order in hedge-mode or reduce-only order in one-way mode cannot attach TPSL
51340	200	Used margin must be greater than {0}{1}
51341	200	Position closing not allowed
51342	200	Closing order already exists. Please try again later
51343	200	TP price must be less than the lower price
51344	200	SL price must be greater than the upper price
51345	200	Policy type is not grid policy
51346	200	The highest price cannot be lower than the lowest price
51347	200	No profit available
51348	200	Stop loss price must be less than the lower price in the range.
51349	200	Take profit price must be greater than the highest price in the range.
51350	200	No recommended parameters
51351	200	Single income must be greater than 0
51352	200	You can have {0} to {1} trading pairs
51353	200	Trading pair {0} already exists
51354	200	The percentages of all trading pairs should add up to 100%

Error Code	HTTP Status code	Error Message
51355	200	Select a date within {0} - {1}
51356	200	Select a time within {0} - {1}
51357	200	Select a time zone within {0} - {1}
51358	200	The investment amount of each crypto must be greater than {amount}
51359	200	Recurring buy not supported for the selected crypto {0}
51370	200	The range of lever is {0}~{1}
51380	200	Market conditions do not meet the strategy running configuration. You can try again later or adjust your tp/sl configuration.
51381	200	Per grid profit ratio must be larger than 0.1% and less or equal to 10%
51382	200	Stop triggerAction is not supported by the current strategy
51383	200	The min_price is lower than the last price
51384	200	The trigger price must be greater than the min price
51385	200	The take profit price needs to be greater than the min price
51386	200	The min price needs to be greater than 1/2 of the last price
51387	200	Stop loss price must be less than the bottom price
51388	200	This Bot is in running status
51389	200	Trigger price should be lower than {0}
51390	200	Trigger price should be lower than the TP price
51391	200	Trigger price should be higher than the SL price
51392	200	TP price should be higher than the trigger price
51393	200	SL price should be lower than the trigger price
51394	200	Trigger price should be higher than the TP price
51395	200	Trigger price should be lower than the SL price
51396	200	TP price should be lower than the trigger price
51397	200	SL price should be higher than the trigger price
51398	200	Current market meets the stop condition. The bot cannot be created.
51399	200	Max margin under current leverage: {amountLimit} {quoteCurrency}. Enter a smaller amount and try again.
51400	200	Order cancellation failed as the order has been filled, canceled or does not exist.
51400	200	Cancellation failed as the order does not exist. (Only applicable to Nitro Spread)
51401	200	Cancellation failed as the order is already canceled. (Only applicable to Nitro Spread)
51402	200	Cancellation failed as the order is already completed. (Only applicable to Nitro Spread)
51403	200	Cancellation failed as the order type doesn't support cancellation.

Error Code	HTTP Status code	Error Message
51404	200	Order cancellation unavailable during the second phase of call auction.
51405	200	Cancellation failed as you don't have any pending orders.
51406	400	Canceled - order count exceeds the limit {param0}.
51407	200	Either order ID or client order ID is required.
51408	200	Pair ID or name doesn't match the order info.
51409	200	Either pair ID or pair name ID is required.
51410	200	Cancellation failed as the order is already in canceling status or pending settlement.
51411	200	Account does not have permission for mass cancellation.
51412	200	Cancellation timed out, please try again later.
51412	200	The order has been triggered and can't be canceled.
51413	200	Cancellation failed as the order type is not supported by endpoint.
51415	200	Unable to place order. Spot trading only supports using the last price as trigger price. Please select "Last" and try again.
51416	200	Order has been triggered and can't be canceled
51500	200	You must enter a price, quantity, or TP/SL
51501	400	Maximum {param0} orders can be modified.
51502	200	Unable to edit order: insufficient balance or margin.
51502	200	Order failed. Insufficient {param0} margin in account
51502	200	Order failed. Insufficient {param0} balance in account and Auto Borrow is not enabled
51502	200	Order failed. Insufficient {param0} margin in account and Auto Borrow is not enabled (Portfolio margin mode can try IOC orders to lower the risks)
51502	200	Order failed. The requested borrowing amount is larger than the available {param0} borrowing amount of your position tier. Existing pending orders and the new order need to borrow {param1}, remaining quota {param2}, total quota {param3}, used {param4}
51502	200	Order failed. The requested borrowing amount is larger than the available {param0} borrowing amount of your position tier. Existing pending orders and the new order need to borrow {param1}, remaining quota {param2}, total quota {param3}, used {param4}
51502	200	Order failed. The requested borrowing amount is larger than the available {param0} borrowing amount of your main account and the allocated VIP quota. Existing pending orders and the new order need to borrow {param1}, remaining quota {param2}, total quota {param3}, used {param4}
51502	200	Order failed. Insufficient available borrowing amount in {param0} crypto pair
51502	200	Order failed. Insufficient available borrowing amount in {param0} loan pool
51502	200	Order failed. Insufficient account balance and the adjusted equity in USD is smaller than the IMR.
51502	200	Order failed. The order didn't pass delta verification. If the order succeeded, the change in adjEq would be smaller than the change in IMR. Increase adjEq or reduce IMR (Portfolio margin mode can try IOC orders to lower the risks)
51502	200	Order modification failed. Your main account's {param0} borrowing quota is insufficient.

Error Code	HTTP Status code	Error Message
51503	200	Your order has already been filled or canceled.
51503	200	Order modification failed as the order does not exist. (Only applicable to Nitro Spread)
51505	200	{instId} is not in call auction
51506	200	Order modification unavailable for the order type.
51507	200	You can only place market orders for this crypto at least 5 minutes after its listing.
51508	200	Orders are not allowed to be modified during the call auction.
51509	200	Modification failed as the order has been canceled. (Only applicable to Nitro Spread)
51510	200	Modification failed as the order has been completed. (Only applicable to Nitro Spread)
51511	200	Modification failed as the order price did not meet the requirement for Post Only.
51512	200	Failed to amend orders in batches. You cannot have duplicate orders in the same amend-batch-orders request.
51513	200	Number of modification requests that are currently in progress for an order cannot exceed 3 times.
51514	200	Order modification failed. The price length must be 32 characters or shorter.
51521	200	Failed to edit. Unable to edit reduce-only order because you don't have any positions of this contract.
51522	200	Failed to edit. A reduce-only order can't be in the same trading direction as your existing positions.
51523	200	Unable to modify the order price of a stop order that closes an entire position. Please modify the trigger price instead.
51524	200	Unable to modify the order quantity of a stop order that closes an entire position. Please modify the trigger price instead.
51525	200	Stop order modification is not available for quick margin
51526	200	Order modification unsuccessful. Take profit/Stop loss conditions cannot be added to or removed from stop orders.
51527	200	Order modification unsuccessful. The stop order does not exist.
51528	200	Unable to modify trigger price type
51529	200	Order modification unsuccessful. Stop order modification only applies to Expiry Futures and Perpetual Futures.
51530	200	Order modification unsuccessful. Take profit/Stop loss conditions cannot be added to or removed from reduce-only orders.
51531	200	Order modification unsuccessful. The stop order must have either take profit or stop loss attached.
51532	200	Your TP/SL can't be modified because it was partially triggered
51536	200	Unable to modify the size of the options order if the price type is pxUsd or pxVol.
51537	200	pxUsd or pxVol are not supported by non-options instruments
51543	200	When modifying take-profit or stop-loss orders for spot or margin trading, you can only adjust the price and quantity. Cancel the order and place a new one for other actions.
51600	200	Status not found.
51601	200	Order status and order id cannot exist at the same time.
51602	200	Either order status or order ID is required.

Error Code	HTTP Status code	Error Message
51603	200	Order does not exist.
51604	200	Initiate a download request before obtaining the hyperlink
51605	200	You can only download transaction data from the past 2 years
51606	200	Transaction data for the current quarter is not available
51607	200	Your previous download request is still being processed
51608	200	No transaction data found for the current quarter
51610	200	You can't download billing statements for the current quarter.
51611	200	You can't download billing statements for the current quarter.
51620	200	Only affiliates can perform this action
51621	200	The user isn't your invitee
51156	200	You're leading trades in long/short mode and can't use this API endpoint to close positions
51159	200	You're leading trades in buy/sell mode. If you want to place orders using this API endpoint, the orders must be in the same direction as your existing positions and open orders.
51162	200	You have {instrument} open orders. Cancel these orders and try again
51163	200	You hold {instrument} positions. Close these positions and try again
51165	200	The number of {instrument} reduce-only orders reached the upper limit of {upLimit}. Cancel some orders to proceed.
51166	200	Currently, we don't support leading trades with this instrument
51167	200	Failed. You have block trading open order(s), please proceed after canceling existing order(s).
51168	200	Failed. You have reduce-only type of open order(s), please proceed after canceling existing order(s)
51320	200	The range of coin percentage is {0}%-{1}%
51321	200	You're leading trades. Currently, we don't support leading trades with arbitrage, iceberg, or TWAP bots
51322	200	You're leading trades that have been filled at market price. We've canceled your open stop orders to close your positions
51323	200	You're already leading trades with take profit or stop loss settings. Cancel your existing stop orders to proceed
51324	200	As a lead trader, you hold positions in {instrument}. To close your positions, place orders in the amount that equals the available amount for closing
51325	200	As a lead trader, you must use market price when placing stop orders
51326	200	As a lead trader, you must use market price when placing orders with take profit or stop loss settings
51820	200	Request failed
51821	200	The payment method is not supported
51822	200	Quote expired
51823	200	Parameter {param} of buy/sell trading is inconsistent with the quotation
51824	200	You must bind a phone number to your account before using this feature

Error Code	HTTP Status code	Error Message
51825	200	You must bind an email to your account before using this feature
54000	200	Margin trading is not supported.
54001	200	Only Multi-currency margin account can be set to borrow coins automatically.
54004	200	Order placement or modification failed because one of the orders in the batch failed.
54005	200	Switch to isolated margin mode to trade pre-market expiry futures.
54006	200	Pre-market expiry future position limit is {posLimit}. Please cancel order or close position
54007	200	Instrument {instId} is not supported
54008	200	This operation is disabled by the 'mass cancel order' endpoint. Please enable it using this endpoint.
54009	200	The range of {param0} should be [{param1}, {param2}].
54011	200	Pre-market trading contracts are only allowed to reduce the number of positions within 1 hour before delivery. Please modify or cancel the order.
54012	200	Due to insufficient order book depth, we are now taking measures to protect your positions. Currently, you can only cancel orders, add margin to your positions, and place post-only orders. Your positions will not be liquidated. Trade will resume once order book depth returns to a safe level.
54018	200	Buy limit of {param0} USD exceeded. Your remaining limit is {param1} USD. (During the call auction)
54019	200	Buy limit of {param0} USD exceeded. Your remaining limit is {param1} USD. (After the call auction)
54024	200	Your order failed because you must enable {ccy} as collateral to trade expiry futures, perpetual futures, and options in cross-margin mode.
54025	200	Your order failed because you must enable {ccy} as collateral to trade margin, expiry futures, perpetual futures, and options in isolated margin mode.
54026	200	Your order failed because you must enable {ccy} and {ccy1} as collateral to trade the margin pair in isolated margin mode.
54027	200	Your order failed because you must enable {ccy} as collateral to trade options.
54028	200	Your order failed because you must enable {ccy} as collateral to trade spot in isolated margin mode.
54029	200	{param0} doesn't exist within {param1}.
54030	200	Order failed. Your total value of same-direction {param0} open positions and orders can't exceed {param1} USD or {param2} of the platform's open interest.
54031	200	Order failed. The {param1} USD open position limit for {param0} has been reached.
54035	200	Order failed. The platform has reached the collateral limit for this crypto, so you can only place reduce-only orders.
54036	200	You can't place fill or kill orders when self-trade prevention is set to both maker and taker orders.
54036	200	You can't place fill or kill orders when self-trade prevention is set to both maker and taker orders.
54070	200	The current function is not supported. Please update to the latest app version if using the app, or use the attachAlgoOrds array to place orders via Open API.
54071	200	Due to the platform system upgrade, this order no longer supports modifications. It is recommended to cancel and place a new order.
54072	200	This contract is currently view-only and not tradable.

Error Code	HTTP Status code	Error Message
54073	200	Couldn't place order, as {param0} is at risk of depegging. Switch settlement currencies and try again.
54074	200	Your settings failed as you have positions, bot or open orders for USD contracts.
54075	200	Cross-crypto contracts are currently not supported in this account mode.

#### DATA CLASS

Error Code	HTTP Status Code	Error Message
52000	200	No market data found.

#### FINANCE

Error Code from 51700 to 51799

Error Code	HTTP Status Code	Error Message
51720	200	Redeem error
51721	200	Cancel redeem error
51722	200	Redeem already complete
51723	200	Early redemption is not supported
51724	200	Redemption is currently not supported
51725	200	Cancellation is currently not supported
51726	200	Cancellation of subscriptions/redemptions is not supported
51727	200	The minimum subscription amount is {minUnit} {ccy}
51728	200	The subscription quantity is above the maximum limit
51729	200	This project has not reached the redemption date
51730	200	Sold out
51731	200	Product is currently suspended for purchase
51732	200	Required user KYC level not met
51733	200	User is under risk control
51734	200	User KYC Country is not supported
51735	200	Sub-account is not supported
51736	200	Insufficient {ccy} balance
51737	200	For security and compliance purposes, please complete the identity verification process to continue using our services.
51738	200	Your funding account is frozen.
51739	200	This function is unavailable temporarily
51750	200	The collateral cannot contain assets in the currency of the loan

Error Code	HTTP Status Code	Error Message
51751	200	The currency {ccy} does not support borrowing
51752	200	The currency {ccy} does not support collateralization
51753	200	The collateral does not include this asset
51754	200	There is currently no debt, no need to increase collateral
51755	200	The currency {ccy} operation is restricted
51756	200	Exceeding the maximum redeemable quantity
51757	200	The collateral amount should not be less than {minAmt}
51758	200	You can redeem at most {maxRedemptionAmount} {crypto}.
51759	200	You've exceed the temporary limit. Try again later.
51760	200	Flexible Loan isn't available to accounts using delta neutral strategy.
51761	200	Failed to verify sub-account strategy type.
51762	200	Redemption partially processed and can't be cancelled

## CONVERT

Error Code from 52900 to 52999

Error Code	HTTP Status Code	Error Message
52900	200	General invalid request
52901	200	Invalid base asset
52902	200	Invalid quote asset
52903	200	Invalid quote amount
52904	200	Invalid quote side
52905	200	Invalid quote price
52907	200	Order not found
52908	200	Invalid order ID
52909	200	Duplicate Client Order Id
52910	500	Service unavailable, please try again later
52911	500	RFQ service unavailable, please try again later
52912	500	Server timeout
52913	200	Trade rejected
52914	200	Insufficient available balance in trading account
52915	200	Cannot quote due to large amounts of RFQ and insufficient liquidity, please try again later
52916	200	Insufficient balance in funding account

Error Code	HTTP Status Code	Error Message
52917	200	RFQ quantity cannot be less than the lower limit
52918	200	RFQ quantity cannot be greater than the upper limit
52919	200	Parameter {param} of convert trading is inconsistent with the quotation
52920	200	Quantity of convert trading cannot exceed the quotation quantity
52921	200	Quote traded, please ask for quote again
52922	200	Quote expired, please ask for quote again
52923	200	Service unavailable. Try again later.
52924	200	Too many orders. Try again later.
52925	200	Duplicate client request ID
52926	200	{param0} has already expired
52927	200	No quote
52928	200	Quantity must be a multiple of the step size
52929	200	Disable Pay trading fees in quote currency, then try converting again.

## FUTURES

Error Code from 55000 to 55999

Error Code	HTTP Status Code	Error Message
55000	200	Cannot be transferred out within 30 minutes after delivery.

## SWAP

No

## OPTION

No

## FUNDING

Error Code from 58000 to 58999

Error Code	HTTP Status Code	Error Message
58002	200	Please activate Savings Account first.
58003	200	Savings does not support this currency type
58004	200	Account blocked.
58005	200	The {behavior} amount must be equal to or less than {minNum}
58006	200	Service unavailable for token {0}.
58007	200	Assets interface is currently unavailable. Try again later
58008	200	You do not have assets in this currency.

Error Code	HTTP Status Code	Error Message
58009	200	Crypto pair doesn't exist
58010	200	Chain {chain} isn't supported
58011	200	Due to local laws and regulations, our services are unavailable to unverified users in {region}. Please verify your account.
58012	200	Due to local laws and regulations, OKX does not support asset transfers to unverified users in {region}. Please make sure your recipient has a verified account.
58013	200	Withdrawals not supported yet, contact customer support for details
58014	200	Deposits not supported yet, contact customer support for details
58015	200	Transfers not supported yet, contact customer support for details
58016	200	The API can only be accessed and used by the trading team's main account
58100	200	The trading product triggers risk control, and the platform has suspended the fund transfer-out function with related users. Please wait patiently.
58101	200	Transfer suspended
58102	429	Rate limit reached. Please refer to API docs and throttle requests accordingly.
58103	200	This account transfer function is temporarily unavailable. Please contact customer service for details.
58104	200	Since your P2P transaction is abnormal, you are restricted from making fund transfers. Please contact customer support to remove the restriction.
58105	200	Since your P2P transaction is abnormal, you are restricted from making fund transfers. Please transfer funds on our website or app to complete identity verification.
58106	200	USD verification failed.
58107	200	Crypto verification failed.
58110	200	Transfers are suspended due to market risk control triggered by your {businessType} {instFamily} trades or positions. Please try again in a few minutes. Contact customer support if further assistance is needed.
58111	200	Fund transfers are unavailable while perpetual contracts are charging funding fees. Try again later.
58112	200	Transfer failed. Contact customer support for assistance
58113	200	Unable to transfer this crypto
58114	400	Transfer amount must be greater than 0
58115	200	Sub-account does not exist.
58116	200	Transfer exceeds the available amount.
58117	200	Transfer failed. Resolve any negative assets before transferring again
58119	200	{0} Sub-account has no permission to transfer out, please set first.
58120	200	Transfers are currently unavailable. Try again later
58121	200	This transfer will result in a high-risk level of your position, which may lead to forced liquidation. You need to re-adjust the transfer amount to make sure the position is at a safe level before proceeding with the transfer.

Error Code	HTTP Status Code	Error Message
58122	200	A portion of your spot is being used for Delta offset between positions. If the transfer amount exceeds the available amount, it may affect current spot-derivatives risk offset structure, which will result in an increased Maintenance Margin Requirement (MMR) rate. Please be aware of your risk level.
58123	200	The From parameter cannot be the same as the To parameter.
58124	200	Your transfer is being processed, transfer id:{trId}. Please check the latest state of your transfer from the endpoint (GET /api/v5/asset/transfer-state)
58125	200	Non-tradable assets can only be transferred from sub-accounts to main accounts
58126	200	Non-tradable assets can only be transferred between funding accounts
58127	200	Main account API key does not support current transfer 'type' parameter. Please refer to the API documentation.
58128	200	Sub-account API key does not support current transfer 'type' parameter. Please refer to the API documentation.
58129	200	{param} is incorrect or {param} does not match with 'type'
58131	200	For compliance, we're unable to provide services to unverified users. Verify your identity to make a transfer.
58132	200	For compliance, we're unable to provide services to users with Basic verification (Level 1). Complete Advanced verification (Level 2) to make a transfer.
58200	200	Withdrawal from {0} to {1} is currently not supported for this currency.
58201	200	Withdrawal amount exceeds daily withdrawal limit.
58202	200	The minimum withdrawal amount for NEO is 1, and the amount must be an integer.
58203	200	Please add a withdrawal address.
58204	200	Withdrawal suspended due to your account activity triggering risk control. Please contact customer support for assistance.
58205	200	Withdrawal amount exceeds the upper limit.
58206	200	Withdrawal amount is less than the lower limit.
58207	200	Withdrawal address isn't on the verified address list. (The format for withdrawal addresses with a label is "address:label".)
58208	200	Withdrawal failed. Please link your email.
58209	200	Sub-accounts don't support withdrawals or deposits. Please use your main account instead
58210	200	You can't proceed with withdrawal as we're unable to verify your identity. Please withdraw via our app or website instead.
58212	200	Withdrawal fee must be {0}% of the withdrawal amount
58213	200	The internal transfer address is illegal. It must be an email, phone number, or account name
58214	200	Withdrawals suspended due to {chainName} maintenance
58215	200	Withdrawal ID does not exist.
58216	200	Operation not allowed.
58217	200	Withdrawals are temporarily suspended for your account due to a risk detected in your withdrawal address. Contact customer support for assistance
58218	200	The internal withdrawal failed. Please check the parameters toAddr and areaCode.

Error Code	HTTP Status Code	Error Message
58219	200	You cannot withdraw crypto within 24 hours after changing your mobile number, email address, or Google Authenticator.
58220	200	Withdrawal request already canceled.
58221	200	The toAddr parameter format is incorrect, withdrawal address needs labels. The format should be "address:label".
58222	200	Invalid withdrawal address
58223	200	This is a contract address with higher withdrawal fees
58224	200	This crypto currently doesn't support on-chain withdrawals to OKX addresses. Withdraw through internal transfers instead
58225	200	Asset transfers to unverified users in {region} are not supported due to local laws and regulations.
58226	200	{chainName} is delisted and not available for crypto withdrawal.
58227	200	Withdrawal of non-tradable assets can be withdrawn all at once only
58228	200	Withdrawal of non-tradable assets requires that the API key must be bound to an IP
58229	200	Insufficient funding account balance to pay fees {fee} USDT
58230	200	According to the OKX compliance policy, you will need to complete your identity verification (Level 1) in order to withdraw
58231	200	The recipient has not completed personal info verification (Level 1) and cannot receive your transfer
58232	200	You've reached the personal information verification (L1) withdrawal limit, complete photo verification (L2) to increase the withdrawal limit
58233	200	For compliance, we're unable to provide services to unverified users. Verify your identity to withdraw.
58234	200	For compliance, the recipient can't receive your transfer yet. They'll need to verify their identity to receive your transfer.
58235	200	For compliance, we're unable to provide services to users with Basic verification (Level 1). Complete Advanced verification (Level 2) to withdraw.
58236	200	For compliance, a recipient with Basic verification (Level 1) is unable to receive your transfer. They'll need to complete Advanced verification (Level 2) to receive it.
58237	200	According to local laws and regulations, please provide accurate recipient information (rcvrlInfo). For the exchange address, please also provide exchange information and recipient identity information ({consentParameters}).
58238	200	Incomplete info. The info of the exchange and the recipient are required if you're withdrawing to an exchange platform.
58239	200	You can't withdraw to a private wallet via API. Please withdraw via our app or website instead.
58240	200	For security and compliance purposes, please complete the identity verification process to use our services. If you prefer not to verify, contact customer support for next steps. We're committed to ensuring a safe platform for users and appreciate your understanding.
58241	200	Due to local compliance requirements, internal withdrawal is unavailable
58242	200	The recipient can't receive your transfer due to their local compliance requirements
58243	200	Your recipient can't receive your transfer as they haven't made a cash deposit yet
58244	200	Make a cash deposit to proceed
58248	200	Due to local regulations, API withdrawal isn't allowed. Withdraw using OKX app or web.

Error Code	HTTP Status Code	Error Message
58249	200	API withdrawal for this currency is currently unavailable. Try withdrawing via our app or website.
58252	200	Withdrawal is restricted for 48h after your first TRY transaction for asset security.
58254	200	Due to local compliance requirements, please complete the digital signature or satoshi test via our app or website first before withdrawing to a private wallet via API.
58256	200	You can withdraw stablecoins valued at up to {dailyLimit} {symbol} every 24 hours. You can still withdraw {quotaRemaining} {symbol}.
58257	200	You can withdraw stablecoins valued at up to {monthlyLimit} {symbol} every month. You can still withdraw {quotaRemaining} {symbol}.
58258	200	You can withdraw crypto valued at up to {dailyLimit} {symbol} every 24 hours. You can still withdraw {quotaRemaining} {symbol}.
58300	200	Deposit-address count exceeds the limit.
58301	200	Deposit-address not exist.
58302	200	Deposit-address needs tag.
58303	200	Deposit for chain {chain} is currently unavailable
58304	200	Failed to create invoice.
58305	200	Unable to retrieve deposit address, please complete identity verification and generate deposit address first.
58306	200	According to the OKX compliance policy, you will need to complete your identity verification (Level 1) in order to deposit
58307	200	You've reached the personal information verification (L1) deposit limit, the excess amount has been frozen, complete photo verification (L2) to increase the deposit limit
58308	200	For compliance, we're unable to provide services to unverified users. Verify your identity to deposit.
58309	200	For compliance, we're unable to provide services to users with Basic verification (Level 1). Complete Advanced verification (Level 2) to deposit.
58310	200	Unable to create new deposit address, try again later
58350	200	Insufficient balance.
58351	200	Invoice expired.
58352	200	Invalid invoice.
58353	200	Deposit amount must be within limits.
58354	200	You have reached the daily limit of 10,000 invoices.
58355	200	Permission denied. Please contact your account manager.
58356	200	The accounts of the same node do not support the Lightning network deposit or withdrawal.
58358	200	The fromCcy parameter cannot be the same as the toCcy parameter.
58373	200	The minimum {ccy} conversion amount is {amount}
58400	200	Request Failed
58401	200	Payment method is not supported

Error Code	HTTP Status Code	Error Message
58402	200	Invalid payment account
58403	200	Transaction cannot be canceled
58404	200	ClientId already exists
58405	200	Withdrawal suspended
58406	200	Channel is not supported
58407	200	API withdrawal isn't allowed for this payment method. Withdraw using OKX app or web

## ACCOUNT

Error Code from 59000 to 59999

Error Code	HTTP Status Code	Error Message
59000	200	Settings failed. Close any open positions or orders before modifying settings.
59001	200	Switching unavailable as you have borrowings.
59002	200	Sub-account settings failed. Close any open positions, orders, or trading bots before modifying settings.
59004	200	Only IDs with the same instrument type are supported
59005	200	When margin is manually transferred in isolated mode, the value of the asset intially allocated to the position must be greater than 10,000 USDT.
59006	200	This feature is unavailable and will go offline soon.
59101	200	Leverage can't be modified. Please cancel all pending isolated margin orders before adjusting the leverage.
59102	200	Leverage exceeds the maximum limit. Please lower the leverage.
59103	200	Account margin is insufficient and leverage is too low. Please increase the leverage.
59104	200	The borrowed position has exceeded the maximum position of this leverage. Please lower the leverage.
59105	400	Leverage can't be less than {0}. Please increase the leverage.
59106	200	The max available margin corresponding to your order tier is {param0}. Please adjust your margin and place a new order.
59107	200	Leverage can't be modified. Please cancel all pending cross-margin orders before adjusting the leverage.
59108	200	Your account leverage is too low and has insufficient margins. Please increase the leverage.
59109	200	Account equity less than the required margin amount after adjustment. Please adjust the leverage.
59110	200	The instrument corresponding to this {param0} does not support the tgtCcy parameter.
59111	200	Leverage query isn't supported in portfolio margin account mode
59112	200	You have isolated/cross pending orders. Please cancel them before adjusting your leverage
59113	200	According to local laws and regulations, margin trading service is not available in your region. If your citizenship is at a different region, please complete KYC2 verification.
59114	200	According to local laws and regulations, margin trading services are not available in your region.

Error Code	HTTP Status Code	Error Message
59117	200	Cannot select more than {param0} crypto types
59118	200	Amount placed should greater than {param0}
59119	200	One-click repay is temporarily unavailable. Try again later.
59120	200	One-click convert is temporarily unavailable. Try again later.
59121	200	This batch is still under processing, please wait patiently.
59122	200	This batch has been processed
59123	200	{param0} order amount must be greater than {param1}
59124	200	The order amount of {param0} must be less than {param1}.
59125	200	{param0} doesn't support the current operation.
59132	200	Unable to switch. Please close or cancel all open orders and refer to the pre-check endpoint to stop any incompatible bots.
59133	200	Unable to switch due to insufficient assets for the chosen account mode.
59134	200	Unable to switch. Refer to the pre-check endpoint and close any incompatible positions.
59135	200	Unable to switch. Refer to the pre-check endpoint and adjust your trades from copy trading.
59136	200	Unable to switch. Pre-set leverage for all cross margin contract positions then try again.
59137	200	Lower leverage to {param0} or below for all cross margin contract positions and try again.
59138	200	Unable to switch due to a position tier check failure.
59139	200	Unable to switch due to a margin check failure.
59140	200	You can only repay with your collateral crypto.
59141	200	The minimum repayment amount is {param0}. Select more available crypto or increase your trading account balance.
59142	200	Instant repay failed. You can only repay borrowable crypto.
59200	200	Insufficient account balance.
59201	200	Negative account balance.
59202	200	No access to max opening amount in cross positions for PM accounts.
59300	200	Margin call failed. Position does not exist.
59301	200	Margin adjustment failed for exceeding the max limit.
59302	200	Margin adjustment failed due to pending close order. Please cancel any pending close orders.
59303	200	Insufficient available margin, add margin or reduce the borrowing amount
59304	200	Insufficient equity for borrowing. Keep enough funds to pay interest for at least one day.
59305	200	Use VIP loan first to set the VIP loan priority
59306	200	Your borrowing amount exceeds the max limit
59307	200	You are not eligible for VIP loans

Error Code	HTTP Status Code	Error Message
59308	200	Unable to repay VIP loan due to insufficient borrow limit
59309	200	Unable to repay an amount that exceeds the borrowed amount
59310	200	Your account does not support VIP loan
59311	200	Setup cannot continue. An outstanding VIP loan exists.
59312	200	{currency} does not support VIP loans
59313	200	Unable to repay. You haven't borrowed any \${ccy} (\${ccyPair}) in Quick margin mode.
59314	200	The current user is not allowed to return the money because the order is not borrowed
59315	200	viploan is upgrade now. Wait for 10 minutes and try again
59316	200	The current user is not allowed to borrow coins because the currency is in the order in the currency borrowing application.
59317	200	The number of pending orders that are using VIP loan for a single currency cannot be more than {maxNumber} (orders)
59319	200	You can't repay your loan order because your funds are in use. Make them available for full repayment.
59401	200	Holdings limit reached.
59402	200	No passed instIDs are in a live state. Please verify instIDs separately.
59410	200	You can only borrow this crypto if it supports borrowing and borrowing is enabled.
59411	200	Manual borrowing failed. Your account's free margin is insufficient.
59412	200	Manual borrowing failed. The amount exceeds your borrowing limit.
59413	200	You didn't borrow this crypto. No repayment needed.
59414	200	Manual borrowing failed. The minimum borrowing limit is {param0}.
59415	200	Manual Simple Borrow for {TOKEN} is temporarily disabled due to high utilization. Please retry later.
59500	200	Only the API key of the main account has permission.
59501	200	Each account can create up to 50 API keys
59502	200	This note name already exists. Enter a unique API key note name
59503	200	Each API key can bind up to 20 IP addresses
59504	200	Sub-accounts don't support withdrawals. Please use your main account for withdrawals.
59505	200	The passphrase format is incorrect.
59506	200	API key doesn't exist.
59507	200	The two accounts involved in a transfer must be 2 different sub-accounts under the same main account.
59508	200	The sub account of {param0} is suspended.
59509	200	Account doesn't have permission to reset market maker protection (MMP) status.
59510	200	Sub-account does not exist

Error Code	HTTP Status Code	Error Message
59512	200	Unable to set up permissions for ND broker subaccounts. By default, all ND subaccounts can transfer funds out.
59515	200	You are currently not on the custody whitelist. Please contact customer service for assistance.
59516	200	Please create the Copper custody funding account first.
59517	200	Please create the Komainu custody funding account first.
59518	200	You can't create a sub-account using the API; please use the app or web.
59519	200	You can't use this function/feature while it's frozen, due to: {freezereason}
59518	200	This account isn't eligible for delta neutral strategy.
59519	200	You must be VIP 1 or above to use delta neutral strategy.
59520	200	You can't use delta neutral strategy in spot or futures mode.
59521	200	Flexible Loan and delta neutral strategy can't be in use at the same time.
59522	200	You can't borrow and transfer or withdraw when using delta neutral strategy.
59523	200	You can't place orders or open positions in isolated mode and use delta neutral strategy at the same time.
59524	200	You can't trade options or open option positions and use delta neutral strategy at the same time.
59525	200	Some bots and copy trades can't be used at the same time as delta neutral strategy.
59526	200	Failed to switch strategy because your delta-to-equity ratio will exceed the threshold and trigger the transfer-out restriction after the switch. Lower your delta and try again.
59527	200	You must set all currencies as collateral when using delta neutral strategy.
59528	200	Failed to switch strategy because your account's {param0} borrowing in the targeted strategy will exceed the main account borrowing limit after the switch. Repay your liabilities and try again.
59601	200	Subaccount name already exists.
59603	200	Maximum number of subaccounts reached.
59604	200	Only the API key of the main account can access this API.
59606	200	Failed to delete sub-account. Transfer all sub-account funds to your main account before deleting your sub-account.
59608	200	Only Broker accounts have permission to access this API.
59609	200	Broker already exists
59610	200	Broker does not exist
59611	200	Broker unverified
59612	200	Cannot convert time format
59613	200	No escrow relationship established with the subaccount.
59614	200	Managed subaccount does not support this operation.
59615	200	The time interval between the Begin Date and End Date cannot be greater than 180 days.
59616	200	The Begin Date cannot be later than the End Date.

Error Code	HTTP Status Code	Error Message
59617	200	Sub-account created. Account level setup failed.
59618	200	Failed to create sub-account.
59619	200	This endpoint does not support ND sub accounts. Please use the dedicated endpoint supported for ND brokers.
59622	200	You're creating a sub-account for a non-existing or incorrect sub-account. Create a sub-account under the ND broker first or use the correct sub-account code.
59623	200	Couldn't delete the sub-account under the ND broker as the sub-account has one or more sub-accounts, which must be deleted first.
59648	200	Your modified spot-in-use amount is insufficient, which may lead to liquidation. Adjust the amount.
59649	200	Disabling spot-derivatives risk offset mode may increase the risk of liquidation. Adjust the size of your positions and ensure your maintenance margin ratio is safe.
59650	200	Switching your offset unit may increase the risk of liquidation. Adjust the size of your positions and ensure your maintenance margin ratio is safe.
59651	200	Enable spot-derivatives risk offset mode to set your spot-in-use amount.
59652	200	You can only set a spot-in-use amount for crypto that can be used as margin.
59658	200	{ccy} isn't supported as collateral.
59658	200	{ccy} and {ccy1} aren't supported as collateral.
59658	200	{ccy}, {ccy1}, and {ccy2} aren't supported as collateral.
59658	200	{ccy}, {ccy1}, {ccy2}, and {number} other crypto aren't supported as collateral.
59659	200	Failed to apply settings because you must also enable {ccy} to enable {ccy1} as collateral.
59660	200	Failed to apply settings because you must also disable {ccy} to disable {ccy1} as collateral.
59661	200	Failed to apply settings because you can't disable {ccy} as collateral.
59662	200	Failed to apply settings because of open orders or positions requiring {ccy} as collateral.
59662	200	Failed to apply settings because of open orders or positions requiring {ccy} and {ccy1} as collateral.
59662	200	Failed to apply settings because of open orders or positions requiring {ccy}, {ccy1}, and {ccy2} as collateral.
59662	200	Failed to apply settings because of open orders or positions requiring {ccy}, {ccy1}, {ccy2}, and {number} other crypto as collateral.
59664	200	Failed to apply settings because you have borrowings in {ccy}.
59664	200	Failed to apply settings because you have borrowings in {ccy} and {ccy1}.
59664	200	Failed to apply settings because you have borrowings in {ccy}, {ccy1}, and {ccy2}.
59664	200	Failed to apply settings because you have borrowings in {ccy}, {ccy1}, {ccy2}, and {number} other crypto.
59665	200	Failed to apply settings. Enable other cryptocurrencies as collateral to meet the position's margin requirements.
59666	200	Failed to apply settings because you can't enable and disable a crypto as collateral at the same time.
59668	200	Cancel isolated margin TP/SL, trailing, trigger, and chase orders or stop bots before adjusting your leverage.
59669	200	Cancel cross-margin TP/SL, trailing, trigger, and chase orders or stop bots before adjusting your leverage.

Error Code	HTTP Status Code	Error Message
59670	200	You have more than {param0} open orders for this trading pair. Cancel to reduce your orders to {param1} or fewer before adjusting your leverage.
59671	200	Auto-earn currently doesn't support {param0}.
59672	200	You can't modify your minimum lending APR when Auto-earn is off.
59673	200	You can't turn off Auto-earn within 24 hours of turning it on. Try again at {param0}.
59674	200	You can't borrow to transfer or withdraw when Auto-earn is on for this cryptocurrency.
59675	200	You've already turned on Auto-earn for {param0}.
59676	200	You can only use Auto-earn if your trading fee tier is {param0} or higher.
59678	200	Switch failed. Please cancel all existing spot orders and try again.
59679	200	Switch failed. Your account does not currently support this fee currency.
59683	200	Set this crypto as your collateral crypto before selecting it as your settlement currency.
59684	200	Borrowing isn't supported for this currency.
59686	200	This crypto can't be set as a settlement currency.

#### BLOCK TRADING AND SPREAD ORDERBOOK

Error Code from 70000

Error Code	HTTP Status Code	Error Message
70000	200	RFQ does not exist.
70001	200	Quote does not exist.
70002	200	Block trade does not exist.
70003	200	Public block trade does not exist.
70004	200	Invalid instrument {instId}
70005	200	The number of legs in RFQ cannot exceed maximum value.
70006	200	Does not meet the minimum asset requirement.
70007	200	Underlying index {instFamily} does not exist under instType {instType}.
70008	200	Operation failed under MMP status.
70009	200	Data must have at least 1 valid element.
70010	200	Timestamp parameters need to be in Unix timestamp format in milliseconds.
70011	200	Duplicate setting for instType {instType}.
70012	200	Duplicate setting for underlying/instId {instId} under the same instType {instType}.
70013	200	endTs needs to be bigger than or equal to beginTs.
70014	200	It's not allowed to have includeAll=True for all the instType.

Error Code	HTTP Status Code	Error Message
70015	200	In order to trade this product, you need to complete advanced verification
70016	200	Please specify your instrument settings for at least one instType.
70060	200	The {account} doesn't exist or the position side is incorrect. "To" and "from" accounts must be under the same main account.
70061	200	To move position, please enter a position that's opposite to your current side and is smaller than or equal to your current size.
70062	200	{account} has reached the maximum number of position transfers allowed per day.
70064	200	Position does not exist.
70065	200	Couldn't move position. Execution price cannot be determined.
70066	200	Moving positions isn't supported in spot mode. Switch to any other account mode and try again.
70067	200	Moving positions isn't supported in margin trading.
70100	200	Duplicate instruments in legs array.
70101	200	Duplicate clRfqId
70102	200	No counterparties specified
70103	200	Invalid counterparty
70105	200	The total value of non all-SPOT RFQs should be greater than the min notional value {nonSpotMinNotional}
70106	200	The trading amount does not meet the min tradable amount requirement
70107	200	The number of counterparties cannot exceed maximum value.
70108	200	The total value of all-spot RFQs should be greater than the min notional value {spotMinNotional}
70109	200	Counterparties for selected instruments are currently unavailable.
70200	200	The RFQ with {rfqState} status cannot be canceled
70203	200	Cancellation failed as rfq count exceeds the limit {rfqLimit}.
70207	200	Cancellation failed as you do not have any active RFQs.
70208	200	Cancellation failed as service is unavailable now, please try again later.
70301	200	Duplicate clQuotId.
70303	200	The RFQ with {rfqState} status cannot be quoted.
70304	200	Price should be an integer multiple of the tick size.
70305	200	Bid price cannot be higher than offer price
70306	200	The legs of quote do not match the legs of {rfqId}
70307	200	Size should be in integral multiples of the lot size.
70308	200	Quote to your own RFQ is not allowed.
70309	200	Quote to the same RFQ with the same side is not allowed.

Error Code	HTTP Status Code	Error Message
70310	200	Quoted price of instId {instId} cannot exceed your preset price limit.
70400	200	The Quote with {quoteState} status cannot be canceled
70408	200	Cancellation failed as quote count exceeds the limit {quoteLimit}.
70409	200	Cancellation failed as you do not have any active Quotes.
70501	200	RFQ {rfqId} is not quoted by {quotId}
70502	200	The legs do not match the legs of {rfqId}
70503	200	Leg sizes specified are under the minimum block size required by Jupiter.
70504	200	Execution failed as the RFQ status is {rfqState}.
70505	200	Execution failed as the Quote status is {quoteState}.
70506	200	Leg sizes specified do not have the same ratios as the whole RFQ.
70507	200	Partial execution was attempted but allowPartialExecution of the RFQ is not enabled.
70508	200	No instrument settings available.
70509	200	Execution failed: counterparty error
70510	200	For error details, refer to the acctAlloc field.
70511	200	Execution is being processed
70514	200	For each symbol, the total size of RFQ legs in all accounts should be equal to its combined amount in the group RFQ.
70515	200	For each sub-account, the ratio of a leg's size to the main account RFQ must be the same across all symbols.
70516	200	You can only select up to {param0} sub-accounts for group RFQ.
70517	200	{param0} doesn't exist or you don't have permission to create group RFQ for it.
70518	200	Make sure you didn't select the same account more than once for group RFQ.
75001	200	Trade ID does not exist
75002	200	{sprdId} : unable to place new orders or modify existing orders at the moment
75003	200	Invalid price
56000	200	Block trade does not exist.
56001	200	The number of multi-legs cannot exceed {legLimit}.
56002	200	The number of multi-legs does not match with the verified one.
56003	200	Duplicate clBlockTdId.
56004	200	Trade with yourself is not allowed.
56005	200	clBlockTdId should be the same as the verified one.
56006	200	The role should be different from the verified one.
56007	200	Leg no.{legNo} does not match with the verified one.

Error Code	HTTP Status Code	Error Message
56008	200	Duplicate instruments in legs array.
<b>COPY TRADING</b>		
Error Code from 59200 to 59300		
Error Code	HTTP Status Code	Error Message
59128	200	As a lead trader, you can't lead trades in {instrument} with leverage higher than {num}x
59129	200	The first crypto you use to repay must be {param0}.
59130	200	If an asset's balance is < 1 USD, it can only repay borrowings of the same crypto.
59206	200	The lead trader doesn't have any more vacancies for copy traders
59216	200	The position doesn't exist. Please try again
59218	200	Closing all positions at market price...
59256	200	To switch to One-way mode, lower the number of traders you copy to 1
59247	200	High leverage causes current position to exceed the maximum position size limit under this leverage. Adjust the leverage.
59260	200	You are not a spot lead trader yet. Complete the application on our website or app first.
59262	200	You aren't a contract lead trader yet. Complete the application first.
59641	200	Can't switch account mode as you have fixed loan borrowings.
59642	200	Lead and copy traders can only use spot or futures modes
59643	200	Couldn't switch account modes as you're currently copying spot trades
59245	200	As a lead trader, number of {param0} contract per order must be no greater than {param1}
59263	200	Only traders on the allowlist can use copy trading. ND brokers can reach out to BD for help.
59264	200	Spot copy trading isn't supported
59267	200	Cancellation failed as you aren't copying this trader
59268	200	You can't copy trades with instId that hasn't been selected by the lead trader
59269	200	This contract lead trader doesn't exist
59270	200	Maximum total amount (copyTotalAmt) can't be lower than amount per order (copyAmt) when using fixed amount
59273	200	You aren't a contract copy trader yet. Start by coping a contract trader.
59274	200	Copying your own trade isn't allowed
59275	200	You can't copy trade as you're applying to become a lead trader
59276	200	You can't copy this lead trader as they've applied to stop leading trades
59277	200	You can't copy this lead trader as they don't have any copy trader vacancies
59278	200	Your request to stop copy trading is being processed. Try again later.

Error Code	HTTP Status Code	Error Message
59279	200	You've already copied this trader
59280	200	You can't modify copy trade settings as you aren't copying this trader
59282	200	Only ND sub-accounts under ND brokers whose main accounts are on the allowlist support this endpoint. Reach out to BD for help.
59283	200	Your account isn't currently using futures mode
59284	200	You've reached the monthly limit of {param0} ratio edits
59286	200	You can't become a futures lead trader when using spot mode
59287	200	Profit sharing ratio should be between {param0} and {param1}
59288	200	You're leading trades but your account is in portfolio margin mode. Switch to futures mode or multiple-currency margin mode and try again.
59130	200	The highest take profit level is {num}%. Enter a smaller number and try again.
59258	200	Action not supported for lead traders
59259	200	Enter a multiplier value that's within the valid range
59285	200	You haven't led or copied any trades yet
59292	200	This lead trader only supports smart sync mode.

#### TRADING BOT

Error Code from 55100 to 55999

Error Code	HTTP Status Code	Error Message
55100	200	Take profit % should be within the range of {parameter1}-{parameter2}
55101	200	Stop loss % should be within the range of {parameter1}-{parameter2}
55102	200	Take profit % should be greater than the current bot's PnL%
55103	200	Stop loss % should be less than the current bot's PnL%
55104	200	Only futures grid supports take profit or stop loss based on profit percentage
55105	200	Increasing positions is not allowed under current status
55106	200	Increased amount should be within the range of {parameter1} - {parameter2}
55111	200	This signal name is in use, please try a new name
55112	200	This signal does not exist
55113	200	Create signal strategies with leverage greater than the maximum leverage of the instruments
55116	200	You can only place one chase order for each trading pair.

## GENERAL CLASS

Error Code	Error Message
60004	Invalid timestamp
60005	Invalid apiKey
60006	Timestamp request expired
60007	Invalid sign
60008	The current WebSocket endpoint does not support subscribing to {0} channels. Please check the WebSocket URL
60009	Login failure
60011	Please log in
60012	Invalid request
60013	Invalid args
60014	Requests too frequent
60018	Wrong URL or {0} doesn't exist. Please use the correct URL, channel and parameters referring to API document.
60019	Invalid op: {op}
60023	Bulk login requests too frequent
60024	Wrong passphrase
60026	Batch login by APIKey and token simultaneously is not supported.
60027	Parameter {0} can not be empty.
60028	The current operation is not supported by this URL. Please use the correct WebSocket URL for the operation.
60029	Only users who are VIP5 and above in trading fee tier are allowed to subscribe to this channel.
60030	Only users who are VIP4 and above in trading fee tier are allowed to subscribe to books50-l2-tbt channel.
60031	The WebSocket endpoint does not allow multiple or repeated logins.
60032	API key doesn't exist.
60033	Parameter {param0} error.
63999	Login failed due to internal error. Please try again later.
64000	Subscription parameter uly is unavailable anymore, please replace uly with instFamily. More details can refer to: <a href="https://www.okx.com/help-center/changes-to-v5-api-websocket-subscription-parameter-and-url">https://www.okx.com/help-center/changes-to-v5-api-websocket-subscription-parameter-and-url</a> .
64001	This channel has been migrated to the '/business' URL. Please subscribe using the new URL. More details can refer to: <a href="https://www.okx.com/help-center/changes-to-v5-api-websocket-subscription-parameter-and-url">https://www.okx.com/help-center/changes-to-v5-api-websocket-subscription-parameter-and-url</a> .
64002	This channel is not supported by "/business" URL. Please use "/private" URL(for private channels), or "/public" URL(for public channels). More details can refer to: <a href="https://www.okx.com/help-center/changes-to-v5-api-websocket-subscription-parameter-and-url">https://www.okx.com/help-center/changes-to-v5-api-websocket-subscription-parameter-and-url</a> .
64003	Your trading fee tier doesn't meet the requirement to access this channel
64004	Subscribe to both {channelName} and books-l2-tbt for {instId} is not allowed. Unsubscribe books-l2-tbt first.

Error Code	Error Message
64007	Operation {0} failed due to WebSocket internal error. Please try again later.
64008	The connection will soon be closed for a service upgrade. Please reconnect.

#### CLOSE FRAME

Status Code	Reason Text
1009	Request message exceeds the maximum frame length
4001	Login Failed
4002	Invalid Request
4003	APIKey subscription amount exceeds the limit 100
4004	No data received in 30s
4005	Buffer is full, cannot write data
4006	Abnormal disconnection
4007	API key has been updated or deleted. Please reconnect.
4008	The number of subscribed channels exceeds the maximum limit.
4009	The number of subscription channels for this connection exceeds the limit

Disclaimer: The availability of products and services listed on this page will depend on your region. Please see your applicable Terms of Service for more detail.