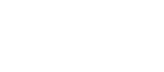
[NAV ](https://www.okx.com/docs-v5/en/)

[](https://www.okx.com/)

* [API](https://www.okx.com/docs-v5/en)
* [Broker](https://www.okx.com/docs-v5/broker_en)
* [Best Practice](https://www.okx.com/docs-v5/trick_en)
* [Change Log](https://www.okx.com/docs-v5/log_en)

API Broker Best Practice Change Log

中文

[HTTP](https://www.okx.com/docs-v5/en/) [Python](https://www.okx.com/docs-v5/en/)

* [Get deposit history](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-deposit-history)
* [Get Deposit Order Detail](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-deposit-order-detail)
* [Get Deposit Order History](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-deposit-order-history)
* [Get deposit address](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-deposit-address)
* [Deposit info channel](https://www.okx.com/docs-v5/en/#funding-account-websocket-deposit-info-channel)
* [Get deposit withdraw status](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-deposit-withdraw-status)
* [Get deposit payment methods](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-deposit-payment-methods)
* [Get currencies](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-currencies)
* [Funding](https://www.okx.com/docs-v5/en/#error-code-rest-api-funding)
* [Get the invitee's detail](https://www.okx.com/docs-v5/en/#affiliate-rest-api-get-the-invitee-39-s-detail)
* [Get withdrawal payment methods](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-withdrawal-payment-methods)
* [Asset bills details](https://www.okx.com/docs-v5/en/#funding-account-rest-api-asset-bills-details)
* [Get account and position risk](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-account-and-position-risk)
* [Block Trading Workflow](https://www.okx.com/docs-v5/en/#block-trading-block-trading-workflow)
* [Balance and position channel](https://www.okx.com/docs-v5/en/#trading-account-websocket-balance-and-position-channel)
* [Get withdrawal history](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-withdrawal-history)
* [Overview](https://www.okx.com/docs-v5/en/#overview)
  + [API Resources and Support](https://www.okx.com/docs-v5/en/#overview-api-resources-and-support)
    - [Tutorials](https://www.okx.com/docs-v5/en/#overview-api-resources-and-support-tutorials)
    - [Python libraries](https://www.okx.com/docs-v5/en/#overview-api-resources-and-support-python-libraries)
    - [Customer service](https://www.okx.com/docs-v5/en/#overview-api-resources-and-support-customer-service)
  + [V5 API Key Creation](https://www.okx.com/docs-v5/en/#overview-v5-api-key-creation)
  + [REST Authentication](https://www.okx.com/docs-v5/en/#overview-rest-authentication)
    - [Making Requests](https://www.okx.com/docs-v5/en/#overview-rest-authentication-making-requests)
    - [Signature](https://www.okx.com/docs-v5/en/#overview-rest-authentication-signature)
  + [WebSocket](https://www.okx.com/docs-v5/en/#overview-websocket)
    - [Overview](https://www.okx.com/docs-v5/en/#overview-websocket-overview)
    - [Connect](https://www.okx.com/docs-v5/en/#overview-websocket-connect)
    - [Connection count limit](https://www.okx.com/docs-v5/en/#overview-websocket-connection-count-limit)
    - [Login](https://www.okx.com/docs-v5/en/#overview-websocket-login)
    - [Subscribe](https://www.okx.com/docs-v5/en/#overview-websocket-subscribe)
    - [Unsubscribe](https://www.okx.com/docs-v5/en/#overview-websocket-unsubscribe)
  + [Account mode](https://www.okx.com/docs-v5/en/#overview-account-mode)
  + [Production Trading Services](https://www.okx.com/docs-v5/en/#overview-production-trading-services)
  + [Demo Trading Services](https://www.okx.com/docs-v5/en/#overview-demo-trading-services)
    - [Demo Trading Explorer](https://www.okx.com/docs-v5/en/#overview-demo-trading-services-demo-trading-explorer)
  + [General Info](https://www.okx.com/docs-v5/en/#overview-general-info)
  + [Transaction Timeouts](https://www.okx.com/docs-v5/en/#overview-transaction-timeouts)
    - [REST API](https://www.okx.com/docs-v5/en/#overview-transaction-timeouts-rest-api)
    - [WebSocket](https://www.okx.com/docs-v5/en/#overview-transaction-timeouts-websocket)
  + [Rate Limits](https://www.okx.com/docs-v5/en/#overview-rate-limits)
    - [Trading-related APIs](https://www.okx.com/docs-v5/en/#overview-rate-limits-trading-related-apis)
    - [Sub-account rate limit](https://www.okx.com/docs-v5/en/#overview-rate-limits-sub-account-rate-limit)
    - [Fill ratio based sub-account rate limit](https://www.okx.com/docs-v5/en/#overview-rate-limits-fill-ratio-based-sub-account-rate-limit)
    - [Best practices](https://www.okx.com/docs-v5/en/#overview-rate-limits-best-practices)
  + [Market Maker Program](https://www.okx.com/docs-v5/en/#overview-market-maker-program)
  + [Broker Program](https://www.okx.com/docs-v5/en/#overview-broker-program)
* [Trading Account](https://www.okx.com/docs-v5/en/#trading-account)
  + [REST API](https://www.okx.com/docs-v5/en/#trading-account-rest-api)
    - [Get instruments](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-instruments)
    - [Get balance](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-balance)
    - [Get positions](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-positions)
    - [Get positions history](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-positions-history)
    - [Get account and position risk](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-account-and-position-risk)
    - [Get bills details (last 7 days)](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-bills-details-last-7-days)
    - [Get bills details (last 3 months)](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-bills-details-last-3-months)
    - [Apply bills details (since 2021)](https://www.okx.com/docs-v5/en/#trading-account-rest-api-apply-bills-details-since-2021)
    - [Get bills details (since 2021)](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-bills-details-since-2021)
    - [Get account configuration](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-account-configuration)
    - [Set position mode](https://www.okx.com/docs-v5/en/#trading-account-rest-api-set-position-mode)
    - [Set leverage](https://www.okx.com/docs-v5/en/#trading-account-rest-api-set-leverage)
    - [Get maximum order quantity](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-maximum-order-quantity)
    - [Get maximum available balance/equity](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-maximum-available-balance-equity)
    - [Increase/decrease margin](https://www.okx.com/docs-v5/en/#trading-account-rest-api-increase-decrease-margin)
    - [Get leverage](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-leverage)
    - [Get leverage estimated info](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-leverage-estimated-info)
    - [Get the maximum loan of instrument](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-the-maximum-loan-of-instrument)
    - [Get fee rates](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-fee-rates)
    - [Get interest accrued data](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-interest-accrued-data)
    - [Get interest rate](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-interest-rate)
    - [Set greeks (PA/BS)](https://www.okx.com/docs-v5/en/#trading-account-rest-api-set-greeks-pa-bs)
    - [Isolated margin trading settings](https://www.okx.com/docs-v5/en/#trading-account-rest-api-isolated-margin-trading-settings)
    - [Get maximum withdrawals](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-maximum-withdrawals)
    - [Get account risk state](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-account-risk-state)
    - [Manual borrow and repay in Quick Margin Mode](https://www.okx.com/docs-v5/en/#trading-account-rest-api-manual-borrow-and-repay-in-quick-margin-mode)
    - [Get borrow and repay history in Quick Margin Mode](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-borrow-and-repay-history-in-quick-margin-mode)
    - [Get borrow interest and limit](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-borrow-interest-and-limit)
    - [Get fixed loan borrow limit](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-fixed-loan-borrow-limit)
    - [Get fixed loan borrow quote](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-fixed-loan-borrow-quote)
    - [Place fixed loan borrowing order](https://www.okx.com/docs-v5/en/#trading-account-rest-api-place-fixed-loan-borrowing-order)
    - [Amend fixed loan borrowing order](https://www.okx.com/docs-v5/en/#trading-account-rest-api-amend-fixed-loan-borrowing-order)
    - [Manual renew fixed loan borrowing order](https://www.okx.com/docs-v5/en/#trading-account-rest-api-manual-renew-fixed-loan-borrowing-order)
    - [Repay fixed loan borrowing order](https://www.okx.com/docs-v5/en/#trading-account-rest-api-repay-fixed-loan-borrowing-order)
    - [Convert fixed loan to market loan](https://www.okx.com/docs-v5/en/#trading-account-rest-api-convert-fixed-loan-to-market-loan)
    - [Reduce liabilities for fixed loan](https://www.okx.com/docs-v5/en/#trading-account-rest-api-reduce-liabilities-for-fixed-loan)
    - [Get fixed loan borrow order list](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-fixed-loan-borrow-order-list)
    - [Manual borrow / repay](https://www.okx.com/docs-v5/en/#trading-account-rest-api-manual-borrow-repay)
    - [Set auto repay](https://www.okx.com/docs-v5/en/#trading-account-rest-api-set-auto-repay)
    - [Get borrow/repay history](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-borrow-repay-history)
    - [Position builder (new)](https://www.okx.com/docs-v5/en/#trading-account-rest-api-position-builder-new)
    - [Set risk offset amount](https://www.okx.com/docs-v5/en/#trading-account-rest-api-set-risk-offset-amount)
    - [Get Greeks](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-greeks)
    - [Get PM position limitation](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-pm-position-limitation)
    - [Set risk offset type](https://www.okx.com/docs-v5/en/#trading-account-rest-api-set-risk-offset-type)
    - [Activate option](https://www.okx.com/docs-v5/en/#trading-account-rest-api-activate-option)
    - [Set auto loan](https://www.okx.com/docs-v5/en/#trading-account-rest-api-set-auto-loan)
    - [Preset account mode switch](https://www.okx.com/docs-v5/en/#trading-account-rest-api-preset-account-mode-switch)
    - [Precheck account mode switch](https://www.okx.com/docs-v5/en/#trading-account-rest-api-precheck-account-mode-switch)
    - [Set account mode](https://www.okx.com/docs-v5/en/#trading-account-rest-api-set-account-mode)
    - [Reset MMP Status](https://www.okx.com/docs-v5/en/#trading-account-rest-api-reset-mmp-status)
    - [Set MMP](https://www.okx.com/docs-v5/en/#trading-account-rest-api-set-mmp)
    - [GET MMP Config](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-mmp-config)
  + [WebSocket](https://www.okx.com/docs-v5/en/#trading-account-websocket)
    - [Account channel](https://www.okx.com/docs-v5/en/#trading-account-websocket-account-channel)
    - [Positions channel](https://www.okx.com/docs-v5/en/#trading-account-websocket-positions-channel)
    - [Balance and position channel](https://www.okx.com/docs-v5/en/#trading-account-websocket-balance-and-position-channel)
    - [Position risk warning](https://www.okx.com/docs-v5/en/#trading-account-websocket-position-risk-warning)
    - [Account greeks channel](https://www.okx.com/docs-v5/en/#trading-account-websocket-account-greeks-channel)
* [Order Book Trading](https://www.okx.com/docs-v5/en/#order-book-trading)
  + [Trade](https://www.okx.com/docs-v5/en/#order-book-trading-trade)
    - [POST / Place order](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-place-order)
    - [POST / Place multiple orders](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-place-multiple-orders)
    - [POST / Cancel order](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-cancel-order)
    - [POST / Cancel multiple orders](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-cancel-multiple-orders)
    - [POST / Amend order](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-amend-order)
    - [POST / Amend multiple orders](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-amend-multiple-orders)
    - [POST / Close positions](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-close-positions)
    - [GET / Order details](https://www.okx.com/docs-v5/en/#order-book-trading-trade-get-order-details)
    - [GET / Order List](https://www.okx.com/docs-v5/en/#order-book-trading-trade-get-order-list)
    - [GET / Order history (last 7 days)](https://www.okx.com/docs-v5/en/#order-book-trading-trade-get-order-history-last-7-days)
    - [GET / Order history (last 3 months)](https://www.okx.com/docs-v5/en/#order-book-trading-trade-get-order-history-last-3-months)
    - [GET / Transaction details (last 3 days)](https://www.okx.com/docs-v5/en/#order-book-trading-trade-get-transaction-details-last-3-days)
    - [GET / Transaction details (last 3 months)](https://www.okx.com/docs-v5/en/#order-book-trading-trade-get-transaction-details-last-3-months)
    - [GET / Easy convert currency list](https://www.okx.com/docs-v5/en/#order-book-trading-trade-get-easy-convert-currency-list)
    - [POST / Place easy convert](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-place-easy-convert)
    - [GET / Easy convert history](https://www.okx.com/docs-v5/en/#order-book-trading-trade-get-easy-convert-history)
    - [GET / One-click repay currency list](https://www.okx.com/docs-v5/en/#order-book-trading-trade-get-one-click-repay-currency-list)
    - [POST / Trade one-click repay](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-trade-one-click-repay)
    - [GET / One-click repay history](https://www.okx.com/docs-v5/en/#order-book-trading-trade-get-one-click-repay-history)
    - [POST / Mass cancel order](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-mass-cancel-order)
    - [POST / Cancel All After](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-cancel-all-after)
    - [GET / Account rate limit](https://www.okx.com/docs-v5/en/#order-book-trading-trade-get-account-rate-limit)
    - [POST / Order precheck](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-order-precheck)
    - [WS / Order channel](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-order-channel)
    - [WS / Fills channel](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-fills-channel)
    - [WS / Place order](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-place-order)
    - [WS / Place multiple orders](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-place-multiple-orders)
    - [WS / Cancel order](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-cancel-order)
    - [WS / Cancel multiple orders](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-cancel-multiple-orders)
    - [WS / Amend order](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-amend-order)
    - [WS / Amend multiple orders](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-amend-multiple-orders)
    - [WS / Mass cancel order](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-mass-cancel-order)
  + [Algo Trading](https://www.okx.com/docs-v5/en/#order-book-trading-algo-trading)
    - [POST / Place algo order](https://www.okx.com/docs-v5/en/#order-book-trading-algo-trading-post-place-algo-order)
    - [POST / Cancel algo order](https://www.okx.com/docs-v5/en/#order-book-trading-algo-trading-post-cancel-algo-order)
    - [POST / Amend algo order](https://www.okx.com/docs-v5/en/#order-book-trading-algo-trading-post-amend-algo-order)
    - [POST / Cancel advance algo order](https://www.okx.com/docs-v5/en/#order-book-trading-algo-trading-post-cancel-advance-algo-order)
    - [GET / Algo order details](https://www.okx.com/docs-v5/en/#order-book-trading-algo-trading-get-algo-order-details)
    - [GET / Algo order list](https://www.okx.com/docs-v5/en/#order-book-trading-algo-trading-get-algo-order-list)
    - [GET / Algo order history](https://www.okx.com/docs-v5/en/#order-book-trading-algo-trading-get-algo-order-history)
    - [WS / Algo orders channel](https://www.okx.com/docs-v5/en/#order-book-trading-algo-trading-ws-algo-orders-channel)
    - [WS / Advance algo orders channel](https://www.okx.com/docs-v5/en/#order-book-trading-algo-trading-ws-advance-algo-orders-channel)
  + [Grid Trading](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading)
    - [POST / Place grid algo order](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-post-place-grid-algo-order)
    - [POST / Amend grid algo order](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-post-amend-grid-algo-order)
    - [POST / Stop grid algo order](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-post-stop-grid-algo-order)
    - [POST / Close position for contract grid](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-post-close-position-for-contract-grid)
    - [POST / Cancel close position order for contract grid](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-post-cancel-close-position-order-for-contract-grid)
    - [POST / Instant trigger grid algo order](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-post-instant-trigger-grid-algo-order)
    - [GET / Grid algo order list](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-get-grid-algo-order-list)
    - [GET / Grid algo order history](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-get-grid-algo-order-history)
    - [GET / Grid algo order details](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-get-grid-algo-order-details)
    - [GET / Grid algo sub orders](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-get-grid-algo-sub-orders)
    - [GET / Grid algo order positions](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-get-grid-algo-order-positions)
    - [POST / Spot grid withdraw income](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-post-spot-grid-withdraw-income)
    - [POST / Compute margin balance](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-post-compute-margin-balance)
    - [POST / Adjust margin balance](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-post-adjust-margin-balance)
    - [POST / Add investment](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-post-add-investment)
    - [GET / Grid AI parameter (public)](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-get-grid-ai-parameter-public)
    - [POST / Compute min investment (public)](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-post-compute-min-investment-public)
    - [GET / RSI back testing (public)](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-get-rsi-back-testing-public)
    - [GET / Max grid quantity (public)](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-get-max-grid-quantity-public)
    - [WS / Spot grid algo orders channel](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-ws-spot-grid-algo-orders-channel)
    - [WS / Contract grid algo orders channel](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-ws-contract-grid-algo-orders-channel)
    - [WS / Grid positions channel](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-ws-grid-positions-channel)
    - [WS / Grid sub orders channel](https://www.okx.com/docs-v5/en/#order-book-trading-grid-trading-ws-grid-sub-orders-channel)
  + [Signal bot trading](https://www.okx.com/docs-v5/en/#order-book-trading-signal-bot-trading)
    - [POST / Create signal](https://www.okx.com/docs-v5/en/#order-book-trading-signal-bot-trading-post-create-signal)
    - [GET / Signals](https://www.okx.com/docs-v5/en/#order-book-trading-signal-bot-trading-get-signals)
    - [POST / Create signal bot](https://www.okx.com/docs-v5/en/#order-book-trading-signal-bot-trading-post-create-signal-bot)
    - [POST / Cancel signal bots](https://www.okx.com/docs-v5/en/#order-book-trading-signal-bot-trading-post-cancel-signal-bots)
    - [POST / Adjust margin balance](https://www.okx.com/docs-v5/en/#order-book-trading-signal-bot-trading-post-adjust-margin-balance)
    - [POST / Amend TPSL](https://www.okx.com/docs-v5/en/#order-book-trading-signal-bot-trading-post-amend-tpsl)
    - [POST / Set instruments](https://www.okx.com/docs-v5/en/#order-book-trading-signal-bot-trading-post-set-instruments)
    - [GET / Signal bot order details](https://www.okx.com/docs-v5/en/#order-book-trading-signal-bot-trading-get-signal-bot-order-details)
    - [GET / Active signal bot](https://www.okx.com/docs-v5/en/#order-book-trading-signal-bot-trading-get-active-signal-bot)
    - [GET / Signal bot history](https://www.okx.com/docs-v5/en/#order-book-trading-signal-bot-trading-get-signal-bot-history)
    - [GET / Signal bot order positions](https://www.okx.com/docs-v5/en/#order-book-trading-signal-bot-trading-get-signal-bot-order-positions)
    - [GET / Position history](https://www.okx.com/docs-v5/en/#order-book-trading-signal-bot-trading-get-position-history)
    - [POST / Close position](https://www.okx.com/docs-v5/en/#order-book-trading-signal-bot-trading-post-close-position)
    - [POST / Place sub order](https://www.okx.com/docs-v5/en/#order-book-trading-signal-bot-trading-post-place-sub-order)
    - [POST / Cancel sub order](https://www.okx.com/docs-v5/en/#order-book-trading-signal-bot-trading-post-cancel-sub-order)
    - [GET / Signal bot sub orders](https://www.okx.com/docs-v5/en/#order-book-trading-signal-bot-trading-get-signal-bot-sub-orders)
    - [GET / Signal bot event history](https://www.okx.com/docs-v5/en/#order-book-trading-signal-bot-trading-get-signal-bot-event-history)
  + [Recurring Buy](https://www.okx.com/docs-v5/en/#order-book-trading-recurring-buy)
    - [POST / Place recurring buy order](https://www.okx.com/docs-v5/en/#order-book-trading-recurring-buy-post-place-recurring-buy-order)
    - [POST / Amend recurring buy order](https://www.okx.com/docs-v5/en/#order-book-trading-recurring-buy-post-amend-recurring-buy-order)
    - [POST / Stop recurring buy order](https://www.okx.com/docs-v5/en/#order-book-trading-recurring-buy-post-stop-recurring-buy-order)
    - [GET / Recurring buy order list](https://www.okx.com/docs-v5/en/#order-book-trading-recurring-buy-get-recurring-buy-order-list)
    - [GET / Recurring buy order history](https://www.okx.com/docs-v5/en/#order-book-trading-recurring-buy-get-recurring-buy-order-history)
    - [GET / Recurring buy order details](https://www.okx.com/docs-v5/en/#order-book-trading-recurring-buy-get-recurring-buy-order-details)
    - [GET / Recurring buy sub orders](https://www.okx.com/docs-v5/en/#order-book-trading-recurring-buy-get-recurring-buy-sub-orders)
    - [WS / Recurring buy orders channel](https://www.okx.com/docs-v5/en/#order-book-trading-recurring-buy-ws-recurring-buy-orders-channel)
  + [Copy Trading](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading)
    - [GET / Existing lead or copy positions](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-existing-lead-or-copy-positions)
    - [GET / Lead or copy position history](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-lead-or-copy-position-history)
    - [POST / Place lead or copy stop order](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-post-place-lead-or-copy-stop-order)
    - [POST / Close lead or copy position](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-post-close-lead-or-copy-position)
    - [GET / Leading instruments](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-leading-instruments)
    - [POST / Amend leading instruments](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-post-amend-leading-instruments)
    - [GET / Profit sharing details](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-profit-sharing-details)
    - [GET / Total profit sharing](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-total-profit-sharing)
    - [GET / Unrealized profit sharing details](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-unrealized-profit-sharing-details)
    - [GET / Total unrealized profit sharing](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-total-unrealized-profit-sharing)
    - [POST / Apply for lead trading](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-post-apply-for-lead-trading)
    - [POST / Stop lead trading](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-post-stop-lead-trading)
    - [POST / Amend profit sharing ratio](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-post-amend-profit-sharing-ratio)
    - [GET / Account configuration](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-account-configuration)
    - [POST / First copy settings](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-post-first-copy-settings)
    - [POST / Amend copy settings](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-post-amend-copy-settings)
    - [POST / Stop copying](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-post-stop-copying)
    - [GET / Copy settings](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-copy-settings)
    - [GET / Multiple leverages](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-multiple-leverages)
    - [POST / Set Multiple leverages](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-post-set-multiple-leverages)
    - [GET / My lead traders](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-my-lead-traders)
    - [GET / My history lead traders](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-my-history-lead-traders)
    - [GET / Copy trading configuration](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-copy-trading-configuration)
    - [GET / Lead trader ranks](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-lead-trader-ranks)
    - [GET / Lead trader weekly pnl](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-lead-trader-weekly-pnl)
    - [GET / Lead trader daily pnl](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-lead-trader-daily-pnl)
    - [GET / Lead trader stats](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-lead-trader-stats)
    - [GET / Lead trader currency preferences](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-lead-trader-currency-preferences)
    - [GET / Lead trader current lead positions](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-lead-trader-current-lead-positions)
    - [GET / Lead trader lead position history](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-lead-trader-lead-position-history)
    - [GET / Copy traders](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-copy-traders)
    - [GET / Lead trader ranks (private)](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-lead-trader-ranks-private)
    - [GET / Lead trader weekly pnl (private)](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-lead-trader-weekly-pnl-private)
    - [GET / Lead trader daily pnl (private)](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-lead-trader-daily-pnl-private)
    - [GET / Lead trader stats (private)](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-lead-trader-stats-private)
    - [GET / Lead trader currency preferences (private)](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-lead-trader-currency-preferences-private)
    - [GET / Lead trader current lead positions (private)](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-lead-trader-current-lead-positions-private)
    - [GET / Lead trader lead position history (private)](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-lead-trader-lead-position-history-private)
    - [GET / Copy traders (private)](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-copy-traders-private)
    - [WS / Copy trading notification channel](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-ws-copy-trading-notification-channel)
    - [WS / Lead trading notification channel](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-ws-lead-trading-notification-channel)
  + [Market Data](https://www.okx.com/docs-v5/en/#order-book-trading-market-data)
    - [GET / Tickers](https://www.okx.com/docs-v5/en/#order-book-trading-market-data-get-tickers)
    - [GET / Ticker](https://www.okx.com/docs-v5/en/#order-book-trading-market-data-get-ticker)
    - [GET / Order book](https://www.okx.com/docs-v5/en/#order-book-trading-market-data-get-order-book)
    - [GET / Full order book](https://www.okx.com/docs-v5/en/#order-book-trading-market-data-get-full-order-book)
    - [GET / Candlesticks](https://www.okx.com/docs-v5/en/#order-book-trading-market-data-get-candlesticks)
    - [GET / Candlesticks history](https://www.okx.com/docs-v5/en/#order-book-trading-market-data-get-candlesticks-history)
    - [GET / Trades](https://www.okx.com/docs-v5/en/#order-book-trading-market-data-get-trades)
    - [GET / Trades history](https://www.okx.com/docs-v5/en/#order-book-trading-market-data-get-trades-history)
    - [GET / Option trades by instrument family](https://www.okx.com/docs-v5/en/#order-book-trading-market-data-get-option-trades-by-instrument-family)
    - [GET / Option trades](https://www.okx.com/docs-v5/en/#order-book-trading-market-data-get-option-trades)
    - [GET / 24H total volume](https://www.okx.com/docs-v5/en/#order-book-trading-market-data-get-24h-total-volume)
    - [GET / Call auction details](https://www.okx.com/docs-v5/en/#order-book-trading-market-data-get-call-auction-details)
    - [WS / Tickers channel](https://www.okx.com/docs-v5/en/#order-book-trading-market-data-ws-tickers-channel)
    - [WS / Candlesticks channel](https://www.okx.com/docs-v5/en/#order-book-trading-market-data-ws-candlesticks-channel)
    - [WS / Trades channel](https://www.okx.com/docs-v5/en/#order-book-trading-market-data-ws-trades-channel)
    - [WS / All trades channel](https://www.okx.com/docs-v5/en/#order-book-trading-market-data-ws-all-trades-channel)
    - [WS / Order book channel](https://www.okx.com/docs-v5/en/#order-book-trading-market-data-ws-order-book-channel)
    - [WS / Option trades channel](https://www.okx.com/docs-v5/en/#order-book-trading-market-data-ws-option-trades-channel)
    - [WS / Call auction details channel](https://www.okx.com/docs-v5/en/#order-book-trading-market-data-ws-call-auction-details-channel)
* [Block Trading](https://www.okx.com/docs-v5/en/#block-trading)
  + [Block Trading Workflow](https://www.okx.com/docs-v5/en/#block-trading-block-trading-workflow)
  + [REST API](https://www.okx.com/docs-v5/en/#block-trading-rest-api)
    - [Get Counterparties](https://www.okx.com/docs-v5/en/#block-trading-rest-api-get-counterparties)
    - [Create RFQ](https://www.okx.com/docs-v5/en/#block-trading-rest-api-create-rfq)
    - [Cancel RFQ](https://www.okx.com/docs-v5/en/#block-trading-rest-api-cancel-rfq)
    - [Cancel multiple RFQs](https://www.okx.com/docs-v5/en/#block-trading-rest-api-cancel-multiple-rfqs)
    - [Cancel all RFQs](https://www.okx.com/docs-v5/en/#block-trading-rest-api-cancel-all-rfqs)
    - [Execute Quote](https://www.okx.com/docs-v5/en/#block-trading-rest-api-execute-quote)
    - [Get Quote products](https://www.okx.com/docs-v5/en/#block-trading-rest-api-get-quote-products)
    - [Set Quote products](https://www.okx.com/docs-v5/en/#block-trading-rest-api-set-quote-products)
    - [Reset MMP status](https://www.okx.com/docs-v5/en/#block-trading-rest-api-reset-mmp-status)
    - [Set MMP](https://www.okx.com/docs-v5/en/#block-trading-rest-api-set-mmp)
    - [Get MMP Config](https://www.okx.com/docs-v5/en/#block-trading-rest-api-get-mmp-config)
    - [Create Quote](https://www.okx.com/docs-v5/en/#block-trading-rest-api-create-quote)
    - [Cancel Quote](https://www.okx.com/docs-v5/en/#block-trading-rest-api-cancel-quote)
    - [Cancel multiple Quotes](https://www.okx.com/docs-v5/en/#block-trading-rest-api-cancel-multiple-quotes)
    - [Cancel all Quotes](https://www.okx.com/docs-v5/en/#block-trading-rest-api-cancel-all-quotes)
    - [Cancel All After](https://www.okx.com/docs-v5/en/#block-trading-rest-api-cancel-all-after)
    - [Get rfqs](https://www.okx.com/docs-v5/en/#block-trading-rest-api-get-rfqs)
    - [Get quotes](https://www.okx.com/docs-v5/en/#block-trading-rest-api-get-quotes)
    - [Get trades](https://www.okx.com/docs-v5/en/#block-trading-rest-api-get-trades)
    - [Get block tickers](https://www.okx.com/docs-v5/en/#block-trading-rest-api-get-block-tickers)
    - [Get block ticker](https://www.okx.com/docs-v5/en/#block-trading-rest-api-get-block-ticker)
    - [Get public multi-leg transactions of block trades](https://www.okx.com/docs-v5/en/#block-trading-rest-api-get-public-multi-leg-transactions-of-block-trades)
    - [Get public single-leg transactions of block trades](https://www.okx.com/docs-v5/en/#block-trading-rest-api-get-public-single-leg-transactions-of-block-trades)
  + [WebSocket Private Channel](https://www.okx.com/docs-v5/en/#block-trading-websocket-private-channel)
    - [Rfqs channel](https://www.okx.com/docs-v5/en/#block-trading-websocket-private-channel-rfqs-channel)
    - [Quotes channel](https://www.okx.com/docs-v5/en/#block-trading-websocket-private-channel-quotes-channel)
    - [Structure block trades channel](https://www.okx.com/docs-v5/en/#block-trading-websocket-private-channel-structure-block-trades-channel)
  + [WebSocket Public Channel](https://www.okx.com/docs-v5/en/#block-trading-websocket-public-channel)
    - [Public structure block trades channel](https://www.okx.com/docs-v5/en/#block-trading-websocket-public-channel-public-structure-block-trades-channel)
    - [Public block trades channel](https://www.okx.com/docs-v5/en/#block-trading-websocket-public-channel-public-block-trades-channel)
    - [Block tickers channel](https://www.okx.com/docs-v5/en/#block-trading-websocket-public-channel-block-tickers-channel)
* [Spread Trading](https://www.okx.com/docs-v5/en/#spread-trading)
  + [Introduction](https://www.okx.com/docs-v5/en/#spread-trading-introduction)
    - [Basic Concepts](https://www.okx.com/docs-v5/en/#spread-trading-introduction-basic-concepts)
    - [High Level Workflow](https://www.okx.com/docs-v5/en/#spread-trading-introduction-high-level-workflow)
  + [Comprehensive API Workflow](https://www.okx.com/docs-v5/en/#spread-trading-comprehensive-api-workflow)
    - [Obtaining Available Spreads](https://www.okx.com/docs-v5/en/#spread-trading-comprehensive-api-workflow-obtaining-available-spreads)
    - [Retrieving Your Orders](https://www.okx.com/docs-v5/en/#spread-trading-comprehensive-api-workflow-retrieving-your-orders)
    - [Retrieving Your Trades](https://www.okx.com/docs-v5/en/#spread-trading-comprehensive-api-workflow-retrieving-your-trades)
    - [Submitting an Order](https://www.okx.com/docs-v5/en/#spread-trading-comprehensive-api-workflow-submitting-an-order)
    - [Spread States](https://www.okx.com/docs-v5/en/#spread-trading-comprehensive-api-workflow-spread-states)
    - [Trade Lifecycle](https://www.okx.com/docs-v5/en/#spread-trading-comprehensive-api-workflow-trade-lifecycle)
    - [Order State](https://www.okx.com/docs-v5/en/#spread-trading-comprehensive-api-workflow-order-state)
    - [Trade State](https://www.okx.com/docs-v5/en/#spread-trading-comprehensive-api-workflow-trade-state)
    - [All Trades](https://www.okx.com/docs-v5/en/#spread-trading-comprehensive-api-workflow-all-trades)
  + [REST API](https://www.okx.com/docs-v5/en/#spread-trading-rest-api)
    - [Place order](https://www.okx.com/docs-v5/en/#spread-trading-rest-api-place-order)
    - [Cancel order](https://www.okx.com/docs-v5/en/#spread-trading-rest-api-cancel-order)
    - [Cancel All orders](https://www.okx.com/docs-v5/en/#spread-trading-rest-api-cancel-all-orders)
    - [Amend order](https://www.okx.com/docs-v5/en/#spread-trading-rest-api-amend-order)
    - [Get order details](https://www.okx.com/docs-v5/en/#spread-trading-rest-api-get-order-details)
    - [Get active orders](https://www.okx.com/docs-v5/en/#spread-trading-rest-api-get-active-orders)
    - [Get orders (last 21 days)](https://www.okx.com/docs-v5/en/#spread-trading-rest-api-get-orders-last-21-days)
    - [Get orders history (last 3 months)](https://www.okx.com/docs-v5/en/#spread-trading-rest-api-get-orders-history-last-3-months)
    - [Get trades (last 7 days)](https://www.okx.com/docs-v5/en/#spread-trading-rest-api-get-trades-last-7-days)
    - [Get Spreads (Public)](https://www.okx.com/docs-v5/en/#spread-trading-rest-api-get-spreads-public)
    - [Get order book (Public)](https://www.okx.com/docs-v5/en/#spread-trading-rest-api-get-order-book-public)
    - [Get ticker (Public)](https://www.okx.com/docs-v5/en/#spread-trading-rest-api-get-ticker-public)
    - [Get public trades (Public)](https://www.okx.com/docs-v5/en/#spread-trading-rest-api-get-public-trades-public)
    - [Get candlesticks](https://www.okx.com/docs-v5/en/#spread-trading-rest-api-get-candlesticks)
    - [Get candlesticks history](https://www.okx.com/docs-v5/en/#spread-trading-rest-api-get-candlesticks-history)
    - [Cancel All After](https://www.okx.com/docs-v5/en/#spread-trading-rest-api-cancel-all-after)
  + [Websocket Trade API](https://www.okx.com/docs-v5/en/#spread-trading-websocket-trade-api)
    - [WS / Place order](https://www.okx.com/docs-v5/en/#spread-trading-websocket-trade-api-ws-place-order)
    - [WS / Amend order](https://www.okx.com/docs-v5/en/#spread-trading-websocket-trade-api-ws-amend-order)
    - [WS / Cancel order](https://www.okx.com/docs-v5/en/#spread-trading-websocket-trade-api-ws-cancel-order)
    - [WS / Cancel all orders](https://www.okx.com/docs-v5/en/#spread-trading-websocket-trade-api-ws-cancel-all-orders)
  + [WebSocket Private Channel](https://www.okx.com/docs-v5/en/#spread-trading-websocket-private-channel)
    - [Order channel](https://www.okx.com/docs-v5/en/#spread-trading-websocket-private-channel-order-channel)
    - [Trades channel](https://www.okx.com/docs-v5/en/#spread-trading-websocket-private-channel-trades-channel)
  + [WebSocket Public Channel](https://www.okx.com/docs-v5/en/#spread-trading-websocket-public-channel)
    - [Order book channel](https://www.okx.com/docs-v5/en/#spread-trading-websocket-public-channel-order-book-channel)
    - [Public Trades channel](https://www.okx.com/docs-v5/en/#spread-trading-websocket-public-channel-public-trades-channel)
    - [Tickers channel](https://www.okx.com/docs-v5/en/#spread-trading-websocket-public-channel-tickers-channel)
    - [Candlesticks channel](https://www.okx.com/docs-v5/en/#spread-trading-websocket-public-channel-candlesticks-channel)
* [Public Data](https://www.okx.com/docs-v5/en/#public-data)
  + [REST API](https://www.okx.com/docs-v5/en/#public-data-rest-api)
    - [Get instruments](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-instruments)
    - [Get delivery/exercise history](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-delivery-exercise-history)
    - [Get open interest](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-open-interest)
    - [Get funding rate](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-funding-rate)
    - [Get funding rate history](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-funding-rate-history)
    - [Get limit price](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-limit-price)
    - [Get option market data](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-option-market-data)
    - [Get estimated delivery/exercise price](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-estimated-delivery-exercise-price)
    - [Get discount rate and interest-free quota](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-discount-rate-and-interest-free-quota)
    - [Get system time](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-system-time)
    - [Get mark price](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-mark-price)
    - [Get position tiers](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-position-tiers)
    - [Get interest rate and loan quota](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-interest-rate-and-loan-quota)
    - [Get underlying](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-underlying)
    - [Get insurance fund](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-insurance-fund)
    - [Unit convert](https://www.okx.com/docs-v5/en/#public-data-rest-api-unit-convert)
    - [Get option tick bands](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-option-tick-bands)
    - [Get premium history](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-premium-history)
    - [Get index tickers](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-index-tickers)
    - [Get index candlesticks](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-index-candlesticks)
    - [Get index candlesticks history](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-index-candlesticks-history)
    - [Get mark price candlesticks](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-mark-price-candlesticks)
    - [Get mark price candlesticks history](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-mark-price-candlesticks-history)
    - [Get oracle](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-oracle)
    - [Get exchange rate](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-exchange-rate)
    - [Get index components](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-index-components)
    - [Get economic calendar data](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-economic-calendar-data)
  + [WebSocket](https://www.okx.com/docs-v5/en/#public-data-websocket)
    - [Instruments channel](https://www.okx.com/docs-v5/en/#public-data-websocket-instruments-channel)
    - [Open interest channel](https://www.okx.com/docs-v5/en/#public-data-websocket-open-interest-channel)
    - [Funding rate channel](https://www.okx.com/docs-v5/en/#public-data-websocket-funding-rate-channel)
    - [Price limit channel](https://www.okx.com/docs-v5/en/#public-data-websocket-price-limit-channel)
    - [Option summary channel](https://www.okx.com/docs-v5/en/#public-data-websocket-option-summary-channel)
    - [Estimated delivery/exercise price channel](https://www.okx.com/docs-v5/en/#public-data-websocket-estimated-delivery-exercise-price-channel)
    - [Mark price channel](https://www.okx.com/docs-v5/en/#public-data-websocket-mark-price-channel)
    - [Index tickers channel](https://www.okx.com/docs-v5/en/#public-data-websocket-index-tickers-channel)
    - [Mark price candlesticks channel](https://www.okx.com/docs-v5/en/#public-data-websocket-mark-price-candlesticks-channel)
    - [Index candlesticks channel](https://www.okx.com/docs-v5/en/#public-data-websocket-index-candlesticks-channel)
    - [Liquidation orders channel](https://www.okx.com/docs-v5/en/#public-data-websocket-liquidation-orders-channel)
    - [ADL warning channel](https://www.okx.com/docs-v5/en/#public-data-websocket-adl-warning-channel)
    - [Economic calendar channel](https://www.okx.com/docs-v5/en/#public-data-websocket-economic-calendar-channel)
* [Trading Statistics](https://www.okx.com/docs-v5/en/#trading-statistics)
  + [REST API](https://www.okx.com/docs-v5/en/#trading-statistics-rest-api)
    - [Get support coin](https://www.okx.com/docs-v5/en/#trading-statistics-rest-api-get-support-coin)
    - [Get contract open interest history](https://www.okx.com/docs-v5/en/#trading-statistics-rest-api-get-contract-open-interest-history)
    - [Get taker volume](https://www.okx.com/docs-v5/en/#trading-statistics-rest-api-get-taker-volume)
    - [Get contract taker volume](https://www.okx.com/docs-v5/en/#trading-statistics-rest-api-get-contract-taker-volume)
    - [Get margin lending ratio](https://www.okx.com/docs-v5/en/#trading-statistics-rest-api-get-margin-lending-ratio)
    - [Get top traders contract long/short ratio](https://www.okx.com/docs-v5/en/#trading-statistics-rest-api-get-top-traders-contract-long-short-ratio)
    - [Get top traders contract long/short ratio (by position)](https://www.okx.com/docs-v5/en/#trading-statistics-rest-api-get-top-traders-contract-long-short-ratio-by-position)
    - [Get contract long/short ratio](https://www.okx.com/docs-v5/en/#trading-statistics-rest-api-get-contract-long-short-ratio)
    - [Get long/short ratio](https://www.okx.com/docs-v5/en/#trading-statistics-rest-api-get-long-short-ratio)
    - [Get contracts open interest and volume](https://www.okx.com/docs-v5/en/#trading-statistics-rest-api-get-contracts-open-interest-and-volume)
    - [Get options open interest and volume](https://www.okx.com/docs-v5/en/#trading-statistics-rest-api-get-options-open-interest-and-volume)
    - [Get put/call ratio](https://www.okx.com/docs-v5/en/#trading-statistics-rest-api-get-put-call-ratio)
    - [Get open interest and volume (expiry)](https://www.okx.com/docs-v5/en/#trading-statistics-rest-api-get-open-interest-and-volume-expiry)
    - [Get open interest and volume (strike)](https://www.okx.com/docs-v5/en/#trading-statistics-rest-api-get-open-interest-and-volume-strike)
    - [Get taker flow](https://www.okx.com/docs-v5/en/#trading-statistics-rest-api-get-taker-flow)
* [Funding Account](https://www.okx.com/docs-v5/en/#funding-account)
  + [REST API](https://www.okx.com/docs-v5/en/#funding-account-rest-api)
    - [Get currencies](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-currencies)
    - [Get balance](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-balance)
    - [Get non-tradable assets](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-non-tradable-assets)
    - [Get account asset valuation](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-account-asset-valuation)
    - [Funds transfer](https://www.okx.com/docs-v5/en/#funding-account-rest-api-funds-transfer)
    - [Get funds transfer state](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-funds-transfer-state)
    - [Asset bills details](https://www.okx.com/docs-v5/en/#funding-account-rest-api-asset-bills-details)
    - [Get deposit address](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-deposit-address)
    - [Get deposit history](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-deposit-history)
    - [Withdrawal](https://www.okx.com/docs-v5/en/#funding-account-rest-api-withdrawal)
    - [Cancel withdrawal](https://www.okx.com/docs-v5/en/#funding-account-rest-api-cancel-withdrawal)
    - [Get withdrawal history](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-withdrawal-history)
    - [Get deposit withdraw status](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-deposit-withdraw-status)
    - [Get exchange list (public)](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-exchange-list-public)
    - [Apply for monthly statement (last year)](https://www.okx.com/docs-v5/en/#funding-account-rest-api-apply-for-monthly-statement-last-year)
    - [GET monthly statement (last year)](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-monthly-statement-last-year)
    - [Get convert currencies](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-convert-currencies)
    - [Get convert currency pair](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-convert-currency-pair)
    - [Estimate quote](https://www.okx.com/docs-v5/en/#funding-account-rest-api-estimate-quote)
    - [Convert trade](https://www.okx.com/docs-v5/en/#funding-account-rest-api-convert-trade)
    - [Get convert history](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-convert-history)
    - [Get deposit payment methods](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-deposit-payment-methods)
    - [Get withdrawal payment methods](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-withdrawal-payment-methods)
    - [Create Withdrawal Order](https://www.okx.com/docs-v5/en/#funding-account-rest-api-create-withdrawal-order)
    - [Cancel Withdrawal Order](https://www.okx.com/docs-v5/en/#funding-account-rest-api-cancel-withdrawal-order)
    - [Get Withdrawal Order History](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-withdrawal-order-history)
    - [Get Withdrawal Order Detail](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-withdrawal-order-detail)
    - [Get Deposit Order History](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-deposit-order-history)
    - [Get Deposit Order Detail](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-deposit-order-detail)
  + [WebSocket](https://www.okx.com/docs-v5/en/#funding-account-websocket)
    - [Deposit info channel](https://www.okx.com/docs-v5/en/#funding-account-websocket-deposit-info-channel)
    - [Withdrawal info channel](https://www.okx.com/docs-v5/en/#funding-account-websocket-withdrawal-info-channel)
* [Sub-account](https://www.okx.com/docs-v5/en/#sub-account)
  + [REST API](https://www.okx.com/docs-v5/en/#sub-account-rest-api)
    - [Get sub-account list](https://www.okx.com/docs-v5/en/#sub-account-rest-api-get-sub-account-list)
    - [Reset the API Key of a sub-account](https://www.okx.com/docs-v5/en/#sub-account-rest-api-reset-the-api-key-of-a-sub-account)
    - [Get sub-account trading balance](https://www.okx.com/docs-v5/en/#sub-account-rest-api-get-sub-account-trading-balance)
    - [Get sub-account funding balance](https://www.okx.com/docs-v5/en/#sub-account-rest-api-get-sub-account-funding-balance)
    - [Get sub-account maximum withdrawals](https://www.okx.com/docs-v5/en/#sub-account-rest-api-get-sub-account-maximum-withdrawals)
    - [Get history of sub-account transfer](https://www.okx.com/docs-v5/en/#sub-account-rest-api-get-history-of-sub-account-transfer)
    - [Get history of managed sub-account transfer](https://www.okx.com/docs-v5/en/#sub-account-rest-api-get-history-of-managed-sub-account-transfer)
    - [Master accounts manage the transfers between sub-accounts](https://www.okx.com/docs-v5/en/#sub-account-rest-api-master-accounts-manage-the-transfers-between-sub-accounts)
    - [Set permission of transfer out](https://www.okx.com/docs-v5/en/#sub-account-rest-api-set-permission-of-transfer-out)
    - [Get custody trading sub-account list](https://www.okx.com/docs-v5/en/#sub-account-rest-api-get-custody-trading-sub-account-list)
* [Financial Product](https://www.okx.com/docs-v5/en/#financial-product)
  + [On-chain earn](https://www.okx.com/docs-v5/en/#financial-product-on-chain-earn)
    - [GET / Offers](https://www.okx.com/docs-v5/en/#financial-product-on-chain-earn-get-offers)
    - [POST / Purchase](https://www.okx.com/docs-v5/en/#financial-product-on-chain-earn-post-purchase)
    - [POST / Redeem](https://www.okx.com/docs-v5/en/#financial-product-on-chain-earn-post-redeem)
    - [POST / Cancel purchases/redemptions](https://www.okx.com/docs-v5/en/#financial-product-on-chain-earn-post-cancel-purchases-redemptions)
    - [GET / Active orders](https://www.okx.com/docs-v5/en/#financial-product-on-chain-earn-get-active-orders)
    - [GET / Order history](https://www.okx.com/docs-v5/en/#financial-product-on-chain-earn-get-order-history)
  + [ETH staking](https://www.okx.com/docs-v5/en/#financial-product-eth-staking)
    - [GET / Product info](https://www.okx.com/docs-v5/en/#financial-product-eth-staking-get-product-info)
    - [POST / Purchase](https://www.okx.com/docs-v5/en/#financial-product-eth-staking-post-purchase)
    - [POST / Redeem](https://www.okx.com/docs-v5/en/#financial-product-eth-staking-post-redeem)
    - [GET / Balance](https://www.okx.com/docs-v5/en/#financial-product-eth-staking-get-balance)
    - [GET / Purchase&Redeem history](https://www.okx.com/docs-v5/en/#financial-product-eth-staking-get-purchase-amp-redeem-history)
    - [GET / APY history (Public)](https://www.okx.com/docs-v5/en/#financial-product-eth-staking-get-apy-history-public)
  + [Simple earn flexible](https://www.okx.com/docs-v5/en/#financial-product-simple-earn-flexible)
    - [GET / Saving balance](https://www.okx.com/docs-v5/en/#financial-product-simple-earn-flexible-get-saving-balance)
    - [POST / Savings purchase/redemption](https://www.okx.com/docs-v5/en/#financial-product-simple-earn-flexible-post-savings-purchase-redemption)
    - [POST / Set lending rate](https://www.okx.com/docs-v5/en/#financial-product-simple-earn-flexible-post-set-lending-rate)
    - [GET / Lending history](https://www.okx.com/docs-v5/en/#financial-product-simple-earn-flexible-get-lending-history)
    - [GET / Public borrow info (public)](https://www.okx.com/docs-v5/en/#financial-product-simple-earn-flexible-get-public-borrow-info-public)
    - [GET / Public borrow history (public)](https://www.okx.com/docs-v5/en/#financial-product-simple-earn-flexible-get-public-borrow-history-public)
  + [Simple earn fixed](https://www.okx.com/docs-v5/en/#financial-product-simple-earn-fixed)
    - [GET / Lending offers (public)](https://www.okx.com/docs-v5/en/#financial-product-simple-earn-fixed-get-lending-offers-public)
    - [GET / Lending APY history (public)](https://www.okx.com/docs-v5/en/#financial-product-simple-earn-fixed-get-lending-apy-history-public)
    - [GET / Lending volume (public)](https://www.okx.com/docs-v5/en/#financial-product-simple-earn-fixed-get-lending-volume-public)
    - [POST / Place lending order](https://www.okx.com/docs-v5/en/#financial-product-simple-earn-fixed-post-place-lending-order)
    - [POST / Amend lending order](https://www.okx.com/docs-v5/en/#financial-product-simple-earn-fixed-post-amend-lending-order)
    - [GET / Lending order list](https://www.okx.com/docs-v5/en/#financial-product-simple-earn-fixed-get-lending-order-list)
    - [GET / Lending sub order list](https://www.okx.com/docs-v5/en/#financial-product-simple-earn-fixed-get-lending-sub-order-list)
  + [Flexible loan](https://www.okx.com/docs-v5/en/#financial-product-flexible-loan)
    - [GET / Borrowable currencies](https://www.okx.com/docs-v5/en/#financial-product-flexible-loan-get-borrowable-currencies)
    - [GET / Collateral assets](https://www.okx.com/docs-v5/en/#financial-product-flexible-loan-get-collateral-assets)
    - [POST / Maximum loan amount](https://www.okx.com/docs-v5/en/#financial-product-flexible-loan-post-maximum-loan-amount)
    - [POST / Adjust collateral](https://www.okx.com/docs-v5/en/#financial-product-flexible-loan-post-adjust-collateral)
    - [GET / Loan info](https://www.okx.com/docs-v5/en/#financial-product-flexible-loan-get-loan-info)
    - [GET / Loan history](https://www.okx.com/docs-v5/en/#financial-product-flexible-loan-get-loan-history)
    - [GET / Accrued interest](https://www.okx.com/docs-v5/en/#financial-product-flexible-loan-get-accrued-interest)
* [Affiliate](https://www.okx.com/docs-v5/en/#affiliate)
  + [REST API](https://www.okx.com/docs-v5/en/#affiliate-rest-api)
    - [Get the invitee's detail](https://www.okx.com/docs-v5/en/#affiliate-rest-api-get-the-invitee-39-s-detail)
    - [Get the user's affiliate rebate information](https://www.okx.com/docs-v5/en/#affiliate-rest-api-get-the-user-39-s-affiliate-rebate-information)
* [Status](https://www.okx.com/docs-v5/en/#status)
  + [GET / Status](https://www.okx.com/docs-v5/en/#status-get-status)
  + [WS / Status channel](https://www.okx.com/docs-v5/en/#status-ws-status-channel)
* [Announcement](https://www.okx.com/docs-v5/en/#announcement)
  + [GET / Announcements](https://www.okx.com/docs-v5/en/#announcement-get-announcements)
  + [GET / Announcement types](https://www.okx.com/docs-v5/en/#announcement-get-announcement-types)
* [Error Code](https://www.okx.com/docs-v5/en/#error-code)
  + [REST API](https://www.okx.com/docs-v5/en/#error-code-rest-api)
    - [Public](https://www.okx.com/docs-v5/en/#error-code-rest-api-public)
    - [Finance](https://www.okx.com/docs-v5/en/#error-code-rest-api-finance)
    - [Convert](https://www.okx.com/docs-v5/en/#error-code-rest-api-convert)
    - [Futures](https://www.okx.com/docs-v5/en/#error-code-rest-api-futures)
    - [Swap](https://www.okx.com/docs-v5/en/#error-code-rest-api-swap)
    - [Option](https://www.okx.com/docs-v5/en/#error-code-rest-api-option)
    - [Funding](https://www.okx.com/docs-v5/en/#error-code-rest-api-funding)
    - [Account](https://www.okx.com/docs-v5/en/#error-code-rest-api-account)
    - [Block Trading and Spread Orderbook](https://www.okx.com/docs-v5/en/#error-code-rest-api-block-trading-and-spread-orderbook)
    - [Copy trading](https://www.okx.com/docs-v5/en/#error-code-rest-api-copy-trading)
    - [Trading bot](https://www.okx.com/docs-v5/en/#error-code-rest-api-trading-bot)
  + [WebSocket](https://www.okx.com/docs-v5/en/#error-code-websocket)
    - [Public](https://www.okx.com/docs-v5/en/#error-code-websocket-public)

# Overview

Welcome to our V5 API documentation. OKX provides REST and WebSocket APIs to suit your trading needs.

* For users who complete registration on my.okx.com, please visit <https://my.okx.com/docs-v5/en/> for the V5 API documentation.

## API Resources and Support

### Tutorials

* Learn how to trade with V5 API: [Best practice to OKX’s v5 API](https://www.okx.com/docs-v5/trick_en/#instrument-configuration)
* Learn python spot trading step by step: [Python Spot Trading Tutorial](https://www.okx.com/help/how-can-i-do-spot-trading-with-the-jupyter-notebook)
* Learn python derivatives trading step by step: [Python Derivatives Trading Tutorial](https://www.okx.com/help/how-can-i-do-derivatives-trading-with-the-jupyter-notebook)

### Python libraries

* Use Python SDK for easier integration: [Python SDK](https://pypi.org/project/python-okx/)
* Get access to our market maker python sample code [Python market maker sample](https://github.com/okxapi/okx-sample-market-maker)

### Customer service

* Get support in our Telegram group: <https://t.me/OKXAPI>
* Subscribe to API related changes: <https://t.me/OKExAPIChannel>
* Please take 1 minute to help us improve: [V5 API Satisfaction Survey](https://forms.gle/Ehou2xFv5GE1xUGr9)
* If you have any questions, please consult online customer service

## V5 API Key Creation

Please refer to [my api page](https://www.okx.com/account/my-api) regarding V5 API Key creation.

#### Generating an API Key

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

* APIKey
* SecretKey
* Passphrase

The system returns randomly-generated APIKeys 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 APIKey.

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

Each APIKey can be linked with up to 20 IP addresses.  
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.)

## 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 APIKey.

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 prehash string of timestamp + method + requestPath + body (where + represents String concatenation).
* Prepare the SecretKey.
* Sign the prehash 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.

The requestPath is the path of requesting an endpoint.

Example: /api/v5/account/balance

The 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 APIKey.

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](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-order-channel)
2. [Account channel](https://www.okx.com/docs-v5/en/#trading-account-websocket-account-channel)
3. [Positions channel](https://www.okx.com/docs-v5/en/#trading-account-websocket-positions-channel)
4. [Balance and positions channel](https://www.okx.com/docs-v5/en/#trading-account-websocket-balance-and-position-channel)
5. [Position risk warning channel](https://www.okx.com/docs-v5/en/#trading-account-websocket-position-risk-warning)
6. [Account greeks channel](https://www.okx.com/docs-v5/en/#trading-account-websocket-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

{

"op": "login",

"args": [

{

"apiKey": "985d5b66-57ce-40fb-b714-afc0b9787083",

"passphrase": "123456",

"timestamp": "1538054050",

"sign": "7L+zFQ+CEgGu5rzCj4+BdV2/uUHGqddA9pI6ztsRRPs="

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation login |
| args | Array | 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 login error |
| 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 APIKey, 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

{

"op": "subscribe",

"args": ["<SubscriptionTopic>"]

}

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

{

"op":"subscribe",

"args":[

{

"channel":"tickers",

"instId":"BTC-USDT"

}

]

}

**Request parameters**

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name |
| > instType | String | No | Instrument type SPOT MARGIN SWAP FUTURES  OPTION  ANY |
| > instFamily | String | No | Instrument family Applicable to FUTURES/SWAP/OPTION |
| > instId | String | No | Instrument ID |

Response Example

{

"event": "subscribe",

"arg": {

"channel": "tickers",

"instId": "BTC-USDT"

},

"connId": "accb8e21"

}

**Return parameters**

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event, subscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name |
| > instType | String | No | Instrument type SPOT MARGIN SWAP FUTURES OPTION  ANY |
| > 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 |
| connId | String | Yes | WebSocket connection ID |

### Unsubscribe

Unsubscribe from one or more channels.

Request format description

{

"op": "unsubscribe",

"args": ["< SubscriptionTopic> "]

}

Request Example

{

"op": "unsubscribe",

"args": [

{

"channel": "tickers",

"instId": "BTC-USDT"

}

]

}

**Request parameters**

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation unsubscribe |
| args | Array | Yes | List of channels to unsubscribe from |
| > channel | String | Yes | Channel name |
| > instType | String | No | Instrument type SPOT MARGIN SWAP FUTURES  OPTION  ANY |
| > instFamily | String | No | Instrument family Applicable to FUTURES/SWAP/OPTION |
| > 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 |

## 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, Spot and 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

AWS URL:

* REST: https://aws.okx.com
* Public WebSocket: wss://wsaws.okx.com:8443/ws/v5/public
* Private WebSocket: wss://wsaws.okx.com:8443/ws/v5/private
* Business WebSocket: wss://wsaws.okx.com:8443/ws/v5/business

## Demo Trading Services

Currently, the V5 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 V5 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](https://www.okx.com/demo-trading-explorer/v5/en)

## 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.

**The introduction of instFamily:**

* There are no difference between uly and instFamily:
  + For BTC-USD-SWAP, uly and instFamily are both BTC-USD. For BTC-USDC-SWAP, uly and instFamily are both BTC-USDC.
  + If you set the request parameter "uly" as BTC-USD, you will get the data for BTC-USD (coin-margined) contracts.
  + If you set the request parameter "instFamily" as BTC-USD, then you also will get data for BTC-USD (coin-margined) contracts.
* You can look up the corresponding instFamily of each instrument from 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.

You should synchronize with our system time. Use [Get system time](https://www.okx.com/docs-v5/en/#public-data-rest-api-get-system-time) to obtain the current system time.

### 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. 1597026383085 |

The following endpoints are supported:

* [Place order](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-place-order)
* [Place multiple orders](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-place-multiple-orders)
* [Amend order](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-amend-order)
* [Amend multiple orders](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-amend-multiple-orders)

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' \ // request effective deadline

-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. 1597026383085 |

The following endpoints are supported:

* [Place order](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-place-order)
* [Place multiple orders](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-place-multiple-orders)
* [Amend order](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-amend-order)
* [Amend multiple orders](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-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](https://www.okx.com/docs-v5/en/#public-data-rest-api-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](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-place-order)
* [POST / Place multiple orders](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-place-multiple-orders)
* [POST / Amend order](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-amend-order)
* [POST / Amend multiple orders](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-amend-multiple-orders)
* [WS / Place order](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-place-order)
* [WS / Place multiple orders](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-place-multiple-orders)
* [WS / Amend order](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-amend-order)
* [WS / Amend multiple orders](https://www.okx.com/docs-v5/en/#order-book-trading-trade-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 | 1  Instrument ID:  BTC-USDT-SWAP  BTC-USD-SWAP  ETH-USDT-SWAP  ETH-USD-SWAP | 0.2 |
| Expiry Futures | Per instrument Family | 0.3  Instrument Family:  BTC-USDT  BTC-USD  ETH-USDT  ETH-USD | 0.1 |
| Spot | Per instrument ID | 0.5  Instrument ID:  BTC-USDT  ETH-USDT | 0.1 |
| Options | Per instrument Family |  | 0.1 |

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 |
| 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](https://www.okx.com/docs-v5/en/#order-book-trading-trade-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 -> 2500 order requests/2s
   2. Account B = max(2.13, 3.01) = 3.01 -> 1750 order requests/2s
   3. Account C = max(3.06, 3.01) = 3.06 -> 1750 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](https://www.okx.com/broker/home)
* [Broker rules](https://www.okx.com/help/introduction-of-rules-on-okx-brokers)
* 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: UserID + InstrumentType

#### HTTP Request

GET /api/v5/account/instruments

Request Example

GET /api/v5/account/instruments?instType=SPOT

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 | Instrument type SPOT: Spot MARGIN: Margin SWAP: Perpetual Futures FUTURES: Expiry Futures OPTION: Option |
| uly | String | Conditional | Underlying  Only applicable to FUTURES/SWAP/OPTION.If instType is OPTION, either uly or instFamily is required. |
| instFamily | String | Conditional | Instrument family Only applicable to FUTURES/SWAP/OPTION. If instType is OPTION, either uly or instFamily is required. |
| instId | String | No | Instrument ID |

Response Example

{

"code": "0",

"data": [

{

"auctionEndTime": "",

"baseCcy": "BTC",

"ctMult": "",

"ctType": "",

"ctVal": "",

"ctValCcy": "",

"expTime": "",

"instFamily": "",

"instId": "BTC-EUR",

"instType": "SPOT",

"lever": "",

"listTime": "1704876947000",

"lotSz": "0.00000001",

"maxIcebergSz": "9999999999.0000000000000000",

"maxLmtAmt": "1000000",

"maxLmtSz": "9999999999",

"maxMktAmt": "1000000",

"maxMktSz": "1000000",

"maxStopSz": "1000000",

"maxTriggerSz": "9999999999.0000000000000000",

"maxTwapSz": "9999999999.0000000000000000",

"minSz": "0.00001",

"optType": "",

"quoteCcy": "EUR",

"settleCcy": "",

"state": "live",

"ruleType": "normal",

"stk": "",

"tickSz": "1",

"uly": ""

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| instType | String | Instrument type |
| instId | String | Instrument ID, e.g. BTC-USD-SWAP |
| uly | String | Underlying, e.g. BTC-USD  Only applicable to MARGIN/FUTURES/SWAP/OPTION |
| instFamily | String | Instrument family, e.g. BTC-USD  Only applicable to MARGIN/FUTURES/SWAP/OPTION |
| baseCcy | String | Base currency, e.g. BTC inBTC-USDT  Only applicable to SPOT/MARGIN |
| quoteCcy | String | Quote currency, e.g. USDT in BTC-USDT  Only applicable to SPOT/MARGIN |
| settleCcy | String | Settlement and margin currency, e.g. BTC  Only applicable to FUTURES/SWAP/OPTION |
| ctVal | String | Contract value  Only applicable to FUTURES/SWAP/OPTION |
| ctMult | String | Contract multiplier  Only applicable to FUTURES/SWAP/OPTION |
| ctValCcy | String | Contract value currency  Only applicable to FUTURES/SWAP/OPTION |
| optType | String | Option type, C: Call P: put  Only applicable to OPTION |
| stk | String | Strike price  Only applicable to OPTION |
| listTime | String | Listing time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| auctionEndTime | String | The end time of call auction, Unix timestamp format in milliseconds, e.g. 1597026383085  Only applicable to SPOT that are listed through call auctions, return "" in other cases |
| expTime | String | Expiry time  Applicable to SPOT/MARGIN/FUTURES/SWAP/OPTION. For FUTURES/OPTION, it is natural delivery/exercise time. It is the instrument offline time when there is SPOT/MARGIN/FUTURES/SWAP/ manual offline. Update once change. |
| lever | String | Max Leverage,  Not applicable to SPOT, OPTION |
| tickSz | String | Tick size, e.g. 0.0001 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 SPOT/MARGIN, the value is the quantity in base currency. |
| minSz | String | Minimum order size If it is a derivatives contract, the value is the number of contracts. If it is SPOT/MARGIN, the value is the quantity in base currency. |
| ctType | String | Contract type linear: linear contract inverse: inverse contract  Only applicable to FUTURES/SWAP |
| state | String | Instrument status live  suspend preopen e.g. Futures and options contracts rollover from generation to trading start; certain symbols before they go live test: Test pairs, can't be traded |
| ruleType | String | Trading rule types normal: normal trading pre\_market: pre-market trading |
| 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 SPOT/MARGIN, the value is the quantity in base currency. |
| 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 SPOT/MARGIN, the value is the quantity in USDT. |
| maxLmtAmt | String | Max USD amount for a single limit order |
| maxMktAmt | String | Max USD amount for a single market order  Only applicable to SPOT/MARGIN |
| 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 SPOT/MARGIN, the value is the quantity in base currency.  The minimum order quantity of a single TWAP order is minSz\*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 SPOT/MARGIN, the value is the quantity in base currency. |
| 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 SPOT/MARGIN, the value is the quantity in base currency. |
| 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 SPOT/MARGIN, the value is the quantity in USDT. |

listTime and auctionEndTime  
For spot symbols listed through a call auction, listTime represents the start time of the auction, and auctionEndTime indicates the end of the auction and the start of continuous trading. For other scenarios, listTime will mark the beginning of continuous trading, and auctionEndTime 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: UserID

#### HTTP Requests

GET /api/v5/account/balance

Request Example

# Get the balance of all assets in the account

GET /api/v5/account/balance

# Get the balance of BTC and ETH assets in the account

GET /api/v5/account/balance?ccy=BTC,ETH

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. BTC or BTC,ETH. |

Response Example

{

"code": "0",

"data": [

{

"adjEq": "55415.624719833286",

"borrowFroz": "0",

"details": [

{

"availBal": "4834.317093622894",

"availEq": "4834.3170936228935",

"borrowFroz": "0",

"cashBal": "4850.435693622894",

"ccy": "USDT",

"crossLiab": "0",

"disEq": "4991.542013297616",

"eq": "4992.890093622894",

"eqUsd": "4991.542013297616",

"smtSyncEq": "0",

"spotCopyTradingEq": "0",

"fixedBal": "0",

"frozenBal": "158.573",

"imr": "",

"interest": "0",

"isoEq": "0",

"isoLiab": "0",

"isoUpl": "0",

"liab": "0",

"maxLoan": "0",

"mgnRatio": "",

"mmr": "",

"notionalLever": "",

"ordFrozen": "0",

"rewardBal": "0",

"spotInUseAmt": "",

"clSpotInUseAmt": "",

"maxSpotInUse": "",

"spotIsoBal": "0",

"stgyEq": "150",

"twap": "0",

"uTime": "1705449605015",

"upl": "-7.545600000000006",

"uplLiab": "0",

"spotBal": "",

"openAvgPx": "",

"accAvgPx": "",

"spotUpl": "",

"spotUplRatio": "",

"totalPnl": "",

"totalPnlRatio": ""

}

],

"imr": "8.57068529",

"isoEq": "0",

"mgnRatio": "143682.59776662575",

"mmr": "0.3428274116",

"notionalUsd": "85.7068529",

"ordFroz": "0",

"totalEq": "55837.43556134779",

"uTime": "1705474164160",

"upl": "-7.543562688000006"

}

],

"msg": ""

}

#### Response Parameters

| **Parameters** | **Types** | **Description** |
| --- | --- | --- |
| uTime | String | Update time of account information, millisecond format of Unix timestamp, e.g. 1597026383085 |
| totalEq | String | The total amount of equity in USD |
| isoEq | String | Isolated margin equity in USD Applicable to Spot and futures mode/Multi-currency margin/Portfolio margin |
| adjEq | String | Adjusted / Effective equity in USD  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 Spot mode/Multi-currency margin and Portfolio margin |
| ordFroz | String | Cross margin frozen for pending orders in USD  Only applicable to Spot mode/Multi-currency margin |
| imr | String | Initial margin requirement in USD  The sum of initial margins of all open positions and pending orders under cross-margin mode in USD.  Applicable to Spot mode/Multi-currency margin/Portfolio margin |
| mmr | String | Maintenance margin requirement in USD  The sum of maintenance margins of all open positions and pending orders under cross-margin mode in USD.  Applicable to Spot mode/Multi-currency margin/Portfolio margin |
| borrowFroz | String | Potential borrowing IMR of the account in USD  Only applicable to Spot mode/Multi-currency margin/Portfolio margin. It is "" for other margin modes. |
| mgnRatio | String | Margin ratio in USD  Applicable to Spot mode/Multi-currency margin/Portfolio margin |
| notionalUsd | String | Notional value of positions in USD  Applicable to Spot mode/Multi-currency margin/Portfolio margin |
| upl | String | Cross-margin info of unrealized profit and loss at the account level in USD Applicable to Multi-currency margin/Portfolio margin |
| details | Array | 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 Spot and futures mode/Multi-currency margin/Portfolio margin |
| > availEq | String | Available equity of currency Applicable to Spot and futures mode/Multi-currency margin/Portfolio margin |
| > disEq | String | Discount equity of currency in USD. |
| > fixedBal | String | Frozen balance for Dip Sniper and Peak Sniper |
| > availBal | String | Available balance of currency |
| > frozenBal | String | Frozen balance of currency |
| > ordFrozen | String | Margin frozen for open orders  Applicable to Spot mode/Spot and futures mode/Multi-currency margin |
| > liab | String | Liabilities of currency It is a positive value, e.g. 21625.64 Applicable to Spot mode/Multi-currency margin/Portfolio margin |
| > upl | String | The sum of the unrealized profit & loss of all margin and derivatives positions of currency.  Applicable to Spot and futures mode/Multi-currency margin/Portfolio margin |
| > uplLiab | String | Liabilities due to Unrealized loss of currency Applicable to Multi-currency margin/Portfolio margin |
| > crossLiab | String | Cross liabilities of currency Applicable to Spot mode/Multi-currency margin/Portfolio margin |
| > rewardBal | String | Trial fund balance |
| > isoLiab | String | Isolated liabilities of currency Applicable to Multi-currency margin/Portfolio margin |
| > mgnRatio | String | Cross margin ratio of currency  The index for measuring the risk of a certain asset in the account.  Applicable to Spot and futures mode and when there is cross position |
| > imr | String | Cross initial margin requirement at the currency level Applicable to Spot and futures mode and when there is cross position |
| > mmr | String | Cross maintenance margin requirement at the currency level Applicable to Spot and futures mode and when there is cross position |
| > interest | String | Accrued interest of currency It is a positive value, e.g. 9.01 Applicable to Spot mode/Multi-currency margin/Portfolio margin |
| > twap | String | Risk indicator of auto liability repayment Divided into multiple levels from 0 to 5, the larger the number, the more likely the auto repayment will be triggered.  Applicable to Spot mode/Multi-currency margin/Portfolio margin |
| > maxLoan | String | Max loan of currency Applicable to cross of Spot mode/Multi-currency margin/Portfolio margin |
| > eqUsd | String | Equity in USD of currency |
| > borrowFroz | String | Potential borrowing IMR of currency in USD  Applicable to Multi-currency margin/Portfolio margin. It is "" for other margin modes. |
| > notionalLever | String | Leverage of currency Applicable to Spot and futures mode |
| > stgyEq | String | Strategy equity |
| > isoUpl | String | Isolated unrealized profit and loss of currency Applicable to Spot and futures mode/Multi-currency margin/Portfolio margin |
| > spotInUseAmt | String | Spot in use amount Applicable to Portfolio margin |
| > clSpotInUseAmt | String | User-defined spot risk offset amount Applicable to Portfolio margin |
| > maxSpotInUse | String | Max possible spot risk offset amount Applicable to Portfolio margin |
| > spotIsoBal | String | Spot isolated balance Applicable to copy trading Applicable to Spot mode/Spot and futures mode. |
| > 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. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |
| > openAvgPx | Array | Spot average cost price. The unit is USD. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |
| > accAvgPx | Array | Spot accumulated cost price. The unit is USD. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |
| > spotUpl | String | Spot unrealized profit and loss. The unit is USD. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |
| > spotUplRatio | String | Spot unrealized profit and loss ratio. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |
| > totalPnl | String | Spot accumulated profit and loss. The unit is USD. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |
| > totalPnlRatio | String | Spot accumulated profit and loss ratio. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |

* Regarding more parameter details, you can refer to product documentations below:  
  [Spot and futures mode: cross margin trading](https://www.okx.com/help/iii-single-currency-margin-cross-margin-trading)   
  [Multi-currency margin mode: cross margin trading](https://www.okx.com/help/iv-multi-currency-margin-mode-cross-margin-trading)   
  [Multi-currency margin mode vs. Portfolio margin mode](https://www.okx.com/help/vi-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** | **Spot and 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 |
| ordFroz | Yes |  | Yes | Yes |
| imr | Yes |  | Yes | Yes |
| mmr | Yes |  | Yes | Yes |
| borrowFroz | Yes |  | Yes | Yes |
| mgnRatio | Yes |  | Yes | Yes |
| notionalUsd | 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 | Yes |
| > availBal | Yes | Yes | Yes | Yes |
| > frozenBal | Yes | Yes | Yes | Yes |
| > ordFrozen | Yes | Yes | Yes | Yes |
| > liab | Yes |  | Yes | Yes |
| > upl |  | Yes | Yes | Yes |
| > uplLiab |  |  | Yes | Yes |
| > crossLiab | Yes |  | 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 |
| > spotUpl | Yes | Yes | Yes | Yes |
| > spotUplRatio | Yes | Yes | Yes | Yes |
| > totalPnl | Yes | Yes | Yes | Yes |
| > totalPnlRatio | Yes | Yes | Yes | 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: UserID

#### HTTP Request

GET /api/v5/account/positions

Request Example

# Query BTC-USDT position information

GET /api/v5/account/positions?instId=BTC-USDT

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 | Instrument type MARGIN SWAP FUTURES OPTION instId will be checked against instType when both parameters are passed. |
| instId | String | No | Instrument ID, e.g. BTC-USDT-SWAP. Single instrument ID or multiple instrument IDs (no more than 10) separated with comma |
| posId | String | No | Single position ID or multiple position IDs (no more than 20) separated with comma.  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. |

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": "",

"idxPx": "62890.5",

"imr": "",

"instId": "BTC-USDT",

"instType": "MARGIN",

"interest": "0",

"last": "62892.9",

"lever": "5",

"liab": "-99.9998177776581948",

"liabCcy": "USDT",

"liqPenalty": "",

"liqPx": "53615.448336593756",

"margin": "0.000317654",

"markPx": "62891.9",

"maxSpotInUseAmt": "",

"mgnMode": "isolated",

"mgnRatio": "9.404143929947395",

"mmr": "0.0000318005395854",

"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": "-0.0000033452492717",

"uplLastPx": "-0.0000033199677697",

"uplRatio": "-0.0105311101755551",

"uplRatioLastPx": "-0.0104515220008934",

"usdPx": "",

"vegaBS": "",

"vegaPA": ""

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| instType | String | Instrument type |
| mgnMode | String | Margin mode cross  isolated |
| posId | String | Position ID |
| posSide | String | Position side long, pos is positive  short, pos is positive  net (FUTURES/SWAP/OPTION: positive pos means long position and negative pos means short position. For MARGIN, pos is always positive, posCcy being base currency means long position, posCcy 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 0 will be generated after the deposit is transferred |
| baseBal | String | (Deprecated) |
| quoteBal | String | (Deprecated) |
| baseBorrowed | String | (Deprecated) |
| baseInterest | String | (Deprecated) |
| quoteBorrowed | String | (Deprecated) |
| quoteInterest | String | (Deprecated) |
| posCcy | String | Position currency, only applicable to MARGIN positions. |
| availPos | String | Position that can be closed  Only applicable to MARGIN and OPTION.  For Margin position, the rest of sz will be SPOT 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 SPOT trading as much as possible. |
| avgPx | String | Average open price |
| 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. BTC-USDT-SWAP |
| lever | String | Leverage Not applicable to OPTION and positions of cross margin mode under Portfolio margin |
| liqPx | String | Estimated liquidation price  Not applicable to OPTION |
| imr | String | Initial margin requirement, only applicable to cross. |
| margin | String | Margin, can be added or reduced. Only applicable to isolated. |
| mgnRatio | String | Margin ratio |
| mmr | String | Maintenance margin requirement |
| liab | String | Liabilities, only applicable to MARGIN. |
| liabCcy | String | Liabilities currency, only applicable to MARGIN. |
| interest | String | Interest. Undeducted interest that has been incurred. |
| tradeId | String | Last trade ID |
| optVal | String | Option Value, only applicable to OPTION. |
| pendingCloseOrdLiabVal | String | The amount of close orders of isolated margin liability. |
| notionalUsd | String | Notional value of positions in USD |
| adl | String | Auto-deleveraging (ADL) indicator Divided into 5 levels, from 1 to 5, the smaller the number, the weaker the adl intensity. |
| ccy | String | Currency used for margin |
| last | String | Latest traded price |
| idxPx | String | Latest underlying index price |
| usdPx | String | Latest USD price of the ccy on the market, only applicable to OPTION |
| bePx | String | Breakeven price |
| deltaBS | String | delta: Black-Scholes Greeks in dollars, only applicable to OPTION |
| deltaPA | String | delta: Greeks in coins, only applicable to OPTION |
| 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 OPTION |
| thetaPA | String | theta：Greeks in coins, only applicable to OPTION |
| vegaBS | String | vega：Black-Scholes Greeks in dollars, only applicable to OPTION |
| vegaPA | String | vega：Greeks in coins, only applicable to OPTION |
| spotInUseAmt | String | Spot in use amount Applicable to Portfolio margin |
| spotInUseCcy | String | Spot in use unit, e.g. BTC Applicable to Portfolio margin |
| clSpotInUseAmt | String | User-defined spot risk offset amount Applicable to Portfolio margin |
| maxSpotInUseAmt | String | Max possible spot risk offset amount Applicable to Portfolio margin |
| bizRefId | String | External business id, e.g. experience coupon id |
| bizRefType | String | External business type |
| realizedPnl | String | Realized profit and loss Only applicable to FUTURES/SWAP/OPTION realizedPnl=pnl+fee+fundingFee+liqPenalty |
| pnl | String | Accumulated pnl of closing order(s) |
| 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 | Close position algo orders attached to the position. This array will have values only after you request "Place algo order" with closeFraction=1. |
| > algoId | String | Algo ID |
| > slTriggerPx | String | Stop-loss trigger price. |
| > slTriggerPxType | String | Stop-loss trigger price type.  last：last price index：index price mark：mark price |
| > tpTriggerPx | String | Take-profit trigger price. |
| > tpTriggerPxType | String | Take-profit trigger price type.  last：last price index：index price mark：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. 1597026383085 |
| uTime | String | Latest time position was adjusted, Unix timestamp format in milliseconds, e.g. 1597026383085 |

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: UserID

#### HTTP Request

GET /api/v5/account/positions-history

Request Example

GET /api/v5/account/positions-history

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 MARGIN SWAP FUTURES OPTION |
| instId | String | No | Instrument ID, e.g. BTC-USD-SWAP |
| mgnMode | String | No | Margin mode cross isolated |
| type | String | No | 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. |
| 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 uTime, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| before | String | No | Pagination of data to return records newer than the requested uTime, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| limit | String | No | Number of results per request. The maximum is 100. The default is 100. All records that have the same uTime will be returned at the current request |

Response Example

{

"code": "0",

"data": [

{

"cTime": "1654177169995",

"ccy": "BTC",

"closeAvgPx": "29786.5999999789081085",

"closeTotalPos": "1",

"instId": "BTC-USD-SWAP",

"instType": "SWAP",

"lever": "10.0",

"mgnMode": "cross",

"openAvgPx": "29783.8999999995535393",

"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"

}

],

"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 |
| 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 realizedPnl=pnl+fee+fundingFee+liqPenalty |
| 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 |
| pnlRatio | String | P&L ratio |
| 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: UserID

#### HTTP Requests

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

Request Example

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

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 USD Applicable to Multi-currency margin and Portfolio margin |
| balData | Array | Detailed asset information in all currencies |
| > ccy | String | Currency |
| > eq | String | Equity of currency |
| > disEq | String | Discount equity of currency in USD. |
| posData | Array | Detailed position information in all currencies |
| > instType | String | Instrument type |
| > mgnMode | String | Margin mode cross  isolated |
| > posId | String | Position ID |
| > instId | String | Instrument ID, e.g. BTC-USDT-SWAP |
| > pos | String | Quantity of positions contract. In the isolated margin mode, when doing manual transfers, a position with pos of 0 will be generated after the deposit is transferred |
| > baseBal | String | (Deprecated) |
| > quoteBal | String | (Deprecated) |
| > posSide | String | Position side long  short  net (FUTURES/SWAP/OPTION: positive pos means long position and negative pos means short position. MARGIN: posCcy being base currency means long position, posCcy being quote currency means short position.) |
| > posCcy | String | Position currency, only applicable to MARGIN positions. |
| > ccy | String | Currency used for margin |
| > notionalCcy | String | Notional value of positions in coin |
| > notionalUsd | String | Notional value of positions in USD |

### 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: UserID

#### HTTP Request

GET /api/v5/account/bills

Request Example

GET /api/v5/account/bills

GET /api/v5/account/bills?instType=MARGIN

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 SPOT MARGIN SWAP FUTURES OPTION |
| instId | String | No | Instrument ID, e.g. BTC-USDT |
| ccy | String | No | Bill currency |
| mgnMode | String | No | Margin mode isolated cross |
| ctType | String | No | Contract type linear inverse Only applicable to FUTURES/SWAP |
| type | String | No | Bill type 1: Transfer 2: Trade 3: Delivery 4: Forced repayment 5: Liquidation 6: Margin transfer 7: Interest deduction 8: Funding fee 9: ADL 10: Clawback 11: System token conversion 12: Strategy transfer 13: DDH 14: Block trade 15: Quick Margin 22: Repay 24: Spread trading 26: Structured products 27: Convert 28: Easy convert 29: One-click repay 30: Simple trade 250: Copy trader profit sharing expenses 251: Copy trader profit sharing refund |
| subType | String | No | Bill subtype 1: Buy 2: Sell 3: Open long 4: Open short 5: Close long 6: Close short 9: Interest deduction for Market loans 11: Transfer in 12: Transfer out 14: Interest deduction for VIP loans 160: Manual margin increase 161: Manual margin decrease 162: Auto margin increase 114: Forced repayment buy 115: Forced repayment sell 118: System token conversion transfer in 119: System token conversion transfer out 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 108:clawback 109: Liquidation fees 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 113: Delivery short 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 open 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 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 294: Fixed loan interest refund 295 : Fixed loan overdue penalty 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 318: Convert in 319: Convert out 320: Simple buy 321: Simple sell |
| 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 ts. Unix timestamp format in milliseconds, e.g. 1597026383085 |
| end | String | No | Filter with an end timestamp ts. 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",

"msg": "",

"data": [{

"bal": "8694.2179403378290202",

"balChg": "0.0219338232210000",

"billId": "623950854533513219",

"ccy": "USDT",

"clOrdId": "",

"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 |
| px | String | Price which related to subType   Trade filled price for1: 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 open 209: block trade close short 114: Forced repayment buy 115: Forced repayment sell   Liquidation Price for100: 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   Delivery price for112: Delivery long 113: Delivery short   Exercise price for170: Exercised 171: Counterparty exercised 172: Expired OTM   Mark price for173: Funding fee expense 174: Funding fee income |
| 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](https://www.okx.com/en/fees) |
| mgnMode | String | Margin mode isolated cross When bills are not generated by position changes, the field returns "" |
| instId | String | Instrument ID, e.g. BTC-USDT |
| ordId | String | Order ID Return order ID when the type is 2/5/9 Return "" when there is no order. |
| execType | String | Liquidity taker or maker T: taker M: maker |
| from | String | The remitting account 6: Funding account 18: Trading account Only applicable to transfer. When bill type is not transfer, the field returns "". |
| to | String | The beneficiary account 6: Funding account 18: Trading account Only applicable to transfer. When bill type is not transfer, the field returns "". |
| notes | String | Notes |
| interest | String | Interest |
| tag | String | Order tag |
| fillTime | String | Last filled time |
| tradeId | 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

### 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: UserID

#### HTTP Request

GET /api/v5/account/bills-archive

Request Example

GET /api/v5/account/bills-archive

GET /api/v5/account/bills-archive?instType=MARGIN

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 | Instrument type SPOT MARGIN SWAP FUTURES OPTION |
| instId | String | No | Instrument ID, e.g. BTC-USDT |
| ccy | String | No | Bill currency |
| mgnMode | String | No | Margin mode isolated cross |
| ctType | String | No | Contract type linear inverse Only applicable to FUTURES/SWAP |
| type | String | No | Bill type 1: Transfer 2: Trade 3: Delivery 4: Forced repayment 5: Liquidation 6: Margin transfer 7: Interest deduction 8: Funding fee 9: ADL 10: Clawback 11: System token conversion 12: Strategy transfer 13: DDH 14: Block trade 15: Quick Margin 22: Repay 24: Spread trading 26: Structured products 250: Copy trader profit sharing expenses 251: Copy trader profit sharing refund |
| subType | String | No | Bill subtype 1: Buy 2: Sell 3: Open long 4: Open short 5: Close long 6: Close short 9: Interest deduction for Market loans 11: Transfer in 12: Transfer out 14: Interest deduction for VIP loans 160: Manual margin increase 161: Manual margin decrease 162: Auto margin increase 114: Forced repayment buy 115: Forced repayment sell 118: System token conversion transfer in 119: System token conversion transfer out 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 108:clawback 109: Liquidation fees 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 113: Delivery short 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 open 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 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 294: Fixed loan interest refund 295 : Fixed loan overdue penalty 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 |
| 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 ts. Unix timestamp format in milliseconds, e.g. 1597026383085 |
| end | String | No | Filter with an end timestamp ts. 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",

"msg": "",

"data": [{

"bal": "8694.2179403378290202",

"balChg": "0.0219338232210000",

"billId": "623950854533513219",

"ccy": "USDT",

"clOrdId": "",

"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 |
| px | String | Price which related to subType   Trade filled price for1: 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 open 209: block trade close short 114: Forced repayment buy 115: Forced repayment sell   Liquidation Price for100: 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   Delivery price for112: Delivery long 113: Delivery short   Exercise price for170: Exercised 171: Counterparty exercised 172: Expired OTM   Mark price for173: Funding fee expense 174: Funding fee income |
| 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](https://www.okx.com/en/fees) |
| mgnMode | String | Margin mode isolated cross When bills are not generated by position changes, the field returns "" |
| instId | String | Instrument ID, e.g. BTC-USDT |
| ordId | String | Order ID Return order ID when the type is 2/5/9 Return "" when there is no order. |
| execType | String | Liquidity taker or maker T: taker M: maker |
| from | String | The remitting account 6: Funding account 18: Trading account Only applicable to transfer. When bill type is not transfer, the field returns "". |
| to | String | The beneficiary account 6: Funding account 18: Trading account Only applicable to transfer. When bill type is not transfer, the field returns "". |
| notes | String | Notes |
| interest | String | Interest |
| tag | String | Order tag |
| fillTime | String | Last filled time |
| tradeId | 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: UserID

#### Permissions: Read

#### HTTP Request

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

Request Example

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

body

{

"year":"2023",

"quarter":"Q1"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| year | String | Yes | 4 digits year |
| quarter | String | Yes | Quarter, valid value is Q1, Q2, Q3, Q4 |

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  true: Existed, can check from "Get bills details (since 2021)".  false: Does not exist and is generating, can check the download link after 2 hours The data of file is in reverse chronological order using billId. |
| ts | String | The first request time when the server receives. Unix timestamp format in milliseconds, e.g. 1597026383085 |

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: UserID

#### Permissions: Read

#### HTTP Request

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

Response Example

GET /api/v5/account/bills-history-archive?year=2023&quarter=Q4

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| year | String | Yes | 4 digits year |
| quarter | String | Yes | Quarter, valid value is Q1, Q2, Q3, Q4 |

Response Example

{

"code": "0",

"data": [

{

"fileHref": "http://xxx",

"state": "finished",

"ts": "1646892328000"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| fileHref | String | Download file link |
| ts | String | The first request time when the server receives. Unix timestamp format in milliseconds, e.g. 1597026383085 |
| state | String | Download link status  "finished" "ongoing" "failed": Failed, please apply again |

#### 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.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 |
| px | String | Price which related to subType   Trade filled price for1: 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 open 209: block trade close short 114: Forced repayment buy 115: Forced repayment sell   Liquidation Price for100: 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   Delivery price for112: Delivery long 113: Delivery short   Exercise price for170: Exercised 171: Counterparty exercised 172: Expired OTM   Mark price for173: Funding fee expense 174: Funding fee income |
| 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](https://www.okx.com/en/fees) |
| mgnMode | String | Margin mode isolated cross When bills are not generated by position changes, the field returns "" |
| instId | String | Instrument ID, e.g. BTC-USDT |
| ordId | String | Order ID Return order ID when the type is 2/5/9 Return "" when there is no order. |
| execType | String | Liquidity taker or maker T: taker M: maker |
| interest | String | Interest |
| tag | String | Order tag |
| fillTime | String | Last filled time |
| tradeId | 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 |

### Get account configuration

Retrieve current account configuration.

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/account/config

Request Example

GET /api/v5/account/config

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",

"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": [],

"traderInsts": [],

"uid": "44705892343619584"

}

],

"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: Spot and futures mode 3: Multi-currency margin 4: Portfolio margin |
| acctStpMode | String | Account self-trade prevention mode  cancel\_maker  cancel\_taker  cancel\_both  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 |
| level | String | The user level of the current real trading volume on the platform, e.g Lv1 |
| levelTmp | String | Temporary experience user level of special users, e.g Lv3 |
| ctIsoMode | String | Contract isolated margin trading settings automatic: Auto transfers autonomy: Manual transfers |
| mgnIsoMode | String | Margin isolated margin trading settings 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 2: Spot-Derivatives(Coin) to be offsetted 3: Only derivatives to be offsetted Only applicable to Portfolio margin |
| roleType | String | Role type 0: General user 1: Leading trader 2: Copy trader |
| traderInsts | Array | 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 | String | Spot lead trading instruments, only applicable to lead trader |
| opAuth | String | Whether the optional trading was activated 0: not activate 1: activated |
| kycLv | String | Main account KYC level 0: No verification 1: level 1 completed 2: level 2 completed 3: level 3 completed 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. |
| 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. 117.37.203.58,117.37.203.57. It is an empty string "" if there is no IP bonded. |
| perm | String | The permission of the current requesting API key or Access token read\_only: Read trade: Trade withdraw: Withdraw |
| liquidationGear | String | The margin ratio level of liquidation alert 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  0 means that there is not alert |
| enableSpotBorrow | Boolean | Whether borrow is allowed or not in Spot mode true: Enabled false: Disabled |
| spotBorrowAutoRepay | Boolean | Whether auto-repay is allowed or not in Spot mode true: Enabled false: Disabled |
| type | String | Account type  0: Main account  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 |

### Set position mode

Spot and 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: UserID

#### HTTP Request

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

Request Example

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

body

{

"posMode":"long\_short\_mode"

}

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 long\_short\_mode: long/short, only applicable to FUTURES/SWAP net\_mode: 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 Spot 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: UserID

#### HTTP Request

POST /api/v5/account/set-leverage

Request Example

# 1. Set leverage for `MARGIN` instruments under `isolated-margin` trade mode at pairs level.

POST /api/v5/account/set-leverage

body

{

"instId":"BTC-USDT",

"lever":"5",

"mgnMode":"isolated"

}

# 2. Set leverage for `MARGIN` instruments under `cross-margin` trade mode and Spot mode (enabled borrow) at currency level.

POST /api/v5/account/set-leverage

body

{

"ccy":"BTC",

"lever":"5",

"mgnMode":"cross"

}

# 3. Set leverage for `MARGIN` instruments under `cross-margin` trade mode and Spot and futures mode account mode at pairs level.

POST /api/v5/account/set-leverage

body

{

"instId":"BTC-USDT",

"lever":"5",

"mgnMode":"cross"

}

# 4. Set leverage for `MARGIN` instruments under `cross-margin` trade mode and Multi-currency margin at currency level.

POST /api/v5/account/set-leverage

body

{

"ccy":"BTC",

"lever":"5",

"mgnMode":"cross"

}

# 5. Set leverage for `MARGIN` instruments under `cross-margin` trade mode and Portfolio margin at currency level.

POST /api/v5/account/set-leverage

body

{

"ccy":"BTC",

"lever":"5",

"mgnMode":"cross"

}

# 6. Set leverage for `FUTURES` instruments under `cross-margin` trade mode at underlying level.

POST /api/v5/account/set-leverage

body

{

"instId":"BTC-USDT-200802",

"lever":"5",

"mgnMode":"cross"

}

# 7. Set leverage for `FUTURES` instruments under `isolated-margin` trade mode and buy/sell order placement mode at contract level.

POST /api/v5/account/set-leverage

body

{

"instId":"BTC-USDT-200802",

"lever":"5",

"mgnMode":"isolated"

}

# 8. Set leverage for `FUTURES` instruments under `isolated-margin` trade mode and long/short order placement mode at contract and position side level.

POST /api/v5/account/set-leverage

body

{

"instId":"BTC-USDT-200802",

"lever":"5",

"posSide":"long",

"mgnMode":"isolated"

}

# 9. Set leverage for `SWAP` instruments under `cross-margin` trade at contract level.

POST /api/v5/account/set-leverage

body

{

"instId":"BTC-USDT-SWAP",

"lever":"5",

"mgnMode":"cross"

}

# 10. Set leverage for `SWAP` instruments under `isolated-margin` trade mode and buy/sell order placement mode at contract level.

POST /api/v5/account/set-leverage

body

{

"instId":"BTC-USDT-SWAP",

"lever":"5",

"mgnMode":"isolated"

}

# 11. Set leverage for `SWAP` instruments under `isolated-margin` trade mode and long/short order placement mode at contract and position side level.

POST /api/v5/account/set-leverage

body

{

"instId":"BTC-USDT-SWAP",

"lever":"5",

"posSide":"long",

"mgnMode":"isolated"

}

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 Under cross mode, either instId or ccy is required; if both are passed, instId will be used by default. |
| ccy | String | Conditional | Currency used for margin, used for the leverage setting for the currency in auto borrow. Only applicable to cross MARGIN of Spot mode/Multi-currency margin/Portfolio margin Required when setting the leverage for automatically borrowing coin. |
| lever | String | Yes | Leverage |
| mgnMode | String | Yes | Margin mode isolated cross  Can only be cross if ccy is passed. |
| posSide | String | Conditional | Position side long short Only required when margin mode is isolated in long/short mode for FUTURES/SWAP. |

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 cross isolated |
| 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: UserID

#### HTTP Request

GET /api/v5/account/max-size

Request Example

GET /api/v5/account/max-size?instId=BTC-USDT&tdMode=isolated

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 smae instrument type separated with comma, e.g. BTC-USDT,ETH-USDT |
| tdMode | String | Yes | Trade mode cross isolated cash spot\_isolated |
| ccy | String | Conditional | Currency used for margin  Only applicable to MARGIN of Spot and futures mode. |
| px | String | No | Price When the price is not specified, it will be calculated according to the current limit price for FUTURES and SWAP, 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 MARGIN/FUTURES/SWAP |
| unSpotOffset | Boolean | No | true: disable Spot-Derivatives risk offset, false: enable Spot-Derivatives risk offset Default is false  Applicable to Portfolio  It is effective when Spot-Derivatives risk offset is turned on, otherwise this parameter is ignored. |

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 | SPOT/MARGIN: The maximum quantity in base currency that you can buy The cross-margin order under Spot and futures mode mode, quantity of coins is based on base currency. FUTURES/SWAP/OPTIONS: The maximum quantity of contracts that you can buy |
| maxSell | String | SPOT/MARGIN: The maximum quantity in quote currency that you can sell The cross-margin order under Spot and futures mode mode, quantity of coins is based on base currency. FUTURES/SWAP/OPTIONS: The maximum quantity of contracts that you can sell |

### 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: UserID

#### HTTP Request

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

Request Example

# Query maximum available transaction amount when cross MARGIN BTC-USDT use BTC as margin

GET /api/v5/account/max-avail-size?instId=BTC-USDT&tdMode=cross&ccy=BTC

# Query maximum available transaction amount for SPOT BTC-USDT

GET /api/v5/account/max-avail-size?instId=BTC-USDT&tdMode=cash

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. BTC-USDT,ETH-USDT |
| ccy | String | Conditional | Currency used for margin Only applicable to cross MARGIN of Spot and futures mode. |
| tdMode | String | Yes | Trade mode cross isolated cash spot\_isolated |
| reduceOnly | Boolean | No | Whether to reduce position only  Only applicable to MARGIN |
| px | String | No | The price of closing position.  Only applicable to reduceOnly MARGIN. |
| unSpotOffset | Boolean | No | true: disable Spot-Derivatives risk offset, false: enable Spot-Derivatives risk offset Default is false  Only applicable to Portfolio margin It is effective when Spot-Derivatives risk offset is turned on, otherwise this parameter is ignored. |
| quickMgnType | String | No | (Deprecated) |

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: UserID

#### HTTP Request

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

Request Example

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

body

{

"instId":"BTC-USDT-200626",

"posSide":"short",

"type":"add",

"amt":"1"

}

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 net long  short  net |
| type | String | Yes | add: add margin  reduce: reduce margin |
| amt | String | Yes | Amount to be increased or decreased. |
| ccy | String | Conditional | Currency, only applicable to MARGIN（Quick Margin Mode） |

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, long short |
| amt | String | Amount to be increase or decrease |
| type | String | add: add margin reduce: 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: UserID

#### HTTP Request

GET /api/v5/account/leverage-info

Request Example

GET /api/v5/account/leverage-info?instId=BTC-USDT-SWAP&mgnMode=cross

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 cross MARGIN of Spot mode/Multi-currency margin/Portfolio margin. Supported single currency or multiple currencies (no more than 20) separated with comma. |
| mgnMode | String | Yes | Margin mode cross isolated |

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 cross MARGIN of Spot mode/Multi-currency margin/Portfolio margin. |
| mgnMode | String | Margin mode |
| posSide | String | Position side long  short  net In long/short mode, the leverage in both directions long/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: UserID

#### HTTP Request

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

Request Example

GET /api/v5/account/adjust-leverage-info?instType=MARGIN&mgnMode=isolated&lever=3&instId=BTC-USDT

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instType | String | Yes | Instrument type MARGIN SWAP FUTURES |
| mgnMode | String | Yes | Margin mode isolated cross |
| lever | String | Yes | Leverage |
| instId | String | Conditional | Instrument ID, e.g. BTC-USDT It is required for these scenarioes: SWAP and FUTURES, Margin isolation, Margin cross in Spot and futures mode. |
| ccy | String | Conditional | Currency used for margin, e.g. BTC It is required for Margin cross in Spot and futures mode, Multi-currency margin and Portfolio margin |
| posSide | String | No | posSide net: The default value long 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 MARGIN |
| 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 MARGIN It is not applicable to the scenario when increasing leverage for isolated position under FUTURES and SWAP |
| 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 MARGIN 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 MARGIN, it is the estimated maximum loan in base currency under the corresponding leverage For SWAP and FUTURES, it is the estimated maximum quantity of contracts that can be opened under the corresponding leverage |
| estQuoteMaxAmt | String | The MARGIN estimated maximum loan in quote currency under the corresponding leverage. |
| existOrd | Boolean | Whether there is pending orders  true false |
| maxLever | String | Maximum leverage |
| minLever | String | Minimum leverage |

### Get the maximum loan of instrument

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/account/max-loan

Request Example

# Max loan of cross `MARGIN` for trading pair in `Spot mode` (enabled borrowing)

GET /api/v5/account/max-loan?instId=BTC-USDT&mgnMode=cross

# Max loan for currency in `Spot mode` (enabled borrowing)

GET /api/v5/account/max-loan?instId=USDT&mgnMode=cross

# Max loan of isolated `MARGIN` in `Spot and futures mode`

GET /api/v5/account/max-loan?instId=BTC-USDT&mgnMode=isolated

# Max loan of cross `MARGIN` in `Spot and futures mode` (Margin Currency is BTC)

GET /api/v5/account/max-loan?instId=BTC-USDT&mgnMode=cross&mgnCcy=BTC

# Max loan of cross `MARGIN` in `Multi-currency margin`

GET /api/v5/account/max-loan?instId=BTC-USDT&mgnMode=cross

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 Spot and 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 isolated cross |
| instId | String | Conditional | Single instrument or multiple instruments (no more than 5) separated with comma, e.g. BTC-USDT,ETH-USDT |
| ccy | String | Conditional | Currency Applicable to get Max loan of manual borrow for the currency in Spot mode (enabled borrowing) |
| mgnCcy | String | Conditional | Margin currency Only applicable to cross MARGIN in Spot and futures mode |

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 buy sell |

### Get fee rates

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/account/trade-fee

Request Example

# Query trade fee rate of SPOT BTC-USDT

GET /api/v5/account/trade-fee?instType=SPOT&instId=BTC-USDT

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 SPOT MARGIN SWAP FUTURES OPTION |
| instId | String | No | Instrument ID, e.g. BTC-USDT Applicable to SPOT/MARGIN |
| uly | String | No | Underlying, e.g. BTC-USD Applicable to FUTURES/SWAP/OPTION |
| instFamily | String | No | Instrument family, e.g. BTC-USD Applicable to FUTURES/SWAP/OPTION |
| ruleType | String | Yes | Trading rule types normal: normal trading pre\_market: pre-market trading ruleType can not be passed through together with instId/instFamily/uly |

Response Example

{

"code": "0",

"msg": "",

"data": [{

"category": "1", //Deprecated

"delivery": "",

"exercise": "",

"instType": "SPOT",

"level": "lv1",

"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 SPOT/MARGIN, it is taker fee rate of the USDT trading pairs.  For FUTURES/SWAP/OPTION, it is the fee rate of crypto-margined contracts |
| maker | String | For SPOT/MARGIN, it is maker fee rate of the USDT trading pairs.  For FUTURES/SWAP/OPTION, it is the fee rate of crypto-margined contracts |
| takerU | String | Taker fee rate of USDT-margined contracts, only applicable to FUTURES/SWAP |
| makerU | String | Maker fee rate of USDT-margined contracts, only applicable to FUTURES/SWAP |
| delivery | String | Delivery fee rate |
| exercise | String | Fee rate for exercising the option |
| instType | String | Instrument type |
| takerUSDC | String | For SPOT/MARGIN, it is taker fee rate of the USDⓈ&Crypto trading pairs. For FUTURES/SWAP, it is the fee rate of USDC-margined contracts |
| makerUSDC | String | For SPOT/MARGIN, it is maker fee rate of the USDⓈ&Crypto trading pairs. For FUTURES/SWAP, it is the fee rate of USDC-margined contracts |
| ruleType | String | Trading rule types normal: normal trading pre\_market: pre-market trading |
| ts | String | Data return time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| category | String | Currency category. Note: this parameter is already deprecated |
| fiat | Array | 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. USDⓈ represent the stablecoin besides USDT and USDC

### Get interest accrued data

Get interest accrued data. Only data within the last one year can be obtained.

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/account/interest-accrued

Request Example

GET /api/v5/account/interest-accrued

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 toMARGIN |
| 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

{

"code": "0",

"data": [

{

"ccy": "USDT",

"instId": "",

"interest": "0.0003960833333334",

"interestRate": "0.0000040833333333",

"liab": "97",

"mgnMode": "",

"ts": "1637312400000",

"type": "1"

},

{

"ccy": "USDT",

"instId": "",

"interest": "0.0004083333333334",

"interestRate": "0.0000040833333333",

"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. BTC |
| instId | String | Instrument ID, e.g. BTC-USDT Only applicable to Market loans |
| mgnMode | String | Margin mode cross  isolated |
| interest | String | Interest |
| interestRate | String | Interest rate (in hour) |
| liab | String | Liability |
| ts | String | Timestamp for interest accured, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### Get interest rate

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

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Requests

GET /api/v5/account/interest-rate

Request Example

GET /api/v5/account/interest-rate

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. BTC |

{

"code":"0",

"msg":"",

"data":[

{

"ccy":"BTC",

"interestRate":"0.0001"

},

{

"ccy":"LTC",

"interestRate":"0.0003"

}

]

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| interestRate | String | interest rate(the current hour) |
| ccy | String | currency |

### Set greeks (PA/BS)

Set the display type of Greeks.

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/account/set-greeks

Request Example

POST /api/v5/account/set-greeks

body

{

"greeksType":"PA"

}

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: UserID

#### HTTP Request

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

Request Example

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

body

{

"isoMode":"automatic",

"type":"MARGIN"

}

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 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: UserID

#### HTTP Request

GET /api/v5/account/max-withdrawal

Request Example

GET /api/v5/account/max-withdrawal

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. BTC or BTC,ETH. |

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 Spot mode/Multi-currency margin/Portfolio margin) |
| maxWdEx | String | Max withdrawal (including borrowed assets under Spot mode/Multi-currency margin/Portfolio margin) |
| spotOffsetMaxWd | String | Max withdrawal under Spot-Derivatives risk offset mode (excluding borrowed assets under Portfolio margin) Applicable to Portfolio margin |
| spotOffsetMaxWdEx | String | Max withdrawal under Spot-Derivatives risk offset mode (including borrowed assets under Portfolio margin) Applicable to Portfolio margin |

### Get account risk state

Only applicable to Portfolio margin account

#### Rate Limit: 10 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Requests

GET /api/v5/account/risk-state

Request Example

GET /api/v5/account/risk-state

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 | String | 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 | derivatives risk unit list |
| atRiskMgn | Array | margin risk unit list |
| ts | String | Unix timestamp format in milliseconds, e.g.1597026383085 |

### Manual borrow and repay in Quick Margin Mode

Please note that this endpoint will be deprecated soon.

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/account/quick-margin-borrow-repay

Request Example

POST /api/v5/account/quick-margin-borrow-repay

body

{

"instId":"BTC-USDT",

"ccy":"USDT",

"side":"borrow",

"amt":"100"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instId | String | Yes | Instrument ID, e.g. BTC-USDT |
| ccy | String | Yes | Loan currency, e.g. BTC |
| side | String | Yes | borrow repay |
| amt | String | Yes | borrow/repay amount |

Response Example

{

"code": "0",

"data": [

{

"amt": "100",

"instId":"BTC-USDT",

"ccy": "USDT",

"side": "borrow"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| instId | String | Instrument ID, e.g. BTC-USDT |
| ccy | String | Loan currency, e.g. BTC |
| side | String | borrow repay |
| amt | String | borrow/repay amount |

### Get borrow and repay history in Quick Margin Mode

Get record in the past 3 months.

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/account/quick-margin-borrow-repay-history

Request Example

GET /api/v5/account/quick-margin-borrow-repay-history

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instId | String | No | Instrument ID, e.g. BTC-USDT |
| ccy | String | No | Loan currency, e.g. BTC |
| side | String | No | borrow repay |
| after | String | No | Pagination of data to return records earlier than the requested refId |
| before | String | No | Pagination of data to return records newer than the requested refId |
| 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": [

{

"instId": "BTC-USDT",

"ccy": "USDT",

"side": "borrow",

"accBorrowed": "0.01",

"amt": "0.005",

"refId": "1637310691470124",

"ts": "1637310691470"

},

{

"instId": "BTC-USDT",

"ccy": "USDT",

"side": "borrow",

"accBorrowed": "0.01",

"amt": "0.005",

"refId": "1637310691470123",

"ts": "1637310691400"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| instId | String | Instrument ID, e.g. BTC-USDT |
| ccy | String | Loan currency, e.g. BTC |
| side | String | borrow repay |
| accBorrowed | String | Accumulate borrow amount |
| amt | String | borrow/repay amount |
| refId | String | The ID of borrowing or repayment |
| ts | String | Timestamp for Borrow/Repay |

### Get borrow interest and limit

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/account/interest-limits

Request Example

GET /api/v5/account/interest-limits?type=1&ccy=BTC

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(

type="1",

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. BTC |

Response Example

{

"code": "0",

"data": [

{

"debt": "0.85893159114900247077000000000000",

"interest": "0.00000000000000000000000000000000",

"loanAlloc": "",

"nextDiscountTime": "1729490400000",

"nextInterestTime": "1729490400000",

"records": [

{

"availLoan": "",

"avgRate": "",

"ccy": "BTC",

"interest": "0",

"loanQuota": "175.00000000",

"posLoan": "",

"rate": "0.0000276",

"surplusLmt": "175.00000000",

"surplusLmtDetails": {},

"usedLmt": "0.00000000",

"usedLoan": ""

}

]

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| debt | String | Current debt in USDT |
| interest | String | Current interest in USDT, the unit is USDT Only applicable to Market loans |
| nextDiscountTime | String | Next deduct time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| nextInterestTime | String | Next accrual time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| 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 | Details for currencies |
| > ccy | String | Loan currency, e.g. BTC |
| > rate | String | Current daily 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 |
| > surplusLmtDetails | Object | The details of available amount across all sub-accounts The value of surplusLmt is the minimum value within this array. It can help you judge the reason that surplusLmt is not enough. Only applicable to VIP loans |
| >> allAcctRemainingQuota | String | Total remaining quota for master account and sub-accounts |
| >> curAcctRemainingQuota | String | The remaining quota for the current account. Only applicable to the case in which the sub-account is assigned the loan allocation |
| >> platRemainingQuota | String | Remaining quota for the platform. The format like "600" will be returned when it is more than curAcctRemainingQuota or allAcctRemainingQuota |
| > usedLmt | String | Borrowed amount across all sub-accounts 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 Market loans |
| > posLoan | String | Frozen amount for current account (Within the locked quota)  Only applicable to VIP loans |
| > availLoan | String | Available amount for current account (Within the locked quota) Only applicable to VIP loans |
| > usedLoan | String | Borrowed amount for current account Only applicable to VIP loans |
| > avgRate | String | Average (hour) interest of already borrowed coin only applicable to VIP loans |

### Get fixed loan borrow limit

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/account/fixed-loan/borrowing-limit

Request Example

GET /api/v5/account/fixed-loan/borrowing-limit

#### Request Parameters

None

Response Example

{

"code": "0",

"data": [

{

"availRepay": "1110.9884",

"borrowed": "60895.1139",

"details": [

{

"availBorrow": "0.0566",

"borrowed": "0",

"ccy": "BTC",

"minBorrow": "0.1433",

"used": "0",

"term": "30D"

},

{

"availBorrow": "0",

"borrowed": "0",

"ccy": "LTC",

"minBorrow": "114.582",

"used": "0",

"term": "30D"

},

{

"availBorrow": "10",

"borrowed": "0",

"ccy": "ETH",

"minBorrow": "2.6666",

"used": "0",

"term": "30D"

},

{

"availBorrow": "248727.5",

"borrowed": "61115",

"ccy": "USDT",

"minBorrow": "9999.5679",

"used": "60000",

"term": "30D"

}

],

"totalAvailBorrow": "289336.6564",

"totalBorrowLmt": "22843016.1921",

"ts": "1716368077071",

"used": "59784.1256"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| totalBorrowLmt | String | (Master account and sub-accounts) Total borrow limit, unit in USD |
| totalAvailBorrow | String | (Master account and sub-accounts) Total available amount to borrow, unit in USD |
| borrowed | String | (Current account) Borrowed amount, unit in USD |
| used | String | (Current account) Used amount, unit in USD |
| availRepay | String | (Current account) Available amount to repay, unit in USD |
| details | Array of object | Borrow limit info in details |
| > ccy | String | Borrowing currency, e.g. BTC |
| > used | String | Used amount of current currency |
| > borrowed | String | Borrowed amount of current currency |
| > availBorrow | String | Available amount to borrow of current currency |
| > minBorrow | String | Minimum borrow amount |
| > term | String | Borrowing term, e.g. 30D: 30 Days |
| ts | String | Data return time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### Get fixed loan borrow quote

#### Rate Limit: 2 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/account/fixed-loan/borrowing-quote

Request Example

# Quote for new order

GET /api/v5/account/fixed-loan/borrowing-quote?type=normal&ccy=BTC&maxRate=0.1&amt=0.1&term=30D

# Quote for renew order

GET /api/v5/account/fixed-loan/borrowing-quote?type=reborrow&ordId=2405162053378222

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| type | String | Yes | Type normal: new order reborrow: renew existing order |
| ccy | String | Optional | Borrowing currency, e.g. BTC if type=normal, the parameter is required. |
| amt | String | Optional | Borrowing amount if type=normal, the parameter is required. |
| maxRate | String | Optional | Maximum acceptable borrow annual rate, in decimal. e.g. 0.01 represents 1% if type=normal, the parameter is required. |
| term | String | Optional | Fixed term for borrowing order 30D：30 Days if type=normal, the parameter is required. |
| ordId | String | Optional | Order ID if type=reborrow, the parameter is required. |

Response Example

{

"code": "0",

"msg": "",

"data": [

{

"ccy": "BTC",

"term":"30D",

"estAvailBorrow":"0.1",

"estRate": "0.01",

"estInterest": "25",

"penaltyInterest": "",

"ts": "1629966436396"

}

]

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ccy | String | Borrowing currency, e.g. BTC |
| term | String | Fixed term for borrowing order 30D：30 Days |
| estAvailBorrow | String | Estimated available borrowing amount |
| estRate | String | Estimated borrowing annual rate, e.g. "0.01" |
| estInterest | String | Estimated borrowing interest |
| penaltyInterest | String | Penalty interest for reborrowing Applicate to type = reborrow, else return "" |
| ts | String | Data return time，Unix timestamp format in milliseconds, e.g. 1597026383085 |

### Place fixed loan borrowing order

For new borrowing orders, they belong to the IOC (immediately close and cancel the remaining) type. For renewal orders, they belong to the FOK (Fill-or-kill) type.  
Order book may refer to [Get lending volume (public)](https://www.okx.com/docs-v5/en/#financial-product-simple-earn-fixed-get-lending-volume-public).

#### Rate Limit: 2 requests per second

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/account/fixed-loan/borrowing-order

Request Example

POST /api/v5/account/fixed-loan/borrowing-order

body

{

"ccy": "BTC",

"amt": "0.1",

"maxRate": "0.01",

"term": "30D",

"reborrow": true,

"reborrowRate": "0.01"

}

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | Yes | Borrowing currency, e.g. BTC |
| amt | String | Yes | Borrowing amount |
| maxRate | String | Yes | Maximum acceptable borrow annual rate, in decimal. e.g. 0.01 represents 1%. |
| term | String | Yes | Fixed term for borrowing order 30D：30 Days |
| reborrow | Boolean | No | Whether or not auto-renew when the term is due true: auto-renew false: not auto-renew Default is false. |
| reborrowRate | String | No | Auto-renew borrowing rate, in decimal. e.g. 0.01 represents 1%. If reborrow is true, the parameter is required. |

Response Example

{

"code": "0",

"data": [

{

"ordId":"2405162053378222"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ordId | String | Borrowing order ID |

### Amend fixed loan borrowing order

#### Rate Limit: 2 requests per second

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/account/fixed-loan/amend-borrowing-order

Request Example

POST /api/v5/account/fixed-loan/amend-borrowing-order

body

{

"ordId": "2405162053378222",

"reborrow": true,

"renewMaxRate": "0.01"

}

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ordId | String | Yes | Borrowing order ID |
| reborrow | Boolean | No | Whether or not reborrowing when the term is due. Default is false. |
| renewMaxRate | String | No | Maximum acceptable auto-renew borrow annual rate for borrowing order, in decimal. e.g. 0.01 represents 1%. If reborrow is true, the parameter is required. |

Response Example

{

"code": "0",

"data": [

{

"ordId":"2405162053378222"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ordId | String | Borrowing order ID |

### Manual renew fixed loan borrowing order

#### Rate Limit: 2 requests per second

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/account/fixed-loan/manual-reborrow

Request Example

POST /api/v5/account/fixed-loan/manual-reborrow

body

{

"ordId": "2405162053378222",

"maxRate": "0.01"

}

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ordId | String | Yes | Borrowing order ID |
| maxRate | String | Yes | Maximum acceptable borrowing annual rate, in decimal. e.g. 0.01 represents 1%. |

Response Example

{

"code": "0",

"data": [

{

"ordId":"2405162053378222"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ordId | String | Borrowing order ID |

### Repay fixed loan borrowing order

#### Rate Limit: 2 requests per second

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/account/fixed-loan/repay-borrowing-order

Request Example

POST /api/v5/account/fixed-loan/repay-borrowing-order

body

{

"ordId": "2405162053378222"

}

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ordId | String | Yes | Borrowing order ID |

Response Example

{

"code": "0",

"data": [

{

"ordId":"2405162053378222"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ordId | String | Borrowing order ID |

### Convert fixed loan to market loan

#### Rate Limit: 2 requests per second

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/account/fixed-loan/convert-to-market-loan

Request Example

POST /api/v5/account/fixed-loan/convert-to-market-loan

body

{

"ordId": "2409141848234868"

}

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ordId | String | Yes | Borrowing order ID |

Response Example

{

"code": "0",

"data": [

{

"ordId": "2409141848234868"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ordId | String | Borrowing order ID |

### Reduce liabilities for fixed loan

Provide the function of "setting pending repay state / canceling pending repay state" for fixed loan order.

Setting pending repay state  
  
• When your order status changes to pending repayment, and the repayment priority changes to fixed Loan first, followed by market loans. New liabilities are then allocated to market loans before fixed Loan.  
  
• After repayment, the repayment priority of this order’s crypto reverts to market loans first, followed by fixed Loan, with new liabilities allocated to fixed Loan before market loans.  
  
• When multiple orders are pending repayment, the order with the earliest maturity date takes priority. If the maturity date is the same, the order with the highest interest rate takes priority. If the interest rates are also the same, the largest order takes priority. Once you repay all the orders or cancel repayment, the repayment priority reverts to the default status.  
  
• You can close your margin position or transfer in the borrowed crypto to reduce liability. If the reduced liabilities are greater than the borrowing amount, repayment will be automatic.  
  
• Orders that are already in a pending repay state cannot be manually repaid. If manual repayment is required, the pending repay state should be cancelled.

#### Rate Limit: 2 requests per second

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/account/fixed-loan/reduce-liabilities

Request Example

POST /api/v5/account/fixed-loan/reduce-liabilities

body

{

"ordId": "2409141802194350",

"pendingRepay": true

}

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ordId | String | Yes | Borrowing order ID |
| pendingRepay | String | Yes | true: Set order state to pending repay false: Cancel pending repay state |

Response Example

{

"code": "0",

"data": [

{

"ordId": "2409141802194350",

"pendingRepay": true

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ordId | String | Borrowing order ID |
| pendingRepay | String | true: Set order state to pending repay false: Cancel pending repay state |

### Get fixed loan borrow order list

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/account/fixed-loan/borrowing-orders-list

Request Example

GET /api/v5/account/fixed-loan/borrowing-orders-list

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ordId | String | No | Borrowing order ID |
| ccy | String | No | Borrowing currency, e.g. BTC |
| state | String | No | State 1: Borrowing 2: Borrowed 3: Settled (Repaid) 4: Borrow failed 5: Overdue 6: Settling 7: Reborrowing 8: Pending repay (more details refer to [Reduce liabilities for fixed loan](https://www.okx.com/docs-v5/en/#trading-account-rest-api-reduce-liabilities-for-fixed-loan)) |
| term | String | No | Borrowing term, e.g. 30D: 30 Days |
| after | String | No | Pagination of data to return records earlier than the requested ordId |
| before | String | No | Pagination of data to return records newer than the requested ordId |
| limit | String | No | Number of results per request. The maximum is 100. The default is 100. |

Response Example

{

"code": "0",

"data": [

{

"accruedInterest": "0.0065753424657534",

"actualBorrowAmt": "0",

"cTime": "1720701491000",

"ccy": "ETH",

"curRate": "0",

"deadlinePenaltyInterest": "1723271891000",

"earlyRepayPenaltyInterest": "",

"expiryTime": "1723293491000",

"failedReason": "1",

"forceRepayTime": "1725885491000",

"ordId": "2407112038109533",

"overduePenaltyInterest": "",

"potentialPenaltyInterest": "",

"reborrow": false,

"reborrowRate": "",

"reqBorrowAmt": "8",

"settleReason": "",

"state": "4",

"term": "30D",

"uTime": "1720701490000"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ccy | String | Borrowing currency, e.g. BTC |
| ordId | String | Borrowing order ID |
| term | String | Fixed term for borrowing order 30D: 30 days |
| state | String | State 1: Borrowing 2: Borrowed 3: Settled (Repaid) 4: Borrow failed 5: Overdue 6: Settling 7: Reborrowing 8: Pending repay (more details refer to [Reduce liabilities for fixed loan](https://www.okx.com/docs-v5/en/#trading-account-rest-api-reduce-liabilities-for-fixed-loan)) |
| failedReason | String | Borrowing failed reason 1: There are currently no matching orders 2: Unable to pay prepaid interest -1: Other reason |
| settleReason | String | Settle reason 1: Order at maturity 2: Extension in advance 3: Early repayment |
| curRate | String | Borrowing annual rate of current order |
| accruedInterest | String | Accrued interest |
| reqBorrowAmt | String | Requested borrowing mount |
| actualBorrowAmt | String | Actual borrowed mount |
| reborrow | Boolean | Whether or not auto-renew when the term is due true: auto-renew false: not auto-renew |
| reborrowRate | String | Auto-renew borrowing rate, in decimal. e.g. 0.01 represents 1% |
| expiryTime | String | Expiry time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| forceRepayTime | String | Force repayment time, unix timestamp format in milliseconds, e.g. 1597026383085 |
| deadlinePenaltyInterest | String | Deadline of penalty interest for early repayment, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| potentialPenaltyInterest | String | Potential penalty Interest for early repayment |
| overduePenaltyInterest | String | Overdue penalty interest |
| earlyRepayPenaltyInterest | String | Early repay penalty interest |
| cTime | String | Creation time for borrowing order, unix timestamp format in milliseconds, e.g. 1597026383085 |
| uTime | String | Update time for borrowing order, unix timestamp format in milliseconds, e.g. 1597026383085 |

### Manual borrow / repay

Only applicable to Spot mode (enabled borrowing)

#### Rate Limit: 1 request per second

#### Rate limit rule: UserID

#### HTTP Request

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

Request Example

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

body

{

"ccy":"USDT",

"side":"borrow",

"amt":"100"

}

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. BTC |
| side | String | Yes | Side borrow repay |
| 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. BTC |
| side | String | Side borrow repay |
| amt | String | Actual amount |

### Set auto repay

Only applicable to Spot mode (enabled borrowing)

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

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

Request Example

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

body

{

"autoRepay": true

}

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 Spot mode true: Enable auto repay false: 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 Spot mode true: Enable auto repay false: 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: UserID

#### HTTP Request

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

Request Example

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

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. BTC |
| type | String | No | Event type auto\_borrow auto\_repay manual\_borrow manual\_repay |
| after | String | No | Pagination of data to return records earlier than the requested ts (included), Unix timestamp format in milliseconds, e.g. 1597026383085 |
| before | String | No | Pagination of data to return records newer than the requested ts(included), 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": [

{

"accBorrowed": "0",

"amt": "6764.802661157592",

"ccy": "USDT",

"ts": "1725330976644",

"type": "auto\_repay"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ccy | String | Currency, e.g. BTC |
| type | String | Event type auto\_borrow auto\_repay manual\_borrow manual\_repay |
| amt | String | Amount |
| accBorrowed | String | Accumulated borrow amount |
| ts | String | Timestamp for the event, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### 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: UserID

#### Permissions: Read

#### HTTP Request

POST /api/v5/account/position-builder

Request Example

# Both real and virtual positions and assets are calculated

POST /api/v5/account/position-builder

body

{

"inclRealPosAndEq": false,

"simPos":[

{

"pos":"-10",

"instId":"BTC-USDT-SWAP"

},

{

"pos":"10",

"instId":"LTC-USDT-SWAP"

}

],

"simAsset":[

{

"ccy": "USDT",

"amt": "100"

}

],

"spotOffsetType":"1",

"greeksType":"CASH"

}

# Only existing real positions are calculated

POST /api/v5/account/position-builder

body

{

"inclRealPosAndEq":true

}

# Only virtual positions are calculated

POST /api/v5/account/position-builder

body

{

"inclRealPosAndEq": false,

"simPos":[

{

"pos":"10",

"instId":"BTC-USDT-SWAP"

},

{

"pos":"10",

"instId":"LTC-USDT-SWAP"

}

]

}

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** |
| --- | --- | --- | --- |
| inclRealPosAndEq | Boolean | No | Whether import existing positions and assets The default is true |
| spotOffsetType | String | No | Spot-derivatives risk offset mode  1: Spot-derivatives (USDT)  2: Spot-derivatives (crypto)  3: Derivatives-only The default is 3 |
| simPos | Array of object | No | List of simulated positions |
| > instId | String | No | Instrument ID, e.g. BTC-USDT-SWAP Applicable to SWAP/FUTURES/OPTION |
| > pos | String | No | Quantity of positions |
| simAsset | Array of object | No | List of simulated assets When inclRealPosAndEq is true, only real assets are considered and virtual assets are ignored |
| > ccy | String | No | Currency, e.g. BTC |
| > amt | String | No | Currency amount |
| greeksType | String | No | Greeks type BS: Black-Scholes Model Greeks PA: Crypto Greeks CASH: Empirical Greeks The default is BS |

Response Example

{

"code": "0",

"data": [

{

"assets": [

{

"availEq": "2.92259509161",

"borrowImr": "0",

"borrowMmr": "0",

"ccy": "BTC",

"spotInUse": "0"

},

{

"availEq": "1",

"borrowImr": "0",

"borrowMmr": "0",

"ccy": "ETH",

"spotInUse": "0"

},

{

"availEq": "-76819.72721896428",

"borrowImr": "9612.484308105535",

"borrowMmr": "1920.4931804741072",

"ccy": "USDT",

"spotInUse": "0"

},

{

"availEq": "100.000001978",

"borrowImr": "0",

"borrowMmr": "0",

"ccy": "OKB",

"spotInUse": "0"

},

{

"availEq": "1.1618286584225905",

"borrowImr": "0",

"borrowMmr": "0",

"ccy": "USDC",

"spotInUse": "0"

}

],

"borrowMmr": "1919.9362374517698",

"derivMmr": "61.63384859899599",

"eq": "50503.83298678435",

"marginRatio": "24.513003004865656",

"riskUnitData": [

{

"delta": "0.010000438961223",

"gamma": "0",

"imr": "79.93948582424999",

"indexUsd": "0.99971",

"mmr": "61.49191217249999",

"mr1": "61.49191217249999",

"mr1FinalResult": {

"pnl": "-61.49191217249999",

"spotShock": "-0.15",

"volShock": "up"

},

"mr1Scenarios": {

"volSame": {

"0": "0",

"-0.05": "-20.497304057499974",

"-0.1": "-40.99460811500002",

"0.1": "40.99460811500002",

"0.15": "61.49191217249999",

"0.05": "20.497304057499974",

"-0.15": "-61.49191217249999"

},

"volShockDown": {

"0": "0",

"-0.05": "-20.497304057499974",

"-0.1": "-40.99460811500002",

"0.1": "40.99460811500002",

"0.15": "61.49191217249999",

"0.05": "20.497304057499974",

"-0.15": "-61.49191217249999"

},

"volShockUp": {

"0": "0",

"-0.05": "-20.497304057499974",

"-0.1": "-40.99460811500002",

"0.1": "40.99460811500002",

"0.15": "61.49191217249999",

"0.05": "20.497304057499974",

"-0.15": "-61.49191217249999"

}

},

"mr2": "0",

"mr3": "0",

"mr4": "0",

"mr5": "0",

"mr6": "61.49191217249999",

"mr6FinalResult": {

"pnl": "-122.98382434499997",

"spotShock": "-0.3"

},

"mr7": "1.8448113495150003",

"portfolios": [

{

"amt": "10",

"delta": "0.010000438961223",

"gamma": "0",

"instId": "BTC-USDT-SWAP",

"instType": "SWAP",

"isRealPos": false,

"notionalUsd": "409.968",

"theta": "0",

"vega": "0"

}

],

"riskUnit": "BTC-USDT",

"theta": "0",

"vega": "0"

},

{

"delta": "0.009998760367833",

"gamma": "0",

"imr": "0.1845173544448",

"indexUsd": "0.99971",

"mmr": "0.141936426496",

"mr1": "0.141936426496",

"mr1FinalResult": {

"pnl": "-0.141936426496",

"spotShock": "-0.2",

"volShock": "volatility\_shock\_up"

},

"mr1Scenarios": {

"volSame": {

"0": "0",

"-0.07": "-0.0496777492736",

"-0.2": "-0.141936426496",

"0.07": "0.0496777492736",

"0.2": "0.141936426496",

"0.14": "0.0993554985472",

"-0.14": "-0.0993554985472"

},

"volShockDown": {

"0": "0",

"-0.07": "-0.0496777492736",

"-0.2": "-0.141936426496",

"0.07": "0.0496777492736",

"0.2": "0.141936426496",

"0.14": "0.0993554985472",

"-0.14": "-0.0993554985472"

},

"volShockUp": {

"0": "0",

"-0.07": "-0.0496777492736",

"-0.2": "-0.141936426496",

"0.07": "0.0496777492736",

"0.2": "0.141936426496",

"0.14": "0.0993554985472",

"-0.14": "-0.0993554985472"

}

},

"mr2": "0",

"mr3": "0",

"mr4": "0",

"mr5": "0",

"mr6": "0.141936426496",

"mr6FinalResult": {

"pnl": "-0.283872852992",

"spotShock": "-0.4"

},

"mr7": "0.004967159106",

"portfolios": [

{

"amt": "10",

"delta": "0.009998760367833",

"gamma": "0",

"instId": "LTC-USDT-SWAP",

"instType": "SWAP",

"isRealPos": false,

"notionalUsd": "0.7098000000000001",

"theta": "0",

"vega": "0"

}

],

"riskUnit": "LTC-USDT",

"theta": "0",

"vega": "0"

}

],

"totalImr": "9689.820690834878",

"totalMmr": "1981.5700860507657",

"ts": "1705915813386"

}

],

"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 margin ratio for the account |
| ts | String | Update time for the account, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| assets | Array of object | Asset info |
| > ccy | String | Currency, e.g. BTC |
| > availEq | String | Currency equity |
| > spotInUse | String | Spot in use |
| > borrowMmr | String | Borrowing MMR (USD) |
| > borrowImr | String | Borrowing IMR (USD) |
| riskUnitData | Array of object | Risk unit info |
| > riskUnit | String | Risk unit, e.g. BTC-USDT |
| > indexUsd | String | Risk unit index price (USD) |
| > mmr | String | Risk unit MMR (USD) |
| > imr | String | Risk unit IMR (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) |
| > mr1Scenarios | Object | MR1 scenario analysis |
| >> volShockDown | Object | When the volatility shock 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", ...} |
| >> volSame | Object | When the volatility keep 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 the volatility shock 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", ...} |
| > mr1FinalResult | Object | 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 |
| > mr6FinalResult | String | 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% |
| > 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 object | Portfolios info |
| >> instId | String | Instrument ID, e.g. BTC-USDT-SWAP |
| >> instType | String | Instrument type SPOT SWAP FUTURES OPTION |
| >> amt | String | When instType is SPOT, it represents spot in use. When instType is SWAP/FUTURES/OPTION, it represents position amount. |
| >> notionalUsd | String | Notional in USD |
| >> delta | String | When instType is SPOT, it represents asset amount. When instType is SWAP/FUTURES/OPTION, 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 instType is SPOT, it will returns "". |
| >> theta | String | The change in contract price each day closer to expiry (by Instrument ID). When instType is SPOT, it will returns "". |
| >> vega | String | The change of the option price when underlying volatility increases by 1% (by Instrument ID). When instType is SPOT, it will returns "". |
| >> isRealPos | Boolean | Whether it is a real position If instType is SWAP/FUTURES/OPTION, it is a valid parameter, else it will returns false |

### 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: UserID

#### HTTP Request

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

Request Example

# Set spot risk offset amount

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

{

"ccy": "BTC",

"clSpotInUseAmt": "0.5"

}

#### 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: UserID

#### HTTP Requests

GET /api/v5/account/greeks

Request Example

# Get the greeks of all assets in the account

GET /api/v5/account/greeks

# Get the greeks of BTC assets in the account

GET /api/v5/account/greeks?ccy=BTC

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. BTC. |

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 OPTION |
| thetaPA | String | theta: Greeks in coins, only applicable to OPTION |
| vegaBS | String | vega: Black-Scholes Greeks in dollars, only applicable to OPTION |
| vegaPA | String | vega：Greeks in coins, only applicable to OPTION |
| 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: UserID

#### HTTP Request

GET /api/v5/account/position-tiers

Request Example

# Query limitation of BTC-USDT

GET /api/v5/account/position-tiers?instType=SWAP&uly=BTC-USDT

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 SWAP FUTURES OPTION |
| uly | String | Conditional | Single underlying or multiple underlyings (no more than 3) separated with comma. Either uly or instFamily is required. If both are passed, instFamily will be used. |
| instFamily | String | Conditional | Single instrument family or instrument families (no more than 5) separated with comma. Either uly or instFamily is required. If both are passed, instFamily will be used. |

Response Example

{

"code": "0",

"data": [

{

"instFamily": "BTC-USDT",

"maxSz": "10000",

"posType": "",

"uly": "BTC-USDT"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| uly | String | Underlying Applicable to FUTURES/SWAP/OPTION |
| instFamily | String | Instrument family Applicable to FUTURES/SWAP/OPTION |
| maxSz | String | Max number of positions |
| posType | String | Limitation of position type, only applicable to cross 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. |

### Set risk offset type

Configure the risk offset type in portfolio margin mode.

#### Rate Limit: 10 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/account/set-riskOffset-type

Request Example

POST /api/v5/account/set-riskOffset-type

body

{

"type":"1"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| type | String | Yes | 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 |

Response Example

{

"code": "0",

"msg": "",

"data": [{

"type": "1"

}]

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| type | String | 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 |

### Activate option

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/account/activate-option

Request Example

POST /api/v5/account/activate-option

#### 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: UserID

#### HTTP Request

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

Request Example

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

body

{

"autoLoan":true,

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| autoLoan | Boolean | No | Whether to automatically make loans  Valid values are true, false  The default is true |

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 Spot and futures mode / Multi-currency margin mode to Portfolio margin mode, you may optionally pre-set riskOffetType. When switching from Portfolio margin mode back to Spot and 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: UserID

#### HTTP Request

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

Request example

// 1. Spot and futures mode -> Multi-currency margin mode

{

"acctLv": "3"

}

// 2. Spot and futures mode/Multi-currency margin mode -> Portfolio margin mode

{

"acctLv": "4",

"riskOffsetType": "1"

}

// 3. Multi-currency margin mode -> Spot and futures mode

{

"acctLv": "2"

}

// 4. Portfolio margin mode -> Spot and futures mode/Multi-currency margin mode, the user have cross-margin contract position and lever is required

{

"acctLv": "2",

"lever": "10"

}

// 5. Portfolio margin mode -> Spot and futures mode/Multi-currency margin mode, the user doesn't have cross-margin contract position and lever is not required

{

"acctLv": "3"

}

#### Request parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| acctLv | String | Yes | Account mode 2: Spot and futures mode 3: Multi-currency margin code 4: Portfolio margin mode |
| lever | String | Optional | Leverage Required when switching from Portfolio margin mode to Spot and futures mode or Multi-currency margin mode, and the user holds cross-margin positions. |
| riskOffsetType | String | Optional | 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 Spot and futures mode or Multi-currency margin mode to Portfolio margin mode. |

Response example

// 1. Spot and futures mode -> Multi-currency margin mode

{

"acctLv": "3",

"curAcctLv": "2",

"lever": "",

"riskOffsetType": ""

}

// 2. Spot and futures mode/Multi-currency margin mode -> Portfolio margin mode

{

"acctLv": "4",

"curAcctLv": "2",

"lever": "",

"riskOffsetType": "1"

}

// 3. Multi-currency margin mode -> Spot and futures mode

{

"acctLv": "2",

"curAcctLv": "3",

"lever": "",

"riskOffsetType": ""

}

// 4. Portfolio margin mode -> Spot and futures mode/Multi-currency margin mode

{

"acctLv": "2",

"curAcctLv": "4",

"lever": "10",

"riskOffsetType": ""

}

// 5. Portfolio margin mode -> Spot and 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 |

lever: When switching from Portfolio margin mode to Spot and 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.   
  
riskOffsetType: Optional when switching from Spot and futures mode or Multi-currency margin mode to Portfolio margin mode. If not provided, it uses the user's current riskOffsetType value as default; if none exists, it uses 3:Derivatives only as default.

### Precheck account mode switch

Retrieve precheck information for account mode switching.

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

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

Request example

GET /api/v5/account/set-account-switch-precheck?acctLv=3

#### Request parameters

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

Response example

// Spot and 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"

}

// Spot and futures mode->Portfolio margin mode, unmatched information

// sCode 1

{

"code": "0",

"data": [

{

"acctLv": "3",

"curAcctLv": "1",

"mgnAft": null,

"mgnBf": null,

"posList": [],

"posTierCheck": [],

"riskOffsetType": "",

"sCode": "1",

"unmatchedInfoCheck": [

{

"posList": [],

"totalAsset": "",

"type": "repay\_borrowings"

}

]

}

],

"msg": ""

}

// 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": ""

}

// 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 | Check code 0: pass all checks 1: unmatched information 3: leverage setting is not finished 4: position tier or margin check is not passed |
| curAcctLv | String | Account mode 1: Spot mode 2: Spot and futures mode 3: Multi-currency margin code 4: Portfolio margin mode Applicable to all scenarios |
| acctLv | String | Account mode 1: Spot mode 2: Spot and futures mode 3: Multi-currency margin code 4: Portfolio margin mode Applicable to all scenarios |
| riskOffsetType | String | 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 acctLv is 4, return "" for other scenarios If the user preset before, it will use the user's specified value; if not, the default value 3 will be applied |
| unmatchedInfoCheck | Array of objects | Unmatched information list Applicable when sCode is 1, indicating there is unmatched information; return [] for other scenarios |
| >> type | String | Unmatched information type asset\_validation: asset validation pending\_orders: order book pending orders pending\_algos: pending algo orders and trading bots, such as iceberg, recurring buy and twap isolated\_margin: isolated margin (quick margin and manual transfers) isolated\_contract: isolated contract (manual transfers) contract\_long\_short: contract positions in hedge mode cross\_margin: cross margin positions cross\_option\_buyer: cross options buyer isolated\_option: isolated options (only applicable to spot mode) growth\_fund: positions with trial funds all\_positions: all positions spot\_lead\_copy\_only\_simple\_single: copy trader and customize lead trader can only use spot mode or spot and futures mode stop\_spot\_custom: spot customize copy trading stop\_futures\_custom: contract customize copy trading lead\_portfolio: lead trader can not switch to portfolio margin mode futures\_smart\_sync: you can not switch to spot mode when having smart contract sync vip\_fixed\_loan: vip loan repay\_borrowings: borrowings compliance\_restriction: due to compliance restrictions, margin trading services are unavailable compliance\_kyc2: Due to compliance restrictions, margin trading services are unavailable. If you are not a resident of this region, please complete kyc2 identity verification. |
| >> totalAsset | String | Total assets Only applicable when type is asset\_validation, return "" for other scenarios |
| >> posList | Array of string | Unmatched position list (posId) Applicable when type is related to positions, return [] for other scenarios |
| posList | Array of objects | Cross margin contract position list Applicable when curAcctLv is 4, acctLv is 2/3 and user has cross margin contract positions Applicable when sCode is 0/3/4 |
| > posId | String | Position ID |
| > lever | String | Leverage of cross margin contract positions after switch |
| posTierCheck | Array of objects | Cross margin contract positions that don't pass the position tier check Only applicable when sCode is 4 |
| > instFamily | String | Instrument family |
| > instType | String | Instrument type SWAP FUTURES OPTION |
| > pos | String | Quantity of position |
| > lever | String | Leverage |
| > maxSz | String | 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. |
| mgnBf | Object | The margin related information before switching account mode Applicable when sCode is 0/4, return null for other scenarios |
| > acctAvailEq | String | Account available equity in USD Applicable when curAcctLv is 3/4, return "" for other scenarios |
| > mgnRatio | String | Margin ratio in USD Applicable when curAcctLv is 3/4, return "" for other scenarios |
| > 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 | 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 | 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 | 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: UserID

#### HTTP Request

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

Request Example

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

body

{

"acctLv":"1"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| acctLv | String | Yes | Account mode 1: Spot mode 2: Spot and 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 |

### 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: UserID

#### HTTP Request

POST /api/v5/account/mmp-reset

Request Example

POST /api/v5/account/mmp-reset

body

{

"instType":"OPTION",

"instFamily":"BTC-USD"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instType | String | No | Instrument type OPTION 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 true, false |

### Set MMP

This endpoint is used to set MMP configure

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 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: UserID

#### HTTP Request

POST /api/v5/account/mmp-config

Request Example

POST /api/v5/account/mmp-config

body

{

"instFamily":"BTC-USD",

"timeInterval":"5000",

"frozenInterval":"2000",

"qtyLimit": "100"

}

#### 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": [

{

"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: UserID

#### HTTP Request

GET /api/v5/account/mmp-config

Request Example

GET /api/v5/account/mmp-config?instFamily=BTC-USD

#### 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 |
| mmpFrozen | Boolean | Whether MMP is currently triggered. true or false |
| 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 mmpFrozenUntil will be "". |
| qtyLimit | String | Trade qty limit in number of contracts |

## 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](https://www.okx.com/docs-v5/en/#overview-websocket-connection-count-limit).

#### URL Path

/ws/v5/private (required login)

Request Example : single

{

"op": "subscribe",

"args": [

{

"channel": "account",

"ccy": "BTC"

}

]

}

Request Example

{

"op": "subscribe",

"args": [

{

"channel": "account",

"extraParams": "

{

\"updateInterval\": \"0\"

}

"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name account |
| > 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

{

"event": "subscribe",

"arg": {

"channel": "account",

"ccy": "BTC"

},

"connId": "a4d3ae55"

}

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "account"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"account\", \"ccy\" : \"BTC\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| 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"

},

"data": [{

"adjEq": "55444.12216906034",

"borrowFroz": "0",

"details": [{

"availBal": "4734.371190691436",

"availEq": "4734.371190691435",

"borrowFroz": "0",

"cashBal": "4750.426970691436",

"ccy": "USDT",

"coinUsdPrice": "0.99927",

"crossLiab": "0",

"disEq": "4889.379316336831",

"eq": "4892.951170691435",

"eqUsd": "4889.379316336831",

"smtSyncEq": "0",

"spotCopyTradingEq": "0",

"fixedBal": "0",

"frozenBal": "158.57998",

"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": "-7.475800000000003",

"uplLiab": "0",

"spotBal": "",

"openAvgPx": "",

"accAvgPx": "",

"spotUpl": "",

"spotUplRatio": "",

"totalPnl": "",

"totalPnlRatio": ""

}],

"imr": "8.5737166146",

"isoEq": "0",

"mgnRatio": "143705.65988369548",

"mmr": "0.342948664584",

"notionalUsd": "85.737166146",

"ordFroz": "0",

"totalEq": "55868.06403501676",

"uTime": "1705564223311",

"upl": "-7.470342666000003"

}]

}

#### Push data parameters

| **Parameters** | **Types** | **Description** |
| --- | --- | --- |
| arg | Object | Successfully subscribed channel |
| > channel | String | Channel name |
| > uid | String | User Identifier |
| data | Array | Subscribed data |
| > uTime | String | The latest time to get account information, millisecond format of Unix timestamp, e.g. 1597026383085 |
| > totalEq | String | The total amount of equity in USD |
| > isoEq | String | Isolated margin equity in USD Applicable to Spot and futures mode/Multi-currency margin/Portfolio margin |
| > adjEq | String | Adjusted / Effective equity in USD  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 Spot mode/Multi-currency margin/Portfolio margin |
| > ordFroz | String | Margin frozen for pending cross orders in USD  Only applicable to Spot mode/Multi-currency margin |
| > imr | String | Initial margin requirement in USD  The sum of initial margins of all open positions and pending orders under cross-margin mode in USD.  Applicable to Spot mode/Multi-currency margin/Portfolio margin |
| > mmr | String | Maintenance margin requirement in USD  The sum of maintenance margins of all open positions and pending orders under cross-margin mode in USD.  Applicable to Spot mode/Multi-currency margin/Portfolio margin |
| > borrowFroz | String | Potential borrowing IMR of the account in USD  Only applicable to Spot mode/Multi-currency margin/Portfolio margin. It is "" for other margin modes. |
| > mgnRatio | String | Margin ratio in USD.  Applicable to Spot mode/Multi-currency margin/Portfolio margin |
| > notionalUsd | String | Notional value of positions in USD  Applicable to Spot mode/Multi-currency margin/Portfolio margin |
| > upl | String | Cross-margin info of unrealized profit and loss at the account level in USD Applicable to Multi-currency margin/Portfolio margin |
| > details | Array | 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. 1597026383085 |
| >> isoEq | String | Isolated margin equity of currency Applicable to Spot and futures mode/Multi-currency margin/Portfolio margin |
| >> availEq | String | Available equity of currency Applicable to Spot and futures mode/Multi-currency margin/Portfolio margin |
| >> disEq | String | Discount equity of currency in USD |
| >> fixedBal | String | Frozen balance for Dip Sniper and Peak Sniper |
| >> availBal | String | Available balance of currency |
| >> frozenBal | String | Frozen balance of currency |
| >> ordFrozen | String | Margin frozen for open orders  Applicable to Spot mode/Spot and futures mode/Multi-currency margin |
| >> liab | String | Liabilities of currency It is a positive value, e.g. 21625.64.  Applicable to Spot mode/Multi-currency margin/Portfolio margin |
| >> upl | String | The sum of the unrealized profit & loss of all margin and derivatives positions of currency.  Applicable to Spot and futures mode/Multi-currency margin/Portfolio margin |
| >> uplLiab | String | Liabilities due to Unrealized loss of currency Applicable to Multi-currency margin/Portfolio margin |
| >> crossLiab | String | Cross Liabilities of currency Applicable to Spot mode/Multi-currency margin/Portfolio margin |
| >> isoLiab | String | Isolated Liabilities of currency Applicable to Multi-currency margin/Portfolio margin |
| >> rewardBal | String | Trial fund balance |
| >> mgnRatio | String | Cross margin ratio of currency  The index for measuring the risk of a certain asset in the account.  Applicable to Spot and futures mode and when there is cross position |
| >> imr | String | Cross initial margin requirement at the currency level Applicable to Spot and futures mode and when there is cross position |
| >> mmr | String | Cross maintenance margin requirement at the currency level Applicable to Spot and futures mode and when there is cross position |
| >> interest | String | Interest of currency It is a positive value, e.g."9.01". Applicable to Spot mode/Multi-currency margin/Portfolio margin |
| >> twap | String | System is forced repayment(TWAP) indicator Divided into multiple levels from 0 to 5, the larger the number, the more likely the auto repayment will be triggered. Applicable to Spot mode/Multi-currency margin/Portfolio margin |
| >> maxLoan | String | Max loan of currency Applicable to cross of Spot mode/Multi-currency margin/Portfolio margin |
| >> eqUsd | String | Equity USD of currency |
| >> borrowFroz | String | Potential borrowing IMR of currency in USD  Only applicable to Spot mode/Multi-currency margin/Portfolio margin. It is "" for other margin modes. |
| >> notionalLever | String | Leverage of currency Applicable to Spot and futures mode |
| >> coinUsdPrice | String | Price index USD of currency |
| >> stgyEq | String | strategy equity |
| >> isoUpl | String | Isolated unrealized profit and loss of currency Applicable to Spot and futures mode/Multi-currency margin/Portfolio margin |
| >> spotInUseAmt | String | Spot in use amount Applicable to Portfolio margin |
| >> clSpotInUseAmt | String | User-defined spot risk offset amount Applicable to Portfolio margin |
| >> maxSpotInUseAmt | String | Max possible spot risk offset amount Applicable to Portfolio margin |
| >> 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. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |
| >> openAvgPx | Array | Spot average cost price. The unit is USD. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |
| >> accAvgPx | Array | Spot accumulated cost price. The unit is USD. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |
| >> spotUpl | String | Spot unrealized profit and loss. The unit is USD. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |
| >> spotUplRatio | String | Spot unrealized profit and loss ratio. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |
| >> totalPnl | String | Spot accumulated profit and loss. The unit is USD. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |
| >> totalPnlRatio | String | Spot accumulated profit and loss ratio. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |

"" 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](https://www.okx.com/docs-v5/en/#overview-websocket-connection-count-limit).

#### URL Path

/ws/v5/private (required login)

Request Example : single

{

"op": "subscribe",

"args": [

{

"channel": "positions",

"instType": "FUTURES",

"instFamily": "BTC-USD"

}

]

}

Request Example

{

"op": "subscribe",

"args": [

{

"channel": "positions",

"instType": "ANY",

"extraParams": "

{

\"updateInterval\": \"0\"

}

"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | 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 | 0: only push due to positions events  2000, 3000, 4000: push by events and regularly according to the time interval setting (ms)   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.   The following format should be strictly followed when using this field.  "extraParams": "  {  \"updateInterval\": \"0\"  } " |

Successful Response Example : single

{

"event": "subscribe",

"arg": {

"channel": "positions",

"instType": "FUTURES",

"instFamily": "BTC-USD"

},

"connId": "a4d3ae55"

}

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "positions",

"instType": "ANY"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"positions\", \"instType\" : \"FUTURES\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name |
| > instType | String | Yes | Instrument type OPTION FUTURES SWAP MARGIN  ANY |
| > 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: single

{

"arg":{

"channel":"positions",

"uid": "77982378738415879",

"instType":"FUTURES"

},

"data":[

{

"adl":"1",

"availPos":"1",

"avgPx":"2566.31",

"cTime":"1619507758793",

"ccy":"ETH",

"deltaBS":"",

"deltaPA":"",

"gammaBS":"",

"gammaPA":"",

"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",

"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 Example

{

"arg": {

"channel": "positions",

"uid": "77982378738415879",

"instType": "ANY"

},

"data": [

{

"adl":"1",

"availPos":"1",

"avgPx":"2566.31",

"cTime":"1619507758793",

"ccy":"ETH",

"deltaBS":"",

"deltaPA":"",

"gammaBS":"",

"gammaPA":"",

"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",

"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":"",

"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": "",

"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",

"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 |
| data | Array | Subscribed data |
| > instType | String | Instrument type |
| > mgnMode | String | Margin mode, cross isolated |
| > posId | String | Position ID |
| > posSide | String | Position side long  short  net (FUTURES/SWAP/OPTION: positive pos means long position and negative pos means short position. MARGIN: posCcy being base currency means long position, posCcy 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 0 will be generated after the deposit is transferred |
| > baseBal | String | (Deprecated) |
| > quoteBal | String | (Deprecated) |
| > baseBorrowed | String | (Deprecated) |
| > baseInterest | String | (Deprecated) |
| > quoteBorrowed | String | (Deprecated) |
| > quoteInterest | String | (Deprecated) |
| > posCcy | String | Position currency, only applicable to MARGIN positions |
| > availPos | String | Position that can be closed  Only applicable to MARGIN and OPTION.  For Margin position, the rest of sz will be SPOT 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 SPOT 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. BTC-USDT-SWAP |
| > lever | String | Leverage, not applicable to OPTION seller |
| > liqPx | String | Estimated liquidation price  Not applicable to OPTION |
| > markPx | String | Latest Mark price |
| > imr | String | Initial margin requirement, only applicable to cross |
| > margin | String | Margin, can be added or reduced. Only applicable to isolated Margin. |
| > mgnRatio | String | Margin ratio |
| > mmr | String | Maintenance margin requirement |
| > liab | String | Liabilities, only applicable to MARGIN. |
| > liabCcy | String | Liabilities currency, only applicable to MARGIN. |
| > interest | String | Interest accrued that has not been settled. |
| > tradeId | String | Last trade ID |
| > notionalUsd | String | Notional value of positions in USD |
| > optVal | String | Option Value, only applicable to OPTION. |
| > pendingCloseOrdLiabVal | String | The amount of close orders of isolated margin liability. |
| > adl | String | Auto decrease line, signal area Divided into 5 levels, from 1 to 5, the smaller the number, the weaker the adl intensity. |
| > 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 ccy on the market, only applicable to OPTION |
| > bePx | String | Breakeven price |
| > deltaBS | String | delta: Black-Scholes Greeks in dollars, only applicable to OPTION |
| > deltaPA | String | delta: Greeks in coins, only applicable to OPTION |
| > 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 OPTION |
| > thetaPA | String | theta: Greeks in coins, only applicable to OPTION |
| > vegaBS | String | vega: Black-Scholes Greeks in dollars, only applicable to OPTION |
| > vegaPA | String | vega: Greeks in coins, only applicable to OPTION |
| > spotInUseAmt | String | Spot in use amount Applicable to Portfolio margin |
| > spotInUseCcy | String | Spot in use unit, e.g. BTC Applicable to Portfolio margin |
| > clSpotInUseAmt | String | User-defined spot risk offset amount Applicable to Portfolio margin |
| > maxSpotInUseAmt | String | Max possible spot risk offset amount Applicable to Portfolio margin |
| > realizedPnl | String | Realized profit and loss Only applicable to FUTURES/SWAP/OPTION realizedPnl=pnl+fee+fundingFee+liqPenalty |
| > pnl | String | Accumulated pnl of closing order(s) |
| > 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 | Close position algo orders attached to the position. This array will have values only after you request "Place algo order" with closeFraction=1. |
| >> algoId | String | Algo ID |
| >> slTriggerPx | String | Stop-loss trigger price. |
| >> slTriggerPxType | String | Stop-loss trigger price type.  last: last price index: index price mark: mark price |
| >> tpTriggerPx | String | Take-profit trigger price. |
| >> tpTriggerPxType | String | Take-profit trigger price type.  last: last price index: index price mark: 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. 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. |

- 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](https://www.okx.com/docs-v5/en/#overview-websocket-connection-count-limit).

#### URL Path

/ws/v5/private (required login)

Request Example

{

"op": "subscribe",

"args": [{

"channel": "balance\_and\_position"

}]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name balance\_and\_position |

Response Example

{

"event": "subscribe",

"arg": {

"channel": "balance\_and\_position"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"balance\_and\_position\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Operation subscribe unsubscribe error |
| arg | Object | No | List of subscribed channels |
| > channel | String | Yes | Channel name balance\_and\_position |
| 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",

"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 | Subscribed data |
| > pTime | String | Push time of both balance and position information, millisecond format of Unix timestamp, e.g. 1597026383085 |
| > eventType | String | Event Type snapshot,delivered,exercised,transferred,filled,liquidation,claw\_back,adl,funding\_fee,adjust\_margin,set\_leverage,interest\_deduction |
| > balData | String | Balance data |
| >> ccy | String | Currency |
| >> cashBal | String | Cash Balance |
| >> uTime | String | Update time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| > posData | String | Position data |
| >> posId | String | Position ID |
| >> tradeId | String | Last trade ID |
| >> instId | String | Instrument ID, e.g BTC-USD-180213 |
| >> instType | String | Instrument type |
| >> mgnMode | String | Margin mode isolated, cross |
| >> avgPx | String | Average open price |
| >> ccy | String | Currency used for margin |
| >> posSide | String | Position side long, short, net |
| >> pos | String | Quantity of positions. In the isolated margin mode, when doing manual transfers, a position with pos of 0 will be generated after the deposit is transferred |
| >> baseBal | String | (Deprecated) |
| >> quoteBal | String | (Deprecated) |
| >> posCcy | String | Position currency, only applicable to MARGIN positions. |
| >> uTime | String | Update time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| > trades | Array | Details of trade |
| >> instId | String | Instrument ID, e.g. BTC-USDT |
| >> 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](https://www.okx.com/docs-v5/en/#overview-websocket-connection-count-limit).

#### URL Path

/ws/v5/private (required login)

Request Example

{

"op": "subscribe",

"args": [

{

"channel": "liquidation-warning",

"instType": "ANY"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name liquidation-warning |
| > 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 |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "liquidation-warning",

"instType": "ANY"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"liquidation-warning\", \"instType\" : \"FUTURES\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name |
| > instType | String | Yes | Instrument type OPTION FUTURES SWAP MARGIN  ANY |
| > 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 | Subscribed data |
| > instType | String | Instrument type |
| > mgnMode | String | Margin mode, cross isolated |
| > posId | String | Position ID |
| > posSide | String | Position side long  short  net (FUTURES/SWAP/OPTION: positive pos means long position and negative pos means short position. MARGIN: posCcy being base currency means long position, posCcy being quote currency means short position.) |
| > pos | String | Quantity of positions |
| > posCcy | String | Position currency, only applicable to MARGIN positions |
| > instId | String | Instrument ID, e.g. BTC-USDT-SWAP |
| > lever | String | Leverage, not applicable to OPTION seller |
| > markPx | String | Mark price |
| > mgnRatio | String | 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](https://www.okx.com/docs-v5/en/#overview-websocket-connection-count-limit).

#### URL Path

/ws/v5/private (required login)

Request Example : single

{

"op": "subscribe",

"args": [{

"channel": "account-greeks",

"ccy": "BTC"

}]

}

Request Example

{

"op": "subscribe",

"args": [{

"channel": "account-greeks"

}]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name account-greeks |
| > ccy | String | No | Currency |

Successful Response Example : single

{

"event": "subscribe",

"arg": {

"channel": "account-greeks",

"ccy": "BTC"

},

"connId": "a4d3ae55"

}

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "account-greeks"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"account-greeks\", \"ccy\" : \"BTC\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Operation subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name,account-greeks |
| > ccy | String | No | 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",

"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": "-0.0000000000230396",

"gammaPA": "0.1376693483440320",

"thetaBS": "0.3314776517141782",

"thetaPA": "0.0000000001316008",

"ts": "1729179001692",

"vegaBS": "-0.0000000045069794",

"vegaPA": "-0.0000000000017267"

}

]

}

#### Push data parameters

| **Parameters** | **Types** | **Description** |
| --- | --- | --- |
| arg | Object | Successfully subscribed channel |
| > channel | String | Channel name |
| > uid | String | User Identifier |
| data | Array | 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 |
| > 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.   
- 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 instruments for Copy Trading: 4 requests per 2 seconds

#### Rate limit rule (except Options): UserID + Instrument ID

#### Rate limit rule (Options only): UserID + Instrument Family

Rate limit of this endpoint will also be affected by the rules [Sub-account rate limit](https://www.okx.com/docs-v5/en/#overview-rate-limits-sub-account-rate-limit) and [Fill ratio based sub-account rate limit](https://www.okx.com/docs-v5/en/#overview-rate-limits-fill-ratio-based-sub-account-rate-limit).

#### HTTP Request

POST /api/v5/trade/order

Request Example

place order for SPOT

POST /api/v5/trade/order

body

{

"instId":"BTC-USDT",

"tdMode":"cash",

"clOrdId":"b15",

"side":"buy",

"ordType":"limit",

"px":"2.15",

"sz":"2"

}

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. BTC-USDT |
| tdMode | String | Yes | Trade mode Margin mode cross isolated Non-Margin mode cash spot\_isolated (only applicable to SPOT lead trading, tdMode should be spot\_isolated for SPOT lead trading.) |
| ccy | String | No | Margin currency  Only applicable to cross MARGIN orders in Spot and futures mode. |
| 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. Only applicable to general order. It will not be posted to algoId when placing TP/SL order after the general order is filled completely. |
| 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 |
| posSide | String | Conditional | Position side  The default is net in the net mode  It is required in the long/short mode, and can only be long or short.  Only applicable to FUTURES/SWAP. |
| ordType | String | Yes | 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 (applicable only to Expiry Futures and Perpetual Futures). 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) |
| sz | String | Yes | Quantity to buy or sell |
| px | String | Conditional | Order price. Only applicable to limit,post\_only,fok,ioc,mmp,mmp\_and\_post\_only 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 USD  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: true or false. The default value is false. Only applicable to MARGIN orders, and FUTURES/SWAP orders in net mode  Only applicable to Spot and futures mode and Multi-currency margin |
| tgtCcy | String | No | Whether the target currency uses the quote or base currency. base\_ccy: Base currency ,quote\_ccy: Quote currency  Only applicable to SPOT Market Orders Default is quote\_ccy for buy, base\_ccy for sell |
| banAmend | Boolean | No | Whether to disallow the system from amending the size of the SPOT Market Order. Valid options: true or false. The default value is false. If true, system will not amend and reject the market order if user does not have sufficient funds.  Only applicable to SPOT Market Orders |
| quickMgnType | String | No | (Deprecated) |
| stpId | String | No | (deprecated) |
| stpMode | String | No | Self trade prevention mode.  Default to cancel maker  cancel\_maker,cancel\_taker, cancel\_both Cancel both does not support FOK. |
| attachAlgoOrds | Array of object | 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 algoClOrdId 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. |
| > 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 condition limit The default is condition |
| > slTriggerPx | String | Conditional | Stop-loss trigger price If you fill in this parameter, you should fill in the stop-loss order price. |
| > 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 last: last price  index: index price  mark: mark price  The default is last |
| > slTriggerPxType | String | No | Stop-loss trigger price type last: last price  index: index price  mark: mark price  The default is last |
| > 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 slTriggerPx will move to avgPx when the first TP order is triggered 0: disable, the default value  1: Enable |

Response Example

{

"code": "0",

"msg": "",

"data": [

{

"clOrdId": "oktswap6",

"ordId": "312269865356374016",

"tag": "",

"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 |
| > 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. 1597026383085123  The time is recorded after authentication. |
| outTime | String | Timestamp at REST gateway when the response is sent, Unix timestamp format in microseconds, e.g. 1597026383085123 |

tdMode  
Trade Mode, when placing an order, you need to specify the trade mode.  
**Spot mode:**  
- SPOT and OPTION buyer: cash  
**Spot and futures mode:**  
- Isolated MARGIN: isolated  
- Cross MARGIN: cross  
- SPOT: cash  
- Cross FUTURES/SWAP/OPTION: cross  
- Isolated FUTURES/SWAP/OPTION: isolated  
**Multi-currency margin mode:**  
- 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 `Spot and 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. Lead trader can‘t use attachAlgoOrds (Array), advice to user previous method, more details can refer to https://www.okx.com/docs-v5/log\_en/#2022-11-21   
3. Attaching TP/SL is neither supported for market buy with tgtCcy is base\_ccy or market sell with tgtCcy is quote\_ccy  
4. If tpOrdKind is limit, and there is only one conditional TP order, attachAlgoClOrdId can be used clOrdId for retrieving on "GET / Order details" endpoint. 5. 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 not 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 default STP mode is `Cancel Maker`. 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.

### 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 instruments for Copy Trading: 4 orders per 2 seconds

#### Rate limit rule (except Options): UserID + Instrument ID

#### Rate limit rule (Options only): UserID + Instrument Family

Rate limit of this endpoint will also be affected by the rules [Sub-account rate limit](https://www.okx.com/docs-v5/en/#overview-rate-limits-sub-account-rate-limit) and [Fill ratio based sub-account rate limit](https://www.okx.com/docs-v5/en/#overview-rate-limits-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

batch place order for SPOT

POST /api/v5/trade/batch-orders

body

[

{

"instId":"BTC-USDT",

"tdMode":"cash",

"clOrdId":"b15",

"side":"buy",

"ordType":"limit",

"px":"2.15",

"sz":"2"

},

{

"instId":"BTC-USDT",

"tdMode":"cash",

"clOrdId":"b16",

"side":"buy",

"ordType":"limit",

"px":"2.15",

"sz":"2"

}

]

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\_clOrdId = [

{"instId": "BTC-USDT", "tdMode": "cash", "clOrdId": "b15", "side": "buy", "ordType": "limit", "px": "2.15", "sz": "2"},

{"instId": "BTC-USDT", "tdMode": "cash", "clOrdId": "b16", "side": "buy", "ordType": "limit", "px": "2.15", "sz": "2"}

]

result = tradeAPI.place\_multiple\_orders(place\_orders\_without\_clOrdId)

print(result)

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instId | String | Yes | Instrument ID, e.g. BTC-USDT |
| tdMode | String | Yes | Trade mode Margin mode cross isolated Non-Margin mode cash spot\_isolated (only applicable to SPOT lead trading, tdMode should be spot\_isolated for SPOT lead trading.) |
| ccy | String | No | Margin currency  Only applicable to cross MARGIN orders in Spot and futures mode. |
| 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 |
| posSide | String | Conditional | Position side  The default is net in the net mode  It is required in the long/short mode, and can only be long or short.  Only applicable to FUTURES/SWAP. |
| ordType | String | Yes | 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 (applicable only to Expiry Futures and Perpetual Futures). 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) |
| sz | String | Yes | Quantity to buy or sell |
| px | String | Conditional | Order price. Only applicable to limit,post\_only,fok,ioc,mmp,mmp\_and\_post\_only 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 USD  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 position size.  Valid options: true or false. The default value is false. Only applicable to MARGIN orders, and FUTURES/SWAP orders in net mode  Only applicable to Spot and futures mode and Multi-currency margin |
| tgtCcy | String | No | 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 |
| banAmend | Boolean | No | Whether to disallow the system from amending the size of the SPOT Market Order. Valid options: true or false. The default value is false. If true, system will not amend and reject the market order if user does not have sufficient funds.  Only applicable to SPOT Market Orders |
| quickMgnType | String | No | (Deprecated) |
| stpId | String | No | (deprecated) |
| stpMode | String | No | Self trade prevention mode.  Default to cancel maker  cancel\_maker,cancel\_taker, cancel\_both Cancel both does not support FOK. |
| attachAlgoOrds | Array of object | 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 algoClOrdId 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. |
| > 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 condition limit The default is condition |
| > slTriggerPx | String | Conditional | Stop-loss trigger price If you fill in this parameter, you should fill in the stop-loss order price. |
| > 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 last: last price  index: index price  mark: mark price  The default is last |
| > slTriggerPxType | String | No | Stop-loss trigger price type last: last price  index: index price  mark: mark price  The default is last |
| > 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 slTriggerPx will move to avgPx when the first TP order is triggered 0: disable, the default value  1: Enable |

Response Example

{

"code":"0",

"msg":"",

"data":[

{

"clOrdId":"oktswap6",

"ordId":"12345689",

"tag":"",

"ts":"1695190491421",

"sCode":"0",

"sMsg":""

},

{

"clOrdId":"oktswap7",

"ordId":"12344",

"tag":"",

"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 |
| > 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. 1597026383085123  The time is recorded after authentication. |
| outTime | String | Timestamp at REST 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.

### POST / Cancel order

Cancel an incomplete order.

#### Rate Limit: 60 requests per 2 seconds

#### Rate limit rule (except Options): UserID + Instrument ID

#### Rate limit rule (Options only): UserID + Instrument Family

#### HTTP Request

POST /api/v5/trade/cancel-order

Request Example

POST /api/v5/trade/cancel-order

body

{

"ordId":"590908157585625111",

"instId":"BTC-USD-190927"

}

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. BTC-USDT |
| 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 |

Response Example

{

"code":"0",

"msg":"",

"data":[

{

"clOrdId":"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. 1597026383085123  The time is recorded after authentication. |
| outTime | String | Timestamp at REST 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 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): UserID + Instrument ID

#### Rate limit rule (Options only): UserID + 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`.

#### HTTP Request

POST /api/v5/trade/cancel-batch-orders

Request Example

POST /api/v5/trade/cancel-batch-orders

body

[

{

"instId":"BTC-USDT",

"ordId":"590908157585625111"

},

{

"instId":"BTC-USDT",

"ordId":"590908544950571222"

}

]

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. BTC-USDT |
| 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 |

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, 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. 1597026383085123  The time is recorded after authentication. |
| outTime | String | Timestamp at REST gateway when the response is sent, Unix timestamp format in microseconds, e.g. 1597026383085123 |

### POST / Amend order

Amend an incomplete order.

#### Rate Limit: 60 requests per 2 seconds

#### Rate Limit of lead instruments for Copy Trading: 4 requests per 2 seconds

#### Rate limit rule (except Options): UserID + Instrument ID

#### Rate limit rule (Options only): UserID + Instrument Family

Rate limit of this endpoint will also be affected by the rules [Sub-account rate limit](https://www.okx.com/docs-v5/en/#overview-rate-limits-sub-account-rate-limit) and [Fill ratio based sub-account rate limit](https://www.okx.com/docs-v5/en/#overview-rate-limits-fill-ratio-based-sub-account-rate-limit).

#### HTTP Request

POST /api/v5/trade/amend-order

Request Example

POST /api/v5/trade/amend-order

body

{

"ordId":"590909145319051111",

"newSz":"2",

"instId":"BTC-USDT"

}

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 |
| cxlOnFail | Boolean | No | Whether the order needs to be automatically canceled when the order amendment fails  Valid options: false or 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. |
| attachAlgoOrds | Array of object | No | TP/SL information attached when placing order |
| > attachAlgoId | String | Conditional | The order ID of attached TP/SL order. It can be used to identity 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. |
| > 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. |
| > newTpOrdKind | String | No | TP order kind condition limit |
| > 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 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 |
| > newSlTriggerPxType | 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":[

{

"clOrdId":"",

"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. 1597026383085123  The time is recorded after authentication. |
| outTime | String | Timestamp at REST 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

### 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 instruments for Copy Trading: 4 orders per 2 seconds

#### Rate limit rule (except Options): UserID + Instrument ID

#### Rate limit rule (Options only): UserID + Instrument Family

Rate limit of this endpoint will also be affected by the rules [Sub-account rate limit](https://www.okx.com/docs-v5/en/#overview-rate-limits-sub-account-rate-limit) and [Fill ratio based sub-account rate limit](https://www.okx.com/docs-v5/en/#overview-rate-limits-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

POST /api/v5/trade/amend-batch-orders

body

[

{

"ordId":"590909308792049444",

"newSz":"2",

"instId":"BTC-USDT"

},

{

"ordId":"590909308792049555",

"newSz":"2",

"instId":"BTC-USDT"

}

]

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 ordId

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 clOrdIdis 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. |
| attachAlgoOrds | Array of object | No | TP/SL information attached when placing order |
| > attachAlgoId | String | Conditional | The order ID of attached TP/SL order. It can be used to identity 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. |
| > 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. |
| > newTpOrdKind | String | No | TP order kind condition limit |
| > 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 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 |
| > newSlTriggerPxType | 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":[

{

"clOrdId":"oktswap6",

"ordId":"12345689",

"ts":"1695190491421",

"reqId":"b12344",

"sCode":"0",

"sMsg":""

},

{

"clOrdId":"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. 1597026383085123  The time is recorded after authentication. |
| outTime | String | Timestamp at REST 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.

### 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): UserID + Instrument ID

#### Rate limit rule (Options only): UserID + Instrument Family

#### HTTP Request

POST /api/v5/trade/close-position

Request Example

POST /api/v5/trade/close-position

body

{

"instId":"BTC-USDT-SWAP",

"mgnMode":"cross"

}

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 net mode, and the default value is net. You can only fill with net. This parameter must be filled in under the long/short mode. Fill in long for close-long and short for close-short. |
| mgnMode | String | Yes | Margin mode cross isolated |
| ccy | String | Conditional | Margin currency, required in the case of closing cross MARGIN position for Spot and futures mode. |
| autoCxl | Boolean | No | Whether any pending orders for closing out needs to be automatically canceled when close position via a market order. false or true, the default is false. |
| 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): UserID + Instrument ID

#### Rate limit rule (Options only): UserID + Instrument Family

#### HTTP Request

GET /api/v5/trade/order

Request Example

GET /api/v5/trade/order?ordId=1753197687182819328&instId=BTC-USDT

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. BTC-USDT Only applicable to live instruments |
| 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 If the clOrdId is associated with multiple orders, only the latest one will be returned. |

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",

"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",

"uTime": "1708587373362"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| instType | String | Instrument type SPOT MARGIN SWAP FUTURES OPTION |
| instId | String | Instrument ID |
| 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 |
| ccy | String | Margin currency  Only applicable to cross MARGIN orders in Spot and futures mode. |
| 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  px: Place an order based on price, in the unit of coin (the unit for the request parameter px is BTC or ETH)  pxVol: Place an order based on pxVol  pxUsd: 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, Applicable to orders which have a trade and aim to close position. It always is 0 in other conditions |
| ordType | String | 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) |
| side | String | Order side |
| posSide | String | Position side |
| tdMode | String | Trade mode |
| accFillSz | String | Accumulated fill quantity The unit is base\_ccy for SPOT and MARGIN, e.g. BTC-USDT, the unit is BTC; For market orders, the unit both is base\_ccy when the tgtCcy is base\_ccy or quote\_ccy; The unit is contract for FUTURES/SWAP/OPTION |
| fillPx | String | Last filled price. If none is filled, it will return "". |
| tradeId | String | Last traded ID |
| fillSz | String | Last filled quantity The unit is base\_ccy for SPOT and MARGIN, e.g. BTC-USDT, the unit is BTC; For market orders, the unit both is base\_ccy when the tgtCcy is base\_ccy or quote\_ccy; The unit is contract for FUTURES/SWAP/OPTION |
| fillTime | String | Last filled time |
| avgPx | String | Average filled price. If none is filled, it will return "". |
| state | String | State  canceled live  partially\_filled filled mmp\_canceled |
| stpId | String | (deprecated) |
| stpMode | String | Self trade prevention mode  Return "" if self trade prevention is not applicable |
| 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 object | TP/SL information attached when placing order |
| > attachAlgoId | String | The order ID of attached TP/SL order. It can be used to identity 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 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. |
| > 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. |
| > 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. |
| > algoId | Object | Algo ID |
| feeCcy | String | Fee currency |
| fee | String | Fee and rebate For spot and margin, it is accumulated fee charged by the platform. It is always negative, e.g. -0.01.  For Expiry Futures, Perpetual Futures and Options, it is accumulated fee and rebate |
| rebateCcy | String | Rebate currency |
| source | String | Order source 6: The normal order triggered by the trigger order 7:The normal order triggered by the TP/SL order  13: The normal order triggered by the algo order 25:The normal order triggered by the trailing stop order 34: The normal order triggered by the chase order |
| rebate | String | Rebate amount, only applicable to spot and margin, the reward of placing orders from the platform (rebate) given to user who has reached the specified trading level. If there is no rebate, this field is "". |
| category | String | Category normal twap  adl full\_liquidation partial\_liquidation delivery  ddh: Delta dynamic hedge |
| 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. |
| cancelSourceReason | String | Reason for the cancellation. |
| quickMgnType | String | Quick Margin type, Only applicable to Quick Margin Mode of isolated margin manual, auto\_borrow, auto\_repay |
| algoClOrdId | String | Client-supplied Algo ID. There will be a value when algo order attaching algoClOrdId is triggered, or it will be "". |
| algoId | 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. 1597026383085 |
| cTime | String | Creation time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### GET / Order List

Retrieve all incomplete orders under the current account.

#### Rate Limit: 60 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/trade/orders-pending

Request Example

GET /api/v5/trade/orders-pending?ordType=post\_only,fok,ioc&instType=SPOT

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 SPOT MARGIN SWAP FUTURES OPTION |
| uly | String | No | Underlying Applicable to FUTURES/SWAP/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) |
| state | String | No | State live  partially\_filled |
| after | String | No | Pagination of data to return records earlier than the requested ordId |
| before | String | No | Pagination of data to return records newer than the requested ordId |
| 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": "",

"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": "",

"uTime": "1724733617998"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| instType | String | Instrument type |
| instId | String | Instrument ID |
| 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 |
| ccy | String | Margin currency  Only applicable to cross MARGIN orders in Spot and futures mode. |
| 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  px: Place an order based on price, in the unit of coin (the unit for the request parameter px is BTC or ETH)  pxVol: Place an order based on pxVol  pxUsd: 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, Applicable to orders which have a trade and aim to close position. It always is 0 in other conditions |
| ordType | String | 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) |
| side | String | Order side |
| posSide | String | Position side |
| tdMode | String | Trade mode |
| accFillSz | String | Accumulated fill quantity |
| fillPx | String | Last filled price |
| tradeId | 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 object | TP/SL information attached when placing order |
| > attachAlgoId | String | The order ID of attached TP/SL order. It can be used to identity 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 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. |
| > 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. |
| > 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 |
| > 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 | Object | Algo ID |
| stpId | String | (deprecated) |
| stpMode | String | Self trade prevention mode  Return "" if self trade prevention is not applicable |
| feeCcy | String | Fee currency |
| fee | String | Fee and rebate For spot and margin, it is accumulated fee charged by the platform. It is always negative, e.g. -0.01.  For Expiry Futures, Perpetual Futures and Options, it is accumulated fee and rebate |
| rebateCcy | String | Rebate currency |
| source | String | Order source 6: The normal order triggered by the trigger order 7:The normal order triggered by the TP/SL order  13: The normal order triggered by the algo order 25:The normal order triggered by the trailing stop order 34: The normal order triggered by the chase order |
| rebate | String | Rebate amount, only applicable to spot and margin, the reward of placing orders from the platform (rebate) given to user who has reached the specified trading level. If there is no rebate, this field is "". |
| category | String | Category  normal |
| 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 |
| algoClOrdId | String | Client-supplied Algo ID. There will be a value when algo order attaching algoClOrdId 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. 1597026383085 |
| cTime | String | Creation time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| cancelSource | String | Code of the cancellation source. |
| cancelSourceReason | String | Reason for the cancellation. |

### 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: UserID

#### HTTP Request

GET /api/v5/trade/orders-history

Request Example

GET /api/v5/trade/orders-history?ordType=post\_only,fok,ioc&instType=SPOT

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 | Instrument type SPOT MARGIN SWAP FUTURES OPTION |
| uly | String | No | Underlying Applicable to FUTURES/SWAP/OPTION |
| instFamily | String | No | Instrument family Applicable to FUTURES/SWAP/OPTION |
| instId | String | No | Instrument ID, e.g. BTC-USDT |
| 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) |
| state | String | No | State canceled filled mmp\_canceled: Order canceled automatically due to Market Maker Protection |
| category | String | No | Category  twap  adl full\_liquidation partial\_liquidation  delivery  ddh: Delta dynamic hedge |
| after | String | No | Pagination of data to return records earlier than the requested ordId |
| before | String | No | Pagination of data to return records newer than the requested ordId |
| begin | String | No | Filter with a begin timestamp cTime. Unix timestamp format in milliseconds, e.g. 1597026383085 |
| end | String | No | Filter with an end timestamp cTime. 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": [

{

"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",

"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 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 |
| ccy | String | Margin currency  Only applicable to cross MARGIN orders in Spot and futures mode. |
| 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  px: Place an order based on price, in the unit of coin (the unit for the request parameter px is BTC or ETH)  pxVol: Place an order based on pxVol  pxUsd: 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  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) |
| 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 "". |
| tradeId | 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 object | TP/SL information attached when placing order |
| > attachAlgoId | String | The order ID of attached TP/SL order. It can be used to identity 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 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. |
| > 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. |
| > 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 |
| > 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 | Object | Algo ID |
| stpId | String | (deprecated) |
| stpMode | String | Self trade prevention mode  Return "" if self trade prevention is not applicable |
| feeCcy | String | Fee currency |
| fee | String | Fee and rebate For spot and margin, it is accumulated fee charged by the platform. It is always negative, e.g. -0.01.  For Expiry Futures, Perpetual Futures and Options, it is accumulated fee and rebate |
| rebateCcy | String | Rebate currency |
| source | String | Order source 6: The normal order triggered by the trigger order 7:The normal order triggered by the TP/SL order  13: The normal order triggered by the algo order 25:The normal order triggered by the trailing stop order 34: The normal order triggered by the chase order |
| rebate | String | Rebate amount, only applicable to spot and margin, the reward of placing orders from the platform (rebate) given to user who has reached the specified trading level. If there is no rebate, this field is "". |
| pnl | String | Profit and loss, Applicable to orders which have a trade and aim to close position. It always is 0 in other conditions |
| category | String | Category  normal twap  adl full\_liquidation partial\_liquidation  delivery  ddh: Delta dynamic hedge |
| 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 algoClOrdId 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. 1597026383085 |
| cTime | String | Creation time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### 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: UserID

#### HTTP Request

GET /api/v5/trade/orders-history-archive

Request Example

GET /api/v5/trade/orders-history-archive?ordType=post\_only,fok,ioc&instType=SPOT

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 |
| uly | String | No | Underlying Applicable to FUTURES/SWAP/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) |
| state | String | No | State canceled  filled mmp\_canceled: Order canceled automatically due to Market Maker Protection |
| category | String | No | Category  twap  adl full\_liquidation partial\_liquidation delivery  ddh: Delta dynamic hedge |
| after | String | No | Pagination of data to return records earlier than the requested ordId |
| before | String | No | Pagination of data to return records newer than the requested ordId |
| begin | String | No | Filter with a begin timestamp cTime. Unix timestamp format in milliseconds, e.g. 1597026383085 |
| end | String | No | Filter with an end timestamp cTime. 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": [

{

"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": "",

"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",

"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 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 |
| ccy | String | Margin currency  Only applicable to cross MARGIN orders in Spot and futures mode. |
| 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  px: Place an order based on price, in the unit of coin (the unit for the request parameter px is BTC or ETH)  pxVol: Place an order based on pxVol  pxUsd: 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  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) |
| 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 "". |
| tradeId | 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 object | TP/SL information attached when placing order |
| > attachAlgoId | String | The order ID of attached TP/SL order. It can be used to identity 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 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. |
| > 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. |
| > 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 |
| > 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 | Object | Algo ID |
| stpId | String | (deprecated) |
| stpMode | String | Self trade prevention mode  Return "" if self trade prevention is not applicable |
| feeCcy | String | Fee currency |
| fee | String | Fee and rebate For spot and margin, it is accumulated fee charged by the platform. It is always negative, e.g. -0.01.  For Expiry Futures, Perpetual Futures and Options, it is accumulated fee and rebate |
| source | String | Order source 6: The normal order triggered by the trigger order 7:The normal order triggered by the TP/SL order  13: The normal order triggered by the algo order 25:The normal order triggered by the trailing stop order 34: The normal order triggered by the chase order |
| rebateCcy | String | Rebate currency |
| rebate | String | Rebate amount, only applicable to spot and margin, the reward of placing orders from the platform (rebate) given to user who has reached the specified trading level. If there is no rebate, this field is "". |
| pnl | String | Profit and loss, Applicable to orders which have a trade and aim to close position. It always is 0 in other conditions |
| category | String | Category  normal twap  adl full\_liquidation partial\_liquidation  delivery  ddh: Delta dynamic hedge |
| 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 algoClOrdId 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. 1597026383085 |
| cTime | String | Creation time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

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: UserID

#### HTTP Request

GET /api/v5/trade/fills

Request Example

GET /api/v5/trade/fills

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 SPOT MARGIN SWAP FUTURES OPTION |
| uly | String | No | Underlying Applicable to FUTURES/SWAP/OPTION |
| instFamily | String | No | Instrument family Applicable to FUTURES/SWAP/OPTION |
| instId | String | No | Instrument ID, e.g. BTC-USDT |
| ordId | String | No | Order ID |
| subType | String | No | Transaction type  1: Buy 2: Sell 3: Open long 4: Open short 5: Close long 6: Close short  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  110: Liquidation transfer in 111: Liquidation transfer out  118: System token conversion transfer in 119: System token conversion transfer out  125: ADL close long 126: ADL close short 127: ADL buy 128: ADL sell  212: Auto borrow of quick margin 213: Auto repay of quick margin  204: block trade buy 205: block trade sell 206: block trade open long 207: block trade open short 208: block trade close open 209: block trade close short  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 |
| after | String | No | Pagination of data to return records earlier than the requested billId |
| before | String | No | Pagination of data to return records newer than the requested billId |
| begin | String | No | Filter with a begin timestamp ts. Unix timestamp format in milliseconds, e.g. 1597026383085 |
| end | String | No | Filter with an end timestamp ts. 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": [

{

"side": "buy",

"fillSz": "0.00192834",

"fillPx": "51858",

"fillPxVol": "",

"fillFwdPx": "",

"fee": "-0.00000192834",

"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"

}

],

"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. 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 FUTURES, SWAP, OPTION |
| side | String | Order side, buy sell |
| posSide | String | Position side  long short  it returns net innet mode. |
| execType | String | Liquidity taker or maker T: taker M: 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. 1597026383085. |
| fillTime | String | Trade time which is the same as fillTime for the order channel. |
| feeRate | String | Fee rate. This field is returned for SPOT and MARGIN only |

tradeId  
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.  
clOrdId  
Client-supplied order ID, always "" for block trading.

### GET / Transaction details (last 3 months)

Retrieve recently-filled transaction details in the last 3 months.

#### Rate Limit: 10 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/trade/fills-history

Request Example

GET /api/v5/trade/fills-history?instType=SPOT

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 | Instrument type SPOT MARGIN SWAP FUTURES OPTION |
| uly | String | No | Underlying Applicable to FUTURES/SWAP/OPTION |
| instFamily | String | No | Instrument family Applicable to FUTURES/SWAP/OPTION |
| instId | String | No | Instrument ID, e.g. BTC-USDT |
| ordId | String | No | Order ID |
| subType | String | No | Transaction type  1: Buy 2: Sell 3: Open long 4: Open short 5: Close long 6: Close short  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  110: Liquidation transfer in 111: Liquidation transfer out  118: System token conversion transfer in 119: System token conversion transfer out  125: ADL close long 126: ADL close short 127: ADL buy 128: ADL sell  212: Auto borrow of quick margin 213: Auto repay of quick margin  204: block trade buy 205: block trade sell 206: block trade open long 207: block trade open short 208: block trade close open 209: block trade close short  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 |
| after | String | No | Pagination of data to return records earlier than the requested billId |
| before | String | No | Pagination of data to return records newer than the requested billId |
| begin | String | No | Filter with a begin timestamp ts. Unix timestamp format in milliseconds, e.g. 1597026383085 |
| end | String | No | Filter with an end timestamp ts. 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": [

{

"side": "buy",

"fillSz": "0.00192834",

"fillPx": "51858",

"fillPxVol": "",

"fillFwdPx": "",

"fee": "-0.00000192834",

"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"

}

],

"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 |
| 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 FUTURES, SWAP, OPTION |
| side | String | Order side buy sell |
| posSide | String | Position side long short it returns net innet mode. |
| execType | String | Liquidity taker or maker T: taker M: 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. 1597026383085. |
| fillTime | String | Trade time which is the same as fillTime for the order channel. |
| feeRate | String | Fee rate. This field is returned for SPOT and MARGIN only |

tradeId  
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.  
clOrdId  
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

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/trade/easy-convert-currency-list

Request Example

GET /api/v5/trade/easy-convert-currency-list

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 | 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 |
| toCcy | Array | Type of mainstream currency convert to, e.g. USDT |

### POST / Place easy convert

Convert small currencies to mainstream currencies.

#### Rate Limit: 1 request per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/trade/easy-convert

Request Example

POST /api/v5/trade/easy-convert

body

{

"fromCcy": ["ADA","USDC"], //Seperated by commas

"toCcy": "OKB"

}

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 | 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": "USDC",

"status": "running",

"toCcy": "OKB",

"uTime": "1661419684687"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| status | String | Current status of easy convert  running: Running  filled: Filled  failed: 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: UserID

#### HTTP Request

GET /api/v5/trade/easy-convert-history

Request Example

GET /api/v5/trade/easy-convert-history

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. 1597026383085 |
| before | String | No | Pagination of data to return records newer than the requested time (exclude), 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": [

{

"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  18: Trading account |
| status | String | Current status of easy convert  running: Running  filled: Filled  failed: 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.

#### Rate Limit: 1 request per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/trade/one-click-repay-currency-list

Request Example

GET /api/v5/trade/one-click-repay-currency-list

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 | Debt currency data list |
| > debtCcy | String | Debt currency type |
| > debtAmt | String | Debt currency amount  Including principal and interest |
| debtType | String | Debt type  cross: cross  isolated: isolated |
| repayData | Array | Repay currency data list |
| > repayCcy | String | Repay currency type |
| > 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.

#### Rate Limit: 1 request per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/trade/one-click-repay

Request Example

POST /api/v5/trade/one-click-repay

body

{

"debtCcy": ["ETH","BTC"],

"repayCcy": "USDT"

}

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 | 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  running: Running  filled: Filled  failed: 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.

#### Rate Limit: 1 request per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/trade/one-click-repay-history

Request Example

GET /api/v5/trade/one-click-repay-history

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 |

### 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: UserID

#### HTTP Request

POST /api/v5/trade/mass-cancel

Request Example

POST /api/v5/trade/mass-cancel

body

{

"instType":"OPTION",

"instFamily":"BTC-USD"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| 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 thorwn when placing order duiring 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 true, false |

### 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: UserID + tag

#### HTTP Request

POST /api/v5/trade/cancel-all-after

Request Example

POST /api/v5/trade/cancel-all-after

{

"timeOut":"60"

}

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. |
| 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. Please note that tag level CAA does not apply to `mmp` and `mmp\_and\_post \_only` order types.   
  
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](https://www.okx.com/docs-v5/en/#overview-rate-limits-fill-ratio-based-sub-account-rate-limit)

#### Rate Limit: 1 request per second

#### Rate limit rule: UserID

#### HTTP Requests

GET /api/v5/trade/account-rate-limit

Request Example

# Get the account rate limit

GET /api/v5/trade/account-rate-limit

#### Request Parameters

None

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: UserID

#### HTTP Request

POST /api/v5/trade/order-precheck

Request Example

place order for SPOT

`POST /api/v5/trade/order-precheck`

body

{

"instId":"BTC-USDT",

"tdMode":"cash",

"clOrdId":"b15",

"side":"buy",

"ordType":"limit",

"px":"2.15",

"sz":"2"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instId | String | Yes | Instrument ID, e.g. BTC-USDT |
| tdMode | String | Yes | Trade mode Margin mode cross isolated Non-Margin mode cash spot\_isolated (only applicable to SPOT lead trading, tdMode should be spot\_isolated for SPOT lead trading.) |
| side | String | Yes | Order side, buy sell |
| posSide | String | Conditional | Position side  The default is net in the net mode  It is required in the long/short mode, and can only be long or short.  Only applicable to FUTURES/SWAP. |
| ordType | String | Yes | 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 (applicable only to Expiry Futures and Perpetual Futures). |
| sz | String | Yes | Quantity to buy or sell |
| px | String | Conditional | Order price. Only applicable to limit,post\_only,fok,ioc,mmp,mmp\_and\_post\_only 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 MARGIN orders, and FUTURES/SWAP orders in net mode  Only applicable to Spot and futures mode and Multi-currency margin |
| tgtCcy | String | No | Whether the target currency uses the quote or base currency. base\_ccy: Base currency ,quote\_ccy: Quote currency  Only applicable to SPOT Market Orders Default is quote\_ccy for buy, base\_ccy for sell |
| attachAlgoOrds | Array of object | 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 algoClOrdId 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. |
| > 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 condition limit The default is condition |
| > slTriggerPx | String | Conditional | Stop-loss trigger price If you fill in this parameter, you should fill in the stop-loss order price. |
| > 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 last: last price  index: index price  mark: mark price  The default is last |
| > slTriggerPxType | String | No | Stop-loss trigger price type last: last price  index: index price  mark: mark price  The default is last |
| > 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": "-57693.044376796888536773622035980224609375",

"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 USD |
| adjEqChg | String | After placing order, changed quantity of adjusted / Effective equity in USD |
| imr | String | Current initial margin requirement in USD |
| imrChg | String | After placing order, changed quantity of initial margin requirement in USD |
| mmr | String | Current Maintenance margin requirement in USD |
| mmrChg | String | After placing order, changed quantity of maintenance margin requirement in USD |
| mgnRatio | String | Current margin ratio in USD |
| mgnRatioChg | String | After placing order, changed quantity of margin ratio in USD |
| availBal | String | Current available balance in margin coin currency, only applicable to turn auto borrow off |
| 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 plaing order, it is base currency. after placing order, it is quota currency. 3: before plaing 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 order updates.

Concurrent connection to this channel will be restricted by the following rules: [WebSocket connection count limit](https://www.okx.com/docs-v5/en/#overview-websocket-connection-count-limit).

#### URL Path

/ws/v5/private (required login)

Request Example : single

{

"op": "subscribe",

"args": [

{

"channel": "orders",

"instType": "FUTURES",

"instId": "BTC-USD-200329"

}

]

}

Request Example

{

"op": "subscribe",

"args": [

{

"channel": "orders",

"instType": "FUTURES",

"instFamily": "BTC-USD"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name orders |
| > instType | String | Yes | Instrument type SPOT MARGIN SWAP FUTURES OPTION ANY |
| > instFamily | String | No | Instrument family Applicable to FUTURES/SWAP/OPTION |
| > instId | String | No | Instrument ID |

Successful Response Example : single

{

"event": "subscribe",

"arg": {

"channel": "orders",

"instType": "FUTURES",

"instId": "BTC-USD-200329"

},

"connId": "a4d3ae55"

}

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "orders",

"instType": "FUTURES",

"instFamily": "BTC-USD"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"orders\", \"instType\" : \"FUTURES\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name |
| > instType | String | Yes | Instrument type SPOT MARGIN SWAP FUTURES OPTION ANY |
| > 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: single

{

"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": "",

"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",

"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 | Subscribed data |
| > instType | String | Instrument type |
| > instId | String | Instrument ID |
| > 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 |
| > ccy | String | Margin currency  Only applicable to cross MARGIN orders in Spot and futures mode. |
| > 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  px: Place an order based on price, in the unit of coin (the unit for the request parameter px is BTC or ETH)  pxVol: Place an order based on pxVol  pxUsd: Place an order based on pxUsd, in the unit of USD (the unit for the request parameter px is USD) |
| > sz | String | The original order quantity, SPOT/MARGIN, in the unit of currency; FUTURES/SWAP/OPTION, in the unit of contract |
| > notionalUsd | String | Estimated national value in USD of order |
| > ordType | String | 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 (applicable only to Expiry Futures and Perpetual Futures) 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) |
| > 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 |
| > fillPx | String | Last filled price |
| > tradeId | String | Last trade ID |
| > fillSz | String | Last filled quantity The unit is base\_ccy for SPOT and MARGIN, e.g. BTC-USDT, the unit is BTC; For market orders, the unit both is base\_ccy when the tgtCcy is base\_ccy or quote\_ccy; The unit is contract for FUTURES/SWAP/OPTION |
| > 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 |
| > fillTime | String | Last filled time |
| > fillFee | String | last filled fee amount or rebate amount:  Negative number represents the user transaction fee charged by the platform;  Positive number represents rebate |
| > fillFeeCcy | String | last filled fee currency or rebate currency. It is fee currency when fillFee is less than 0; It is rebate currency when fillFee>=0. |
| > 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 FUTURES, SWAP, OPTION |
| > execType | String | Liquidity taker or maker of the last filled, T: taker M: maker |
| > accFillSz | String | Accumulated fill quantity The unit is base\_ccy for SPOT and MARGIN, e.g. BTC-USDT, the unit is BTC; For market orders, the unit both is base\_ccy when the tgtCcy is base\_ccy or quote\_ccy; The unit is contract for FUTURES/SWAP/OPTION |
| > fillNotionalUsd | String | Filled notional value in USD of order |
| > avgPx | String | Average filled price. If none is filled, it will return 0. |
| > state | String | Order state  canceled live  partially\_filled  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, it |
| > tpTriggerPxType | String | Take-profit trigger price type.  last: last price index: index price mark: mark price |
| > tpOrdPx | String | Take-profit order price, it |
| > slTriggerPx | String | Stop-loss trigger price, it |
| > slTriggerPxType | String | Stop-loss trigger price type.  last: last price index: index price mark: mark price |
| > slOrdPx | String | Stop-loss order price, it |
| > attachAlgoOrds | Array of object | TP/SL information attached when placing order |
| >> attachAlgoId | String | The order ID of attached TP/SL order. It can be used to identity 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 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. |
| >> 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. |
| >> 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 |
| > linkedAlgoOrd | Object | Linked SL order detail, only applicable to TP limit order of one-cancels-the-other order(oco) |
| >> algoId | Object | Algo ID |
| > stpId | String | (deprecated) |
| > stpMode | String | Self trade prevention mode  Return "" if self trade prevention is not applicable |
| > feeCcy | String | Fee currency  SPOT/MARGIN: If you buy, you will receive base currency; if you sell, you will receive quota currency  FUTURES/SWAP/OPTION What is charged is the margin |
| > fee | String | Fee and rebate For spot and margin, it is accumulated fee charged by the platform. It is always negative, e.g. -0.01.  For Expiry Futures, Perpetual Futures and Options, it is accumulated fee and rebate |
| > rebateCcy | String | Rebate currency, if there is no rebate, this field is "". |
| > rebate | String | Rebate accumulated amount, only applicable to spot and margin, the reward of placing orders from the platform (rebate) given to user who has reached the specified trading level. If there is no rebate, this field is "". |
| > pnl | String | Profit and loss, 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 6: The normal order triggered by the trigger order 7:The normal order triggered by the TP/SL order  13: The normal order triggered by the algo order 25:The normal order triggered by the trailing stop order |
| > cancelSource | String | Source of the order cancellation. Valid values and the corresponding meanings are: 0: Order canceled by system 1: Order canceled by user 2: Order canceled: Pre reduce-only order canceled, due to insufficient margin in user position 3: Order canceled: Risk cancellation was triggered. Pending order was canceled due to insufficient margin ratio and forced-liquidation risk. 4: Order canceled: Borrowings of crypto reached hard cap, order was canceled by system. 6: Order canceled: ADL order cancellation was triggered. Pending order was canceled due to a low margin ratio and forced-liquidation risk.  7: Order canceled: Futures contract delivery.  9: Order canceled: Insufficient balance after funding fees deducted.  13: Order canceled: FOK order was canceled due to incompletely filled. 14: Order canceled: IOC order was partially canceled due to incompletely filled. 15: Order canceled: The order price is beyond the limit 17: Order canceled: Close order was canceled, due to the position was already closed at market price. 20: Cancel all after triggered 21: Order canceled: The TP/SL order was canceled because the position had been closed 22, 23: Order canceled: Reduce-only orders only allow reducing your current position. System has already canceled this order. 27: Order canceled: Price limit verification failed because the price difference between counterparties exceeds 5%  31: The post-only order will take liquidity in taker orders  32: Self trade prevention  33: The order exceeds the maximum number of order matches per taker order 36: Your TP limit order was canceled because the corresponding SL order was triggered.  37: Your TP limit order was canceled because the corresponding SL order was canceled. 38: You have canceled market maker protection (MMP) orders. 39: Your order was canceled because market maker protection (MMP) was triggered. |
| > amendSource | String | Source of the order amendation.  1: Order amended by user 2: Order amended by user, but the order quantity is overriden by system due to reduce-only 3: New order placed by user, but the order quantity is overriden by system due to reduce-only 4: Order amended by system due to other pending orders 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. |
| > category | String | Category  normal  twap  adl  full\_liquidation  partial\_liquidation delivery  ddh: Delta dynamic hedge |
| > 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 | The result of amending the order  -1: failure  0: success  1: Automatic cancel (amendment request returned success but amendment subsequently failed then automatically canceled by the system)  2: Automatic amendation successfully, only applicable to pxVol and pxUsd orders of Option. When amending the order through API and cxlOnFail is set to true in the order amendment request but the amendment is rejected, "" is returned.  When amending the order through API, the order amendment acknowledgement returns success and the amendment subsequently failed, -1 will be returned if cxlOnFail is set to false, 1 will be returned if cxlOnFail is set to true.  When amending the order through Web/APP and the amendment failed, -1 will be returned. |
| > 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 |
| > algoClOrdId | String | Client-supplied Algo ID. There will be a value when algo order attaching algoClOrdId 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 "" |

For market orders, it's likely the orders channel will show order state as "filled" while showing the "last filled quantity (fillSz)" as 0.

When OPTION contract expires, the orders channel doesn’t update for the canceled pending order.

### 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 tradeId > 0.

The channel is exclusively available to users with trading fee tier VIP5 or above. For other users, please use [WS / Order channel](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-order-channel).

#### URL Path

/ws/v5/private (required login)

Request Example: single

{

"op": "subscribe",

"args": [

{

"channel": "fills",

"instId": "BTC-USDT-SWAP"

}

]

}

Request Example

{

"op": "subscribe",

"args": [

{

"channel": "fills"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name fills |
| > instId | String | No | Instrument ID |

Successful Response Example: single

{

"event": "subscribe",

"arg": {

"channel": "fills",

"instId": "BTC-USDT-SWAP"

},

"connId": "a4d3ae55"

}

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "fills"

},

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| 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: 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",

"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 | 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 buy sell |
| > ts | String | Filled time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| > ordId | String | Order ID |
| > tradeId | String | The last trade ID in the trades aggregation |
| > execType | String | Liquidity taker or maker, T: taker M: maker |
| > count | String | The count of trades aggregated |

- The channel is exclusively available to users with trading fee tier VIP5 or above. Others will receive error code 60029 when subscribing to it.   
- 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.   
- When a fill event is received by this channel, the account balance, margin, and position information might not have changed yet.   
- Taker orders will be aggregated based on different fill prices. When aggregation occurs, the count field indicates the number of orders matched, and the tradeId represents the tradeId of the last trade in the aggregation. Maker orders will not be aggregated.   
- 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.

### 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 instruments for Copy Trading: 4 requests per 2 seconds

#### Rate limit rule (except Options): UserID + Instrument ID

#### Rate limit rule (Options only): UserID + Instrument Family

Rate limit of this endpoint will also be affected by the rules [Sub-account rate limit](https://www.okx.com/docs-v5/en/#overview-rate-limits-sub-account-rate-limit) and [Fill ratio based sub-account rate limit](https://www.okx.com/docs-v5/en/#overview-rate-limits-fill-ratio-based-sub-account-rate-limit).

Rate limit is shared with the `Place order` REST API endpoints

Request Example

{

"id": "1512",

"op": "order",

"args": [

{

"side": "buy",

"instId": "BTC-USDT",

"tdMode": "isolated",

"ordType": "market",

"sz": "100"

}

]

}

#### 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 order |
| args | Array | Yes | Request parameters |
| > instId | String | Yes | Instrument ID, e.g. BTC-USDT |
| > tdMode | String | Yes | Trade mode  Margin mode isolated cross  Non-Margin mode cash spot\_isolated (only applicable to SPOT lead trading, tdMode should be spot\_isolated for SPOT lead trading.) |
| > ccy | String | No | Margin currency  Only applicable to cross MARGIN orders in Spot and futures mode. |
| > 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 |
| > posSide | String | Conditional | Position side  The default is net in the net mode  It is required in the long/short mode, and can only be long or short.  Only applicable to FUTURES/SWAP. |
| > ordType | String | Yes | 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) |
| > sz | String | Yes | Quantity to buy or sell. |
| > px | String | Conditional | Order price. Only applicable to limit,post\_only,fok,ioc,mmp,mmp\_and\_post\_only 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 USD  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: true or false. The default value is false. Only applicable to MARGIN orders, and FUTURES/SWAP orders in net mode  Only applicable to Spot and futures mode and Multi-currency margin |
| > tgtCcy | String | No | 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 |
| > banAmend | Boolean | No | Whether to disallow the system from amending the size of the SPOT Market Order. Valid options: true or false. The default value is false. If true, system will not amend and reject the market order if user does not have sufficient funds.  Only applicable to SPOT Market Orders |
| > quickMgnType | String | No | (Deprecated) |
| > stpId | String | No | (deprecated) |
| > stpMode | String | No | Self trade prevention mode.  Default to cancel maker  cancel\_maker,cancel\_taker, cancel\_both Cancel both does not support FOK. |
| expTime | String | No | Request effective deadline. Unix timestamp format in milliseconds, e.g. 1597026383085 |

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": "5XXXX",

"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 | 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 |

tdMode  
Trade Mode, when placing an order, you need to specify the trade mode.  
**Spot mode:**  
- SPOT and OPTION buyer: cash  
**Spot and 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 `Spot and 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. 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 default STP mode is `Cancel Maker`. 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

#### URL Path

/ws/v5/private (required login)

#### Rate Limit: 300 orders per 2 seconds

#### Rate Limit of lead instruments for Copy Trading: 4 orders per 2 seconds

#### Rate limit rule (except Options): UserID + Instrument ID

#### Rate limit rule (Options only): UserID + Instrument Family

Rate limit of this endpoint will also be affected by the rules [Sub-account rate limit](https://www.okx.com/docs-v5/en/#overview-rate-limits-sub-account-rate-limit) and [Fill ratio based sub-account rate limit](https://www.okx.com/docs-v5/en/#overview-rate-limits-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

{

"id": "1513",

"op": "batch-orders",

"args": [

{

"side": "buy",

"instId": "BTC-USDT",

"tdMode": "cash",

"ordType": "market",

"sz": "100"

},

{

"side": "buy",

"instId": "LTC-USDT",

"tdMode": "cash",

"ordType": "market",

"sz": "1"

}

]

}

#### 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 | Yes | Request Parameters |
| > instId | String | Yes | Instrument ID, e.g. BTC-USDT |
| > tdMode | String | Yes | Trade mode  Margin mode isolated cross  Non-Margin mode cash spot\_isolated (only applicable to SPOT lead trading, tdMode should be spot\_isolated for SPOT lead trading.) |
| > ccy | String | No | Margin currency  Only applicable to cross MARGIN orders in Spot and futures mode. |
| > 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 |
| > posSide | String | Conditional | Position side  The default net in the net mode  It is required in the long/short mode, and only be long or short.  Only applicable to FUTURES/SWAP. |
| > ordType | String | Yes | 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 (applicable only to Expiry Futures and Perpetual Futures) 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). |
| > sz | String | Yes | Quantity to buy or sell. |
| > px | String | Conditional | Order price. Only applicable to limit,post\_only,fok,ioc,mmp,mmp\_and\_post\_only 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 USD  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: true or false. The default value is false. Only applicable to MARGIN orders, and FUTURES/SWAP orders in net mode  Only applicable to Spot and futures mode and Multi-currency margin |
| > tgtCcy | String | No | 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 |
| > banAmend | Boolean | No | Whether to disallow the system from amending the size of the SPOT Market Order. Valid options: true or false. The default value is false. If true, system will not amend and reject the market order if user does not have sufficient funds.  Only applicable to SPOT Market Orders |
| > quickMgnType | String | No | (Deprecated) |
| > stpId | String | No | (deprecated) |
| > stpMode | String | No | Self trade prevention mode.  Default to cancel maker  cancel\_maker,cancel\_taker, cancel\_both Cancel both does not support FOK. |
| expTime | String | No | Request effective deadline. Unix timestamp format in milliseconds, e.g. 1597026383085 |

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": "5XXXX",

"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": "5XXXX",

"sMsg": "Insufficient margin"

},

{

"clOrdId": "oktswap7",

"ordId": "",

"tag": "",

"ts": "1695190491421",

"sCode": "5XXXX",

"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 | 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): UserID + Instrument ID

#### Rate limit rule (Options only): UserID + Instrument Family

Rate limit is shared with the `Cancel order` REST API endpoints

Request Example

{

"id": "1514",

"op": "cancel-order",

"args": [

{

"instId": "BTC-USDT",

"ordId": "2510789768709120"

}

]

}

#### 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 | Yes | Request Parameters |
| > instId | String | Yes | Instrument ID |
| > 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": "cancel-order",

"data": [

{

"clOrdId": "",

"ordId": "2510789768709120",

"ts": "1695190491421",

"sCode": "0",

"sMsg": ""

}

],

"code": "0",

"msg": "",

"inTime": "1695190491421339",

"outTime": "1695190491423240"

}

Failure Response Example

{

"id": "1514",

"op": "cancel-order",

"data": [

{

"clOrdId": "",

"ordId": "2510789768709120",

"ts": "1695190491421",

"sCode": "5XXXX",

"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 | 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 the get order state.

### 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): UserID + Instrument ID

#### Rate limit rule (Options only): UserID + 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

{

"id": "1515",

"op": "batch-cancel-orders",

"args": [

{

"instId": "BTC-USDT",

"ordId": "2517748157541376"

},

{

"instId": "LTC-USDT",

"ordId": "2517748155771904"

}

]

}

#### 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-cancel-orders |
| args | Array | Yes | Request Parameters |
| > instId | String | Yes | Instrument ID |
| > 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. |

Response Example When All Succeed

{

"id": "1515",

"op": "batch-cancel-orders",

"data": [

{

"clOrdId": "oktswap6",

"ordId": "2517748157541376",

"ts": "1695190491421",

"sCode": "0",

"sMsg": ""

},

{

"clOrdId": "oktswap7",

"ordId": "2517748155771904",

"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": "5XXXX",

"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": "5XXXX",

"sMsg": "order not exist"

},

{

"clOrdId": "oktswap7",

"ordId": "2517748155771904",

"ts": "1695190491421",

"sCode": "5XXXX",

"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": "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 | 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 instruments for Copy Trading: 4 requests per 2 seconds

#### Rate limit rule (except Options): UserID + Instrument ID

#### Rate limit rule (Options only): UserID + Instrument Family

Rate limit of this endpoint will also be affected by the rules [Sub-account rate limit](https://www.okx.com/docs-v5/en/#overview-rate-limits-sub-account-rate-limit) and [Fill ratio based sub-account rate limit](https://www.okx.com/docs-v5/en/#overview-rate-limits-fill-ratio-based-sub-account-rate-limit).

Rate limit is shared with the `Amend order` REST API endpoints

Request Example

{

"id": "1512",

"op": "amend-order",

"args": [

{

"instId": "BTC-USDT",

"ordId": "2510789768709120",

"newSz": "2"

}

]

}

#### 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 amend-order |
| args | Array | Yes | Request Parameters |
| > instId | String | Yes | Instrument ID |
| > cxlOnFail | Boolean | No | Whether the order needs to be automatically canceled when the order amendment fails  Valid options: false or 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. |
| > newSz | String | Conditional | New quantity after amendment and it has to be larger than 0. Either newSz or newPx is required. 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. |
| 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": "",

"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 | 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 |
| 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 instruments for Copy Trading: 4 orders per 2 seconds

#### Rate limit rule (except Options): UserID + Instrument ID

#### Rate limit rule (Options only): UserID + Instrument Family

Rate limit of this endpoint will also be affected by the rules [Sub-account rate limit](https://www.okx.com/docs-v5/en/#overview-rate-limits-sub-account-rate-limit) and [Fill ratio based sub-account rate limit](https://www.okx.com/docs-v5/en/#overview-rate-limits-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

{

"id": "1513",

"op": "batch-amend-orders",

"args": [

{

"instId": "BTC-USDT",

"ordId": "12345689",

"newSz": "2"

},

{

"instId": "BTC-USDT",

"ordId": "12344",

"newSz": "2"

}

]

}

#### 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 | Yes | Request Parameters |
| > instId | String | Yes | Instrument ID |
| > cxlOnFail | Boolean | No | Whether the order needs to be automatically canceled when the order amendment fails  Valid options: false or 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. |
| > newSz | String | Conditional | New quantity after amendment and it has to be larger than 0. Either newSz or newPx is required. 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. |
| expTime | String | No | Request effective deadline. Unix timestamp format in milliseconds, e.g. 1597026383085 |

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"

}

#### 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 | 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. 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.

### 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: UserID

Rate limit is shared with the `Mass Cancel Order` REST API endpoints

Request Example

{

"id": "1512",

"op": "mass-cancel",

"args": [{

"instType":"OPTION",

"instFamily":"BTC-USD"

}]

}

#### 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 | 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 thorwn when placing order duiring 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 | 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 instruments for Copy Trading: 1 request per 2 seconds

#### Rate limit rule (except Options): UserID + Instrument ID

#### Rate limit rule (Options only): UserID + Instrument Family

#### HTTP Request

POST /api/v5/trade/order-algo

Request Example

# Place Take Profit / Stop Loss Order

POST /api/v5/trade/order-algo

body

{

"instId":"BTC-USDT",

"tdMode":"cross",

"side":"buy",

"ordType":"conditional",

"sz":"2",

"tpTriggerPx":"15",

"tpOrdPx":"18"

}

# Place Trigger Order

POST /api/v5/trade/order-algo

body

{

"instId": "BTC-USDT-SWAP",

"side": "buy",

"tdMode": "cross",

"posSide": "net",

"sz": "1",

"ordType": "trigger",

"triggerPx": "25920",

"triggerPxType": "last",

"orderPx": "-1",

"attachAlgoOrds": [{

"attachAlgoClOrdId": "",

"slTriggerPx": "100",

"slOrdPx": "600",

"tpTriggerPx": "25921",

"tpOrdPx": "2001"

}]

}

# Place Trailing Stop Order

POST /api/v5/trade/order-algo

body

{

"instId": "BTC-USDT-SWAP",

"tdMode": "cross",

"side": "buy",

"ordType": "move\_order\_stop",

"sz": "10",

"posSide": "net",

"callbackRatio": "0.05",

"reduceOnly": true

}

# Place TWAP Order

POST /api/v5/trade/order-algo

body

{

"instId": "BTC-USDT-SWAP",

"tdMode": "cross",

"side": "buy",

"ordType": "twap",

"sz": "10",

"posSide": "net",

"szLimit": "10",

"pxLimit": "100",

"timeInterval": "10",

"pxSpread": "10"

}

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. BTC-USDT |
| tdMode | String | Yes | Trade mode Margin mode cross isolated Non-Margin mode cash spot\_isolated (only applicable to SPOT lead trading) |
| ccy | String | No | Margin currency  Only applicable to cross MARGIN orders in Spot and futures mode. |
| side | String | Yes | Order side, buy sell |
| posSide | String | Conditional | Position side  Required in long/short mode and only be long or short |
| ordType | String | Yes | Order type  conditional: One-way stop order oco: One-cancels-the-other order chase: chase order, only applicable to FUTURES and SWAP trigger: Trigger order move\_order\_stop: Trailing order twap: TWAP order |
| sz | String | Conditional | Quantity to buy or sell Either sz or closeFraction is required. |
| tag | String | No | Order tag  A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters. |
| tgtCcy | String | No | Order quantity unit setting for sz base\_ccy: Base currency ,quote\_ccy: Quote currency  Only applicable to SPOT traded with Market buy conditional order Default is quote\_ccy for buy, base\_ccy for sell |
| algoClOrdId | String | No | Client-supplied Algo ID A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. |
| closeFraction | String | Conditional | 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 1. For the same position, only one TPSL pending order for fully closing the position is supported.  This is only applicable to FUTURES or SWAP instruments. If posSide is net, reduceOnly must be true. This is only applicable if ordType is conditional or oco. 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 sz or closeFraction is required. |

**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](https://www.okx.com/help/11015447687437)

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| tpTriggerPx | String | No | Take-profit trigger price If you fill in this parameter, you should fill in the take-profit order price as well. |
| tpTriggerPxType | String | No | Take-profit trigger price type last: last price index: index price mark: mark price  The default is last |
| 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 -1, take-profit will be executed at the market price. |
| tpOrdKind | String | No | TP order kind condition limit The default is condition |
| slTriggerPx | String | No | Stop-loss trigger price  If you fill in this parameter, you should fill in the stop-loss order price. |
| slTriggerPxType | String | No | Stop-loss trigger price type last: last price index: index price mark: mark price  The default is last |
| 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 -1, stop-loss will be executed at the market price. |
| cxlOnClosePos | Boolean | No | 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.  Valid values:  true: Place a TP/SL order associated with the position  false: Place a TP/SL order that is not associated with the position  The default value is false. If true is passed in, users must pass reduceOnly = true as well, indicating that when placing a TP/SL order associated with a position, it must be a reduceOnly order.  Only applicable to Spot and futures mode and Multi-currency margin. |
| reduceOnly | Boolean | No | Whether the order can only reduce the position size.  Valid options: true or false. The default value is false. |
| quickMgnType | String | No | (Deprecated) |

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 | Chase type. distance: distance from best bid/ask price, the default value. ratio: ratio. |
| chaseVal | String | No | Chase value. It represents distance from best bid/ask price when chaseType is distance.  For USDT-margined contract, the unit is USDT.  For USDC-margined contract, the unit is USDC.  For Crypto-margined contract, the unit is USD.  It represents ratio when chaseType is ratio. 0.1 represents 10%. The default value is 0. |
| maxChaseType | String | Conditional | Maximum chase type. distance: maximum distance from best bid/ask price ratio: the ratio.   maxChaseTyep and maxChaseVal need to be used together or none of them. |
| maxChaseVal | String | Conditional | Maximum chase value. It represents maximum distance when maxChaseType is distance. It represents ratio when maxChaseType is ratio. 0.1 represents 10%. |
| reduceOnly | Boolean | No | Whether the order can only reduce the position size.  Valid options: true or false. The default value is false. |

**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](https://www.okx.com/help/11015447687437)

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| triggerPx | String | Yes | Trigger price |
| orderPx | String | Yes | Order Price  If the price is -1, the order will be executed at the market price. |
| triggerPxType | String | No | Trigger price type  last: last price index: index price mark: mark price  The default is last |
| quickMgnType | String | No | (Deprecated) |
| attachAlgoOrds | Array of object | No | Attached SL/TP orders info Applicable to Spot and futures mode/Multi-currency margin/Portfolio margin |
| > 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. |
| > tpTriggerPxType | String | No | Take-profit trigger price type last: last price index: index price mark: mark price The default is last |
| > 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 -1, 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. |
| > slTriggerPxType | String | No | Stop-loss trigger price type last: last price index: index price mark: mark price  The default is last |
| > 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 -1, 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 × (1 – 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 × (1 + Trail variance).  
You can use the activation price to set the activation condition for a trailing stop order.

learn more about [Trailing Stop Order](https://www.okx.com/help/11015447687437)

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| callbackRatio | String | Conditional | Callback price ratio, e.g. 0.01 represents 1% Either callbackRatio or callbackSpread 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. |
| quickMgnType | String | No | (Deprecated) |
| reduceOnly | Boolean | No | Whether the order can only reduce the position size.  Valid options: true or false. The default value is false. This parameter is only valid in the FUTRUES/SWAP 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](https://www.okx.com/help/xiii-time-weighted-average-price-twap)

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| pxVar | String | Conditional | Price variance by percentage, range between [0.0001 ~ 0.01], e.g. 0.01 represents 1% 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 pxVar or pxSpread is allowed to be passed. |
| pxSpread | String | Conditional | Price variance by constant, should be no less then 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 then 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 second ake 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 | (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): UserID + Instrument ID

#### Rate limit rule (Options only): UserID + Instrument Family

#### HTTP Request

POST /api/v5/trade/cancel-algos

Request Example

POST /api/v5/trade/cancel-algos

body

[

{

"algoId":"590919993110396111",

"instId":"BTC-USDT"

},

{

"algoId":"590920138287841222",

"instId":"BTC-USDT"

}

]

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** |
| --- | --- | --- | --- |
| algoId | String | Yes | Algo ID |
| instId | String | Yes | Instrument ID, e.g. BTC-USDT |

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, 0 means success. |
| sMsg | String | Rejection message if the request is unsuccessful. |
| clOrdId | String | (Deprecated) |
| algoClOrdId | String | (Deprecated) |
| tag | String | (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: UserID + Instrument ID

#### HTTP Request

POST /api/v5/trade/amend-algos

Request Example

POST /api/v5/trade/amend-algos

body

{

"algoId":"2510789768709120",

"newSz":"2",

"instId":"BTC-USDT"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instId | String | Yes | Instrument ID |
| algoId | String | Conditional | Algo ID Either algoId or algoClOrdId is required. If both are passed, algoId will be used. |
| algoClOrdId | String | Conditional | Client-supplied Algo ID Either algoId or algoClOrdId is required. If both are passed, algoId will be used. |
| cxlOnFail | Boolean | No | Whether the order needs to be automatically canceled when the order amendment fails  Valid options: false or true, the default is false. |
| 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 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. |

**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 last: last price  index: index price  mark: mark price |
| newSlTriggerPxType | String | Conditional | Stop-loss trigger price type last: last price  index: index price  mark: 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 -1, the order will be executed at the market price. |
| newTriggerPxType | String | No | New trigger price type after amendment  last: last price index: index price mark: mark price  The default is last |
| attachAlgoOrds | Array of object | No | Attached SL/TP orders info Applicable to Spot and futures mode/Multi-currency margin/Portfolio margin |
| > newTpTriggerPx | String | No | Take-profit trigger price If you fill in this parameter, you should fill in the take-profit order price as well. |
| > 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. |
| > newSlTriggerPx | String | No | Stop-loss trigger price If you fill in this parameter, you should fill in the stop-loss order price. |
| > newSlTriggerPxType | String | No | Stop-loss trigger price type last: last price index: index price mark: mark price  The default is last |
| > newSlOrdPx | 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":"po103ux",

"sCode":"0",

"sMsg":""

}

]

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| algoId | 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. |

### POST / Cancel advance algo order

This endpoint will be offline soon, please use [Cancel algo order](https://www.okx.com/docs-v5/en/#order-book-trading-algo-trading-post-cancel-algo-order)

Cancel unfilled algo orders (including Iceberg order, TWAP order, Trailing Stop order). 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): UserID + Instrument ID

#### Rate limit rule (Options only): UserID + Instrument Family

#### HTTP Request

POST /api/v5/trade/cancel-advance-algos

Request Example

POST /api/v5/trade/cancel-advance-algos

body

[

{

"algoId":"590920768125665111",

"instId":"BTC-USDT"

},

{

"algoId":"590920799650058222",

"instId":"BTC-USDT"

}

]

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 (including Iceberg order, TWAP order, Trailing Stop order)

algo\_orders\_advance = [

{"instId": "BTC-USDT", "algoId": "590920768125665111"},

{"instId": "BTC-USDT", "algoId": "590920799650058222"}

]

result = tradeAPI.cancel\_advance\_algos(algo\_orders\_advance)

print(result)

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | String | Yes | Algo ID |
| instId | String | Yes | Instrument ID, e.g. BTC-USDT |

Response Example

{

"code":"0",

"msg":"",

"data":[

{

"algoId":"1234",

"sCode":"0",

"sMsg":""

}

]

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| algoId | String | Order ID |
| 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: UserID

#### HTTP Request

GET /api/v5/trade/order-algo

Request Example

GET /api/v5/trade/order-algo?algoId=1753184812254216192

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | String | Conditional | Algo ID Either algoId or algoClOrdId is required.If both are passed, algoId 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": "",

"algoClOrdId": "",

"algoId": "681187161907138560",

"amendPxOnTriggerType": "",

"attachAlgoOrds": [],

"cTime": "1708679675244",

"uTime": "1708679675245",

"callbackRatio": "0.05",

"callbackSpread": "",

"ccy": "",

"clOrdId": "",

"closeFraction": "",

"failCode": "",

"instId": "BTC-USDT",

"instType": "SPOT",

"last": "50962.7",

"lever": "",

"linkedOrd": {

"ordId": ""

},

"moveTriggerPx": "53423.160",

"ordId": "",

"ordIdList": [],

"ordPx": "",

"ordType": "move\_order\_stop",

"posSide": "net",

"pxLimit": "",

"pxSpread": "",

"pxVar": "",

"quickMgnType": "",

"reduceOnly": "false",

"side": "buy",

"slOrdPx": "",

"slTriggerPx": "",

"slTriggerPxType": "",

"state": "live",

"sz": "10",

"szLimit": "",

"tag": "",

"tdMode": "cash",

"tgtCcy": "",

"timeInterval": "",

"tpOrdPx": "",

"tpTriggerPx": "",

"tpTriggerPxType": "",

"triggerPx": "",

"triggerPxType": "",

"triggerTime": "",

"isTradeBorrowMode": "true"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| instType | String | Instrument type |
| instId | String | Instrument ID |
| ccy | String | Margin currency  Only applicable to cross MARGIN orders in Spot and futures mode. |
| ordId | String | Latest order ID. It will be deprecated soon |
| ordIdList | Array | Order ID list. There will be multiple order IDs when there is TP/SL splitting order. |
| algoId | 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 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 |
| state | String | State  live  pause  partially\_effective effective  canceled  order\_failed partially\_failed |
| lever | String | Leverage, from 0.01 to 125.  Only applicable to MARGIN/FUTURES/SWAP |
| 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. |
| 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 |
| failCode | String | It represents that the reason that algo order fails to trigger. It is "" when the state is effective/canceled. There will be value when the state is order\_failed, e.g. 51008; Only applicable to Stop Order, Trailing Stop Order, Trigger order. |
| 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  1: Enable |
| attachAlgoOrds | Array of object | Attached SL/TP orders info Applicable to Spot and futures mode/Multi-currency margin/Portfolio margin |
| > 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. |
| > 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. |
| > 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 trigger order, trailing order and twap order |
| chaseType | String | Chase type. Only applicable to chase order. |
| chaseVal | String | Chase value. Only applicable to chase order. |
| maxChaseType | String | Maximum chase type. Only applicable to chase order. |
| maxChaseVal | String | Maximum chase value. Only applicable to chase order. |

### 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: UserID

#### HTTP Request

GET /api/v5/trade/orders-algo-pending

Request Example

GET /api/v5/trade/orders-algo-pending?ordType=conditional

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 | Order type conditional: One-way stop order  oco: One-cancels-the-other order  chase: chase order, only applicable to FUTURES and SWAP trigger: Trigger order  move\_order\_stop: Trailing order  iceberg: Iceberg order  twap: TWAP order For every request, unlike other ordType which only can use one type, conditional and oco both can be used and separated with comma. |
| algoId | String | No | Algo ID |
| algoClOrdId | String | No | Client-supplied Algo ID A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. |
| instType | String | No | Instrument type SPOT SWAP FUTURES MARGIN |
| instId | String | No | Instrument ID, e.g. BTC-USDT |
| 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": [

{

"activePx": "",

"actualPx": "",

"actualSide": "buy",

"actualSz": "0",

"algoClOrdId": "",

"algoId": "681096944655273984",

"amendPxOnTriggerType": "",

"attachAlgoOrds": [],

"cTime": "1708658165774",

"uTime": "1708679675245",

"callbackRatio": "",

"callbackSpread": "",

"ccy": "",

"clOrdId": "",

"closeFraction": "",

"failCode": "",

"instId": "BTC-USDT",

"instType": "SPOT",

"last": "51014.6",

"lever": "",

"moveTriggerPx": "",

"ordId": "",

"ordIdList": [],

"ordPx": "-1",

"ordType": "trigger",

"posSide": "net",

"pxLimit": "",

"pxSpread": "",

"pxVar": "",

"quickMgnType": "",

"reduceOnly": "false",

"side": "buy",

"slOrdPx": "",

"slTriggerPx": "",

"slTriggerPxType": "",

"state": "live",

"sz": "10",

"szLimit": "",

"tag": "",

"tdMode": "cash",

"tgtCcy": "",

"timeInterval": "",

"tpOrdPx": "",

"tpTriggerPx": "",

"tpTriggerPxType": "",

"triggerPx": "100",

"triggerPxType": "last",

"triggerTime": "0",

"linkedOrd":{

"ordId":"98192973880283",

},

"isTradeBorrowMode": "true"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| instType | String | Instrument type |
| instId | String | Instrument ID |
| ccy | String | Margin currency  Only applicable to cross MARGIN orders in Spot and futures mode. |
| ordId | String | Latest order ID. It will be deprecated soon |
| ordIdList | Array | Order ID list. There will be multiple order IDs when there is TP/SL splitting order. |
| algoId | 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 sz base\_ccy: Base currency ,quote\_ccy: Quote currency  Only applicable to SPOT traded with Market order |
| state | String | State live pause |
| lever | String | Leverage, from 0.01 to 125.  Only applicable to MARGIN/FUTURES/SWAP |
| 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 |
| 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 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 |
| failCode | String | It represents that the reason that algo order fails to trigger. There will be value when the state is order\_failed, 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  1: Enable |
| attachAlgoOrds | Array of object | Attached SL/TP orders info Applicable to Spot and futures mode/Multi-currency margin/Portfolio margin |
| > 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. |
| > 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. |
| > 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 trigger order, trailing order and twap order |
| chaseType | String | Chase type. Only applicable to chase order. |
| chaseVal | String | Chase value. Only applicable to chase order. |
| maxChaseType | String | Maximum chase type. Only applicable to chase order. |
| maxChaseVal | String | Maximum chase value. Only applicable to chase order. |

### GET / Algo order history

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: UserID

#### HTTP Request

GET /api/v5/trade/orders-algo-history

Request Example

GET /api/v5/trade/orders-algo-history?ordType=conditional&state=effective

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 | Order type  conditional: One-way stop order  oco: One-cancels-the-other order  chase: chase order, only applicable to FUTURES and SWAP trigger: Trigger order  move\_order\_stop: Trailing order  iceberg: Iceberg order  twap: TWAP order For every request, unlike other ordType which only can use one type, conditional and oco both can be used and separated with comma. |
| state | String | Conditional | State effective canceled order\_failed Either state or algoId is required |
| algoId | String | Conditional | Algo ID  Either state or algoId is required. |
| instType | String | No | Instrument type SPOT SWAP FUTURES MARGIN |
| instId | String | No | Instrument ID, e.g. BTC-USDT |
| after | String | No | Pagination of data to return records earlier than the requested algoId |
| before | String | No | Pagination of data to return records new 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": [

{

"activePx": "",

"actualPx": "",

"actualSide": "buy",

"actualSz": "0",

"algoClOrdId": "",

"algoId": "681096944655273984",

"amendPxOnTriggerType": "",

"attachAlgoOrds": [],

"cTime": "1708658165774",

"uTime": "1708679675245",

"callbackRatio": "",

"callbackSpread": "",

"ccy": "",

"clOrdId": "",

"closeFraction": "",

"failCode": "",

"instId": "BTC-USDT",

"instType": "SPOT",

"last": "51014.6",

"lever": "",

"moveTriggerPx": "",

"ordId": "",

"ordIdList": [],

"ordPx": "-1",

"ordType": "trigger",

"posSide": "net",

"pxLimit": "",

"pxSpread": "",

"pxVar": "",

"quickMgnType": "",

"reduceOnly": "false",

"side": "buy",

"slOrdPx": "",

"slTriggerPx": "",

"slTriggerPxType": "",

"state": "canceled",

"sz": "10",

"szLimit": "",

"tag": "",

"tdMode": "cash",

"tgtCcy": "",

"timeInterval": "",

"tpOrdPx": "",

"tpTriggerPx": "",

"tpTriggerPxType": "",

"triggerPx": "100",

"triggerPxType": "last",

"triggerTime": "",

"linkedOrd":{

"ordId":"98192973880283",

},

"isTradeBorrowMode": "true"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| instType | String | Instrument type |
| instId | String | Instrument ID |
| ccy | String | Margin currency  Only applicable to cross MARGIN orders in Spot and futures mode. |
| ordId | String | Latest order ID. It will be deprecated soon |
| ordIdList | Array | Order ID list. There will be multiple order IDs when there is TP/SL splitting order. |
| algoId | 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 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 |
| state | String | State  effective  canceled  order\_failed  partially\_failed |
| lever | String | Leverage, from 0.01 to 125.  Only applicable to MARGIN/FUTURES/SWAP |
| 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. |
| 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 |
| failCode | String | It represents that the reason that algo order fails to trigger. It is "" when the state is effective/canceled. There will be value when the state is order\_failed, e.g. 51008; Only applicable to Stop Order, Trailing Stop Order, Trigger order. |
| algoClOrdId | String | Client Algo Order ID as assigned by the client. |
| 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 object | Attached SL/TP orders info Applicable to Spot and futures mode/Multi-currency margin/Portfolio margin |
| > 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. |
| > 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. |
| > 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 trigger order, trailing order and twap order |
| chaseType | String | Chase type. Only applicable to chase order. |
| chaseVal | String | Chase value. Only applicable to chase order. |
| maxChaseType | String | Maximum chase type. Only applicable to chase order. |
| maxChaseVal | String | Maximum chase value. Only applicable to chase order. |

### 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

{

"op": "subscribe",

"args": [

{

"channel": "orders-algo",

"instType": "FUTURES",

"instFamily": "BTC-USD",

"instId": "BTC-USD-200329"

}

]

}

Request Example

{

"op": "subscribe",

"args": [

{

"channel": "orders-algo",

"instType": "FUTURES",

"instFamily": "BTC-USD"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name orders-algo |
| > instType | String | Yes | Instrument type  SPOT MARGIN SWAP FUTURES  ANY |
| > instFamily | String | No | Instrument family Applicable to FUTURES/SWAP/OPTION |
| > instId | String | No | Instrument ID |

Successful Response Example : single

{

"event": "subscribe",

"arg": {

"channel": "orders-algo",

"instType": "FUTURES",

"instFamily": "BTC-USD",

"instId": "BTC-USD-200329"

},

"connId": "a4d3ae55"

}

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "orders-algo",

"instType": "FUTURES",

"instFamily": "BTC-USD"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"orders-algo\", \"instType\" : \"FUTURES\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name |
| > instType | String | Yes | Instrument type  SPOT MARGIN SWAP FUTURES  ANY |
| > 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 |
| 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",

"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": "",

"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 | Subscribed data |
| > instType | String | Instrument type |
| > instId | String | Instrument ID |
| > ccy | String | Margin currency  Only applicable to cross MARGIN orders in Spot and futures mode. |
| > ordId | String | Latest order ID, the order ID associated with the algo order. It will be deprecated soon |
| > ordIdList | Array | Order ID list. There will be multiple order IDs when there is TP/SL splitting order. |
| > algoId | 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 |
| > 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 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 |
| > last | String | Last filled price while placing |
| > actualSz | String | Actual order quantity |
| > actualPx | String | Actual order price |
| > notionalUsd | String | Estimated national value in USD 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. 1597026383085 |
| > reduceOnly | String | Whether the order can only reduce the position size. Valid options: true or false. |
| > failCode | String | It represents that the reason that algo order fails to trigger. It is "" when the state is effective/canceled. There will be value when the state is order\_failed, 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 -1: failure  0: success |
| > 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 object | Attached SL/TP orders info Applicable to Spot and futures mode/Multi-currency margin/Portfolio margin |
| >> 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. |
| >> 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. |
| >> 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 trigger order, trailing order and twap order |
| > chaseType | String | Chase type. Only applicable to chase order. |
| > chaseVal | String | Chase value. Only applicable to chase order. |
| > maxChaseType | String | Maximum chase type. Only applicable to chase order. |
| > maxChaseVal | String | Maximum chase value. Only applicable to chase order. |

### 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

{

"op": "subscribe",

"args": [

{

"channel": "algo-advance",

"instType": "SPOT",

"instId": "BTC-USDT"

}

]

}

Request Example

{

"op": "subscribe",

"args": [

{

"channel": "algo-advance",

"instType": "SPOT"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name algo-advance |
| > instType | String | Yes | Instrument type  SPOT MARGIN SWAP FUTURES  ANY |
| > instId | String | No | Instrument ID |
| > algoId | String | No | Algo Order ID |

Successful Response Example : single

{

"event": "subscribe",

"arg": {

"channel": "algo-advance",

"instType": "SPOT",

"instId": "BTC-USDT"

},

"connId": "a4d3ae55"

}

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "algo-advance",

"instType": "SPOT"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"algo-advance\", \"instType\" : \"FUTURES\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name |
| > instType | String | Yes | Instrument type  SPOT MARGIN SWAP FUTURES  ANY |
| > instId | String | No | Instrument ID |
| > algoId | 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":"",

"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 | Subscribed data |
| > instType | String | Instrument type |
| > instId | String | Instrument ID |
| > ccy | String | Margin currency  Only applicable to cross MARGIN orders in Spot and futures mode. |
| > 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. SPOT/MARGIN: in the unit of currency. FUTURES/SWAP/OPTION: in the unit of contract. |
| > ordType | String | Order type  iceberg: Iceberg order  twap: TWAP order  move\_order\_stop: Trailing 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 partially\_effective: partially effective canceled: canceled  order\_failed: order failed  pause: 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 USD of order |
| > tag | String | Order tag |
| > actualSide | String | Actual trigger side |
| > triggerTime | String | Trigger time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| > cTime | String | Creation 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 |
| > count | String | Algo Order count  Only applicable to iceberg order or 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 |
| > failCode | String | It represents that the reason that algo order fails to trigger. It is "" when the state is effective/canceled. There will be value when the state is order\_failed, 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: true or false. |
| > pTime | String | Push time of algo order information, millisecond format of Unix timestamp, e.g. 1597026383085 |
| > isTradeBorrowMode | Boolean | Whether borrowing currency automatically true false Only applicable to trigger order, trailing order and twap order |

## 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: UserID + Instrument ID

#### HTTP Request

POST /api/v5/tradingBot/grid/order-algo

Request Example

# Place spot grid algo order

POST /api/v5/tradingBot/grid/order-algo

body

{

"instId": "BTC-USDT",

"algoOrdType": "grid",

"maxPx": "5000",

"minPx": "400",

"gridNum": "10",

"runType": "1",

"quoteSz": "25",

"triggerParams":[

{

"triggerAction":"stop",

"triggerStrategy":"price",

"triggerPx":"1000"

}

]

}

# Place contract grid algo order

POST /api/v5/tradingBot/grid/order-algo

body

{

"instId": "BTC-USDT-SWAP",

"algoOrdType": "contract\_grid",

"maxPx": "5000",

"minPx": "400",

"gridNum": "10",

"runType": "1",

"sz": "200",

"direction": "long",

"lever": "2",

"triggerParams":[

{

"triggerAction":"start",

"triggerStrategy":"rsi",

"timeframe":"30m",

"thold":"10",

"triggerCond":"cross",

"timePeriod":"14"

},

{

"triggerAction":"stop",

"triggerStrategy":"price",

"triggerPx":"1000",

"stopType":"2"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instId | String | Yes | Instrument ID, e.g. BTC-USDT-SWAP |
| 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 | No | Grid type 1: Arithmetic, 2: Geometric Default is Arithmetic |
| tpTriggerPx | String | No | TP tigger price Applicable to Spot grid/Contract grid |
| slTriggerPx | String | No | SL tigger price Applicable to Spot grid/Contract grid |
| algoClOrdId | 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 0,0.1,0.2,0.3 0.1 represents 10% |
| triggerParams | Array of object | No | Trigger Parameters Applicable to Spot grid/Contract grid |
| > triggerAction | String | Yes | Trigger action start stop |
| > triggerStrategy | String | Yes | Trigger strategy instant price rsi Default is instant |
| > delaySeconds | String | No | Delay seconds after action triggered |
| > timeframe | String | No | 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 | No | Threshold The value should be an integer between 1 to 100 This field is only valid when triggerStrategy is rsi |
| > triggerCond | String | No | Trigger condition cross\_up cross\_down above below cross This field is only valid when triggerStrategy is rsi |
| > timePeriod | String | No | Time Period 14 This field is only valid when triggerStrategy is rsi |
| > triggerPx | String | No | Trigger Price This field is only valid when triggerStrategy is price |
| > 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 triggerAction is stop |

Spot Grid Order

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| quoteSz | String | Conditional | Invest amount for quote currency Either quoteSz or baseSz is required |
| baseSz | String | Conditional | Invest amount for base currency Either quoteSz or baseSz is required |

Contract Grid Order

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| sz | String | Yes | Used margin based on USDT |
| direction | String | Yes | Contract grid type long,short,neutral |
| lever | String | Yes | Leverage |
| basePos | Boolean | No | Whether or not open a position when the strategy activates  Default is false 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** |
| --- | --- | --- |
| 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 grid algo order

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/tradingBot/grid/amend-order-algo

Request Example

POST /api/v5/tradingBot/grid/amend-order-algo

body

{

"algoId":"448965992920907776",

"instId":"BTC-USDT-SWAP",

"slTriggerPx":"1200",

"tpTriggerPx":""

}

POST /api/v5/tradingBot/grid/amend-order-algo

body

{

"algoId":"578963447615062016",

"instId":"BTC-USDT",

"triggerParams":[

{

"triggerAction":"stop",

"triggerStrategy":"price",

"triggerPx":"1000"

}

]

}

POST /api/v5/tradingBot/grid/amend-order-algo

body

{

"algoId":"578963447615062016",

"instId":"BTC-USDT-SWAP",

"triggerParams":[

{

"triggerAction":"stop",

"triggerStrategy":"instant",

"stopType":"1"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | String | Yes | Algo ID |
| instId | String | Yes | Instrument ID, e.g. BTC-USDT-SWAP |
| slTriggerPx | String | No | New stop-loss trigger price if slTriggerPx is set "" means stop-loss trigger price is canceled. Either slTriggerPx or tpTriggerPx is required. |
| tpTriggerPx | String | No | New take-profit trigger price if tpTriggerPx 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. |
| 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. |
| triggerParams | Array of object | No | Trigger Parameters |
| > triggerAction | String | Yes | Trigger action start stop |
| > triggerStrategy | String | Yes | Trigger strategy instant price rsi |
| > triggerPx | String | No | Trigger Price This field is only valid when triggerStrategy is price |
| > 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 triggerAction is stop |

Response Example

{

"code":"0",

"msg":"",

"data":[

{

"algoClOrdId": "",

"algoId":"448965992920907776",

"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 / Stop grid algo order

A maximum of 10 orders can be stopped per request.

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/tradingBot/grid/stop-order-algo

Request Example

POST /api/v5/tradingBot/grid/stop-order-algo

body

[

{

"algoId":"448965992920907776",

"instId":"BTC-USDT",

"stopType":"1",

"algoOrdType":"grid"

}

]

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | String | Yes | Algo ID |
| instId | String | Yes | Instrument ID, e.g. BTC-USDT |
| algoOrdType | String | Yes | Algo order type grid: Spot grid contract\_grid: Contract grid |
| stopType | String | Yes | Stop type Spot grid 1: Sell base currency 2: Keep base currency Contract grid 1: Market Close All positions 2: Keep positions |

Response Example

{

"code": "0",

"data": [

{

"algoClOrdId": "",

"algoId": "448965992920907776",

"sCode": "0",

"sMsg": "",

"tag": ""

}

],

"msg": ""

}

#### 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 / 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: UserID

#### HTTP Request

POST /api/v5/tradingBot/grid/close-position

Request Example

POST /api/v5/tradingBot/grid/close-position

body

{

"algoId":"448965992920907776",

"mktClose":true

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | 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** |
| --- | --- | --- |
| algoId | 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: UserID

#### HTTP Request

POST /api/v5/tradingBot/grid/cancel-close-order

Request Example

POST /api/v5/tradingBot/grid/cancel-close-order

body

{

"algoId":"448965992920907776",

"ordId":"570627699870375936"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | 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** |
| --- | --- | --- |
| algoId | 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: UserID + Instrument ID

#### HTTP Request

POST /api/v5/tradingBot/grid/order-instant-trigger

Request Example

POST /api/v5/tradingBot/grid/order-instant-trigger

body

{

"algoId":"561564133246894080"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | String | Yes | Algo ID |

Response Example

{

"code": "0",

"data": [

{

"algoClOrdId": "",

"algoId": "561564133246894080"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| algoId | String | Algo ID |
| algoClOrdId | String | Client-supplied Algo ID |

### GET / Grid algo order list

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/tradingBot/grid/orders-algo-pending

Request Example

GET /api/v5/tradingBot/grid/orders-algo-pending?algoOrdType=grid

#### 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": "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",

"triggerType": "manual",

"triggerTime": ""

}

],

"uTime": "1682062564350",

"uly": "BTC-USDT",

"profitSharingRatio": "",

"copyType": "0",

"tpRatio": "",

"slRatio": "",

"fee": "",

"fundingFee": ""

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| 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 |
| 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 object | Rebate transfer info |
| > rebate | String | Rebate amount |
| > rebateCcy | String | Rebate currency |
| triggerParams | Array of object | 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 Contract grid 1: Market Close All positions 2: Keep positions This field is only valid when triggerAction is stop |
| 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 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 Spot grid |
| baseSz | String | Base currency investment amount Only applicable to Spot grid |
| direction | String | Contract grid type long,short,neutral Only applicable to contract grid |
| basePos | Boolean | Whether or not to open a position when the strategy is activated Only applicable to contract grid |
| sz | String | Used margin based on USDT Only applicable to contract grid |
| lever | String | Leverage Only applicable to contract grid |
| actualLever | String | Actual Leverage Only applicable to contract grid |
| liqPx | String | Estimated liquidation price Only applicable to contract grid |
| uly | String | Underlying Only applicable to contract grid |
| instFamily | String | Instrument family Only applicable to FUTURES/SWAP/OPTION Only applicable to contract grid |
| ordFrozen | String | Margin used by pending orders Only applicable to contract grid |
| availEq | String | Available margin Only applicable to contract grid |
| 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 |
| 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 "" |

### GET / Grid algo order history

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/tradingBot/grid/orders-algo-history

Request Example

GET /api/v5/tradingBot/grid/orders-algo-history?algoOrdType=grid

#### 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",

"triggerType": "manual",

"triggerTime": "1681181186484"

}

],

"uTime": "1681181186496",

"uly": "BTC-USDT",

"profitSharingRatio": "",

"copyType": "0",

"tpRatio": "",

"slRatio": "",

"fee": "",

"fundingFee": ""

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| 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 |
| algoOrdType | String | Algo order type grid: Spot grid contract\_grid: Contract grid |
| state | String | Algo order state stopped |
| rebateTrans | Array of object | Rebate transfer info |
| > rebate | String | Rebate amount |
| > rebateCcy | String | Rebate currency |
| triggerParams | Array of object | 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 Contract grid 1: Market Close All positions 2: Keep positions This field is only valid when triggerAction is stop |
| 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 Spot grid |
| baseSz | String | Base currency investment amount Only applicable to Spot grid |
| direction | String | Contract grid type long,short,neutral Only applicable to contract grid |
| basePos | Boolean | Whether or not to open a position when the strategy is activated Only applicable to contract grid |
| sz | String | Used margin based on USDT Only applicable to contract grid |
| lever | String | Leverage Only applicable to contract grid |
| actualLever | String | Actual Leverage Only applicable to contract grid |
| liqPx | String | Estimated liquidation price Only applicable to contract grid |
| uly | String | Underlying Only applicable to contract grid |
| instFamily | String | Instrument family Only applicable to FUTURES/SWAP/OPTION Only applicable to contract grid |
| ordFrozen | String | Margin used by pending orders Only applicable to contract grid |
| availEq | String | Available margin Only applicable to contract grid |
| 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 |
| 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 "" |

### GET / Grid algo order details

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/tradingBot/grid/orders-algo-details

Request Example

GET /api/v5/tradingBot/grid/orders-algo-details?algoId=448965992920907776&algoOrdType=grid

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoOrdType | String | Yes | Algo order type grid: Spot grid contract\_grid: Contract grid |
| 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",

"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": ""

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| 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 |
| algoOrdType | String | Algo order type grid: Spot grid contract\_grid: Contract grid |
| state | String | Algo order state starting running stopping no\_close\_position: stopped algo order but have not closed position yet stopped |
| rebateTrans | Array of object | Rebate transfer info |
| > rebate | String | Rebate amount |
| > rebateCcy | String | Rebate currency |
| triggerParams | Array of object | 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 Contract grid 1: Market Close All positions 2: Keep positions This field is only valid when triggerAction is stop |
| 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 |
| 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 | 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 0: None 1: Manual stop 2: Take profit 3: Stop loss 4: Risk control 5: Delivery 6: 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 |
| activeOrdNum | String | Total count of pending sub orders |
| quoteSz | String | Quote currency investment amount Only applicable to Spot grid |
| baseSz | String | Base currency investment amount Only applicable to Spot grid |
| curQuoteSz | String | Assets of quote currency currently held Only applicable to Spot grid |
| curBaseSz | String | Assets of base currency currently held Only applicable to Spot grid |
| profit | String | Current available profit based on quote currency Only applicable to Spot grid |
| 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 Spot grid |
| direction | String | Contract grid type long,short,neutral Only applicable to contract grid |
| basePos | Boolean | Whether or not to open a position when the strategy is activated Only applicable to contract grid |
| sz | String | Used margin based on USDT Only applicable to contract grid |
| lever | String | Leverage Only applicable to contract grid |
| actualLever | String | Actual Leverage Only applicable to contract grid |
| liqPx | String | Estimated liquidation price Only applicable to contract grid |
| uly | String | Underlying Only applicable to contract grid |
| instFamily | String | Instrument family Only applicable to FUTURES/SWAP/OPTION Only applicable to contract grid |
| ordFrozen | String | Margin used by pending orders Only applicable to contract grid |
| availEq | String | Available margin Only applicable to contract grid |
| eq | String | Total equity of strategy account Only applicable to contract grid |
| 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 |
| 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 "" |

### GET / Grid algo sub orders

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/tradingBot/grid/sub-orders

Request Example

GET /api/v5/tradingBot/grid/sub-orders?algoId=123456&type=live&algoOrdType=grid

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoOrdType | String | Yes | Algo order type grid: Spot grid contract\_grid: Contract grid |
| algoId | 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 ordId. |
| before | String | No | Pagination of data to return records newer than the requested ordId. |
| 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": "448965992920907776",

"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** |
| --- | --- | --- |
| 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 |
| 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 Spot and 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 |

### GET / Grid algo order positions

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/tradingBot/grid/positions

Request Example

GET /api/v5/tradingBot/grid/positions?algoId=448965992920907776&algoOrdType=contract\_grid

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoOrdType | String | Yes | Algo order type contract\_grid: Contract grid |
| algoId | 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": "40.907720000000005",

"notionalUsd": "10216.70307",

"pos": "35",

"posSide": "net",

"uTime": "1653400066938",

"upl": "1.674999999999818",

"uplRatio": "0.0008190504784478"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| algoId | 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 |
| ccy | String | Margin currency |
| lever | String | Leverage |
| liqPx | String | Estimated liquidation price |
| posSide | String | Position side net |
| pos | String | Quantity of positions |
| mgnMode | String | Margin mode cross isolated |
| mgnRatio | String | 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 USD |
| adl | String | Auto decrease line, 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: UserID

#### HTTP Request

POST /api/v5/tradingBot/grid/withdraw-income

Request Example

POST /api/v5/tradingBot/grid/withdraw-income

body

{

"algoId":"448965992920907776"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | String | Yes | Algo ID |

Response Example

{

"code":"0",

"msg":"",

"data":[

{

"algoClOrdId": "",

"algoId":"448965992920907776",

"profit":"100"

}

]

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| algoId | String | Algo ID |
| algoClOrdId | String | Client-supplied Algo ID |
| profit | String | Withdraw profit |

### POST / Compute margin balance

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/tradingBot/grid/compute-margin-balance

Request Example

POST /api/v5/tradingBot/grid/compute-margin-balance

body {

"algoId":"123456",

"type":"add",

"amt":"10"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | String | Yes | Algo ID |
| type | String | Yes | Adjust margin balance type add reduce |
| 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: UserID

#### HTTP Request

POST /api/v5/tradingBot/grid/margin-balance

Request Example

POST /api/v5/tradingBot/grid/margin-balance

body {

"algoId":"123456",

"type":"add",

"amt":"10"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | String | Yes | Algo ID |
| type | String | Yes | Adjust margin balance type add reduce |
| amt | String | Conditional | Adjust margin balance amount Either amt or percent is required. |
| percent | String | Conditional | Adjust margin balance percentage |

Response Example

{

"code": "0",

"data": [

{

"algoClOrdId": "",

"algoId": "123456"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| algoId | String | Algo ID |
| algoClOrdId | String | Client-supplied Algo ID |

### POST / Add investment

It is used to add investment and only applicable to contract gird.

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/tradingBot/grid/adjust-investment

Request Example

POST /api/v5/tradingBot/grid/adjust-investment

body

{

"algoId":"448965992920907776",

"amt":"12"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | String | Yes | Algo ID |
| amt | String | Yes | The amount is going to be added |

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

#### HTTP Request

GET /api/v5/tradingBot/grid/ai-param

Request Example

GET /api/v5/tradingBot/grid/ai-param?instId=BTC-USDT&algoOrdType=grid

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoOrdType | String | Yes | Algo order type grid: Spot grid contract\_grid: Contract grid |
| instId | String | Yes | Instrument ID, e.g. BTC-USDT |
| direction | String | Conditional | Contract grid type long,short,neutral Required in the case of contract\_grid |
| duration | String | No | Back testing duration 7D: 7 Days, 30D: 30 Days, 180D: 180 Days The default is 7D for Spot grid Only 7D is available for Contract grid |

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",

"perMaxProfitRate": "0.0733865364573281",

"perMinProfitRate": "0.0561101403446263",

"runType": "1"

}

],

"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 |
| 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 |

### POST / Compute min investment (public)

Authentication is not required for this public endpoint.

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: IP

#### HTTP Request

POST /api/v5/tradingBot/grid/min-investment

Request Example

POST /api/v5/tradingBot/grid/min-investment

body

{

"instId": "ETH-USDT",

"algoOrdType":"grid",

"gridNum": "50",

"maxPx":"5000",

"minPx":"3000",

"runType":"1",

"investmentData":[

{

"amt":"0.01",

"ccy":"ETH"

},

{

"amt":"100",

"ccy":"USDT"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instId | String | Yes | Instrument ID, e.g. BTC-USDT-SWAP |
| 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 stragety,  instant price rsi |
| investmentData | Array of object | 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 object | Minimum invest Data |
| > amt | String | Minimum invest amount |
| > ccy | String | Minimum Invest currency |
| singleAmt | String | Single grid trading amount In terms of spot grid, the unit is quote currency In terms of contract grid, the unit is contract |

### GET / RSI back testing (public)

Authentication is not required for this public endpoint.

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: IP

#### HTTP Request

GET /api/v5/tradingBot/public/rsi-back-testing

Request Example

GET /api/v5/tradingBot/public/rsi-back-testing?instId=BTC-USDT&thold=30&timeframe=3m&timePeriod=14

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instId | String | Yes | Instrument ID, e.g. BTC-USDT Only applicable to SPOT |
| timeframe | String | Yes | K-line type 3m, 5m, 15m, 30m (m: minute) 1H, 4H (H: hour) 1D (D: day) |
| thold | String | Yes | Threshold The value should be an integer between 1 to 100 |
| timePeriod | String | Yes | Time Period 14 |
| triggerCond | String | No | Trigger condition cross\_up cross\_down above below cross Default is cross\_down |
| duration | String | No | Back testing duration 1M (M: month) Default is 1M |

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

#### HTTP Request

GET /api/v5/tradingBot/grid/grid-quantity

Request Example

GET /api/v5/tradingBot/grid/grid-quantity?instId=BTC-USDT-SWAP&runType=1&algoOrdType=contract\_grid&maxPx=70000&minPx=50000&lever=5

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instId | String | Yes | Instrument ID, e.g. BTC-USDT |
| runType | String | Yes | Grid type 1: Arithmetic 2: Geometric |
| 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 |
| 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.

#### URL Path

/ws/v5/business (required login)

Request Example

{

"op": "subscribe",

"args": [{

"channel": "grid-orders-spot",

"instType": "SPOT"

}]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name grid-orders-spot |
| > instType | String | Yes | Instrument type SPOT ANY |
| > instId | String | No | Instrument ID |
| > algoId | String | No | Algo Order ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "grid-orders-spot",

"instType": "ANY"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"grid-orders-spot\", \"instType\" : \"FUTURES\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name |
| > instType | String | Yes | Instrument type |
| > instId | String | No | Instrument ID |
| > algoId | 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",

"maxPx": "5000",

"minPx": "400",

"pTime": "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"

}]

}

#### 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 | 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 |
| > algoOrdType | String | Algo order type grid: Spot grid |
| > state | String | Algo order state starting running stopping stopped |
| > rebateTrans | Array of object | Rebate transfer info |
| >> rebate | String | Rebate amount |
| >> rebateCcy | String | Rebate currency |
| > triggerParams | Array of object | 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 Contract grid 1: Market Close All positions 2: Keep positions This field is only valid when triggerAction is stop |
| > 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 |
| > 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 0: None 1: Manual stop 2: Take profit 3: Stop loss 4: Risk control 5: Delivery 6: Signal |
| > stopType | String | Stop type 1: Sell base currency 2: Keep base currency |
| > quoteSz | String | Quote currency investment amount Only applicable to Spot grid |
| > baseSz | String | Base currency investment amount Only applicable to Spot grid |
| > curQuoteSz | String | Assets of quote currency currently held Only applicable to Spot grid |
| > curBaseSz | String | Assets of base currency currently held Only applicable to Spot grid |
| > profit | String | Current available profit based on quote currency Only applicable to Spot grid |
| > 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 Spot grid |
| > 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 |

### 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

{

"op": "subscribe",

"args": [{

"channel": "grid-orders-contract",

"instType": "SWAP"

}]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | 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 |
| > algoId | String | No | Algo Order ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "grid-orders-contract",

"instType": "ANY"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"grid-orders-contract\", \"instType\" : \"FUTURES\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name |
| > instType | String | Yes | Instrument type |
| > instId | String | No | Instrument ID |
| > algoId | 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",

"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 | 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 |
| > algoOrdType | String | Algo order type contract\_grid: Contract grid |
| > state | String | Algo order state starting running stopping no\_close\_position: stopped algo order but hadn't close position yet stopped |
| > rebateTrans | Array of object | Rebate transfer info |
| >> rebate | String | Rebate amount |
| >> rebateCcy | String | Rebate currency |
| > triggerParams | Array of object | 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 Contract grid 1: Market Close All positions 2: Keep positions This field is only valid when triggerAction is stop |
| > 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 |
| > 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 | 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 0: None 1: Manual stop 2: Take profit 3: Stop loss 4: Risk control 5: Delivery 6: 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 contract grid |
| > basePos | Boolean | Whether or not to open a position when the strategy is activated Only applicable to contract grid |
| > sz | String | Used margin based on USDT Only applicable to contract grid |
| > lever | String | Leverage Only applicable to contract grid |
| > actualLever | String | Actual Leverage Only applicable to contract grid |
| > liqPx | String | Estimated liquidation price Only applicable to contract grid |
| > ordFrozen | String | Margin used by pending orders Only applicable to contract grid |
| > availEq | String | Available margin Only applicable to contract grid |
| > eq | String | Total equity of strategy account Only applicable to contract grid |
| > 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 |
| > 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

{

"op": "subscribe",

"args": [{

"channel": "grid-positions",

"algoId": "449327675342323712"

}]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name grid-positions |
| > algoId | String | Yes | Algo Order ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "grid-positions",

"algoId": "449327675342323712"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"grid-positions\", \"instType\" : \"FUTURES\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name |
| > algoId | 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": "29181.4638888888888895",

"cTime": "1653400065917",

"ccy": "USDT",

"imr": "2089.2690000000002",

"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 | 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 |
| > pos | String | Quantity of positions |
| > mgnMode | String | Margin mode cross isolated |
| > mgnRatio | String | 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 USD |
| > adl | String | Auto decrease line, 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. 1597026383085 |

### 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

{

"op": "subscribe",

"args": [{

"channel": "grid-sub-orders",

"algoId": "449327675342323712"

}]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name grid-sub-orders |
| > algoId | String | Yes | Algo Order ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "grid-sub-orders",

"algoId": "449327675342323712"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"grid-sub-orders\", \"instType\" : \"FUTURES\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name |
| > algoId | 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-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",

"lever": "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 | 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 |
| > 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 |
| > tag | String | Order tag |
| > 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/OPTION |
| > lever | String | Leverage |
| > pTime | String | Push time of orders information, Unix timestamp format in milliseconds, e.g. 1597026383085 |

## Signal bot trading

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](https://www.okx.com/learn/signal-trading)

### POST / Create signal

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/tradingBot/signal/create-signal

Request Example

POST /api/v5/tradingBot/signal/create-signal

body

{

"signalChanName": "long short",

"signalDesc": "this is the first version"

}

#### 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",

"signalToken":"dojuckew331lkx"

}

],

"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: UserID

#### HTTP Request

GET /api/v5/tradingBot/signal/signals

Request Example

GET /api/v5/tradingBot/signal/signals?signalSourceType=1

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| signalSourceType | String | Yes | Signal source type 1: Created by yourself 2: Subscribe 3: Free signal |
| signalChanId | String | No | Signal channel id |
| after | String | No | Pagination of data to return records signalChanId earlier than the requested timestamp, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| before | String | No | Pagination of data to return records signalChanId newer than the requested 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": [

{

"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: UserID

#### HTTP Request

POST /api/v5/tradingBot/signal/order-algo

Request Example

# Create signal bot

POST /api/v5/tradingBot/signal/order-algo

body

{

"signalChanId": "627921182788161536",

"instIds": [

"BTC-USDT-SWAP",

"ETH-USDT-SWAP",

"LTC-USDT-SWAP"

],

"lever": "10",

"investAmt": "100",

"subOrdType": "9",

"entrySettingParam": {

"allowMultipleEntry": true,

"entryType": "1",

"amt": "",

"ratio": ""

},

"exitSettingParam": {

"tpSlType": "2",

"tpPct": "",

"slPct": ""

}

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| signalChanId | String | Yes | Signal channel Id |
| lever | String | Yes | Leverage Only applicable to contract signal |
| 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 false. true: include false : exclude |
| instIds | String | No | Instrument IDs. Single currency or multiple currencies separated with comma. When includeAll is true, 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 subOrdType is limit 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 true。 true：Allow false：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 entryType in 2/3 |
| > ratio | Array of object | No | Amount ratio per order Only applicable to entryType in 4/5 |
| exitSettingParam | String | No | Exit setting |
| > tpSlType | String | 是 | Type of set the take-profit and stop-loss trigger price  pnl: 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 | 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** |
| --- | --- | --- |
| algoId | 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: UserID

#### HTTP Request

POST /api/v5/tradingBot/signal/stop-order-algo

Request Example

POST /api/v5/tradingBot/signal/stop-order-algo

body

[

{

"algoId":"448965992920907776"

}

]

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | String | Yes | Algo ID |

Response Example

{

"code": "0",

"data": [

{

"algoId": "448965992920907776",

"sCode": "0",

"sMsg": "",

"algoClOrdId": ""

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| algoId | String | Algo ID |
| sCode | String | The code of the event execution result, 0 means success. |
| sMsg | String | Rejection or success message of event execution. |
| algoClOrdId | String | Client-supplied Algo ID |

### POST / Adjust margin balance

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/tradingBot/signal/margin-balance

Request Example

POST /api/v5/tradingBot/signal/margin-balance

body

{

"algoId":"123456",

"type":"add",

"amt":"10"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | String | Yes | Algo ID |
| type | String | Yes | Adjust margin balance type add reduce |
| amt | String | Yes | Adjust margin balance amount Either amt or percent is required. |
| allowReinvest | Boolean | No | Whether to reinvest with newly added margin. The default value is false.  false:it will be used as passive margin to prevent liquidation and will not be used as active investment true: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** |
| --- | --- | --- |
| algoId | String | Algo ID |

### POST / Amend TPSL

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/tradingBot/signal/amendTPSL

Request Example

POST /api/v5/tradingBot/signal/amendTPSL

body

{

"algoId": "637039348240277504",

"exitSettingParam": {

"tpSlType": "pnl",

"tpPct": "0.01",

"slPct": "0.01"

}

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | String | Yes | Algo ID |
| exitSettingParam | String | Yes | Exit setting |
| > tpSlType | String | Yes | Type of set the take-profit and stop-loss trigger price pnl: 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 | No | Take-profit percentage |
| > slPct | String | No | Stop-loss percentage |

Response Example

{

"code": "0",

"data": [

{

"algoId": "637039348240277504"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| algoId | String | Algo ID |

### POST / Set instruments

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/tradingBot/signal/set-instruments

Request Example

POST /api/v5/tradingBot/signal/set-instruments

body

{

"algoId": "637039348240277504",

"instIds": [

"SHIB-USDT-SWAP",

"ETH-USDT-SWAP"

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | String | Yes | Algo ID |
| instIds | Array | Yes | Instrument IDs. When includeAll is true, it is ignored |
| includeAll | Boolean | Yes | Whether to include all USDT-margined contract.The default value is false. true: include false : exclude |

Response Example

{

"code": "0",

"data": [

{

"algoId": "637039348240277504"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| algoId | String | Algo ID |

### GET / Signal bot order details

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/tradingBot/signal/orders-algo-details

Request Example

GET /api/v5/tradingBot/signal/orders-algo-details?algoId=623833708424069120&algoOrdType=contract

#### 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": [

{

"algoId": "623833708424069120",

"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",

"signalChanId": "623827579484770304",

"signalChanName": "testing",

"signalSourceType": "1",

"state": "running",

"subOrdType": "9",

"totalEq": "26.824296901312227",

"totalPnl": "-73.1757030986877733",

"totalPnlRatio": "-0.7317570309868777",

"uTime": "1697029422313"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| algoId | String | Algo ID |
| algoClOrdId | String | Client-supplied Algo ID |
| instType | String | Instrument type |
| instIds | Array of string | 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 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 |
| > tpSlType | String | Type of set the take-profit and stop-loss trigger price pnl: 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: UserID

#### HTTP Request

GET /api/v5/tradingBot/signal/orders-algo-pending

Request Example

GET /api/v5/tradingBot/signal/orders-algo-pending?algoOrdType=contract

#### 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 algoId earlier than the requested timestamp, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| before | String | No | Pagination of data to return records algoId newer than the requested 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": [

{

"algoId": "623833708424069120",

"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",

"signalChanId": "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** |
| --- | --- | --- |
| algoId | String | Algo ID |
| algoClOrdId | String | Client-supplied Algo ID |
| instType | String | Instrument type |
| instIds | Array of string | 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 starting running stopping |
| cancelType | String | Algo order stop reason 0: 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 |
| 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 |
| > tpSlType | String | Type of set the take-profit and stop-loss trigger price pnl: 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 history

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/tradingBot/signal/orders-algo-history

Request Example

GET /api/v5/tradingBot/signal/orders-algo-history?algoId=623833708424069120&algoOrdType=contract

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoOrdType | String | Yes | Algo order type contract: Contract signal |
| algoId | String | Yes | Algo ID |
| after | String | Yes | Pagination of data to return records algoId earlier than the requested timestamp, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| before | String | No | Pagination of data to return records algoId newer than the requested 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": [

{

"algoId": "623833708424069120",

"algoOrdType": "contract",

"availBal": "1.6561369013122267",

"cTime": "1695005546360",

"cancelType": "1",

"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",

"signalChanId": "623827579484770304",

"signalChanName": "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** |
| --- | --- | --- |
| algoId | String | Algo ID |
| algoClOrdId | String | Client-supplied Algo ID |
| instType | String | Instrument type |
| instIds | Array of string | 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 |
| > tpSlType | String | Type of set the take-profit and stop-loss trigger price pnl: 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: UserID

#### HTTP Request

GET /api/v5/tradingBot/signal/positions

Request Example

GET /api/v5/tradingBot/signal/positions?algoId=623833708424069120&algoOrdType=contract

#### 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": [

{

"adl": "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 |
| algoClOrdId | String | Client-supplied Algo ID. Used to be extended in the future. |
| 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 |
| ccy | String | Margin currency |
| lever | String | Leverage |
| liqPx | String | Estimated liquidation price |
| posSide | String | Position side net |
| pos | String | Quantity of positions |
| mgnMode | String | Margin mode cross isolated |
| mgnRatio | String | 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 USD |
| adl | String | Auto decrease line, 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: 1 request per 10 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/tradingBot/signal/positions-history

Request Example

GET /api/v5/tradingBot/signal/positions-history?algoId=1234

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | String | Yes | Algo ID |
| instId | String | No | Instrument ID, e.g.：BTC-USD-SWAP |
| after | String | No | Pagination of data to return records earlier than the requested uTime, Unix timestamp format in milliseconds, e.g.1597026383085 |
| before | String | No | Pagination of data to return records newer than the requested uTime, 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": [

{

"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 cross isolated |
| 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: long short |
| 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: UserID

#### HTTP Request

POST /api/v5/tradingBot/signal/close-position

Request Example

POST /api/v5/tradingBot/signal/close-position

body

{

"instId":"BTC-USDT-SWAP",

"algoId":"448965992920907776"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | String | Yes | Algo ID |
| instId | String | Yes | Instrument ID |

Response Example

{

"code": "0",

"data": [

{

"algoId": "448965992920907776"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| algoId | 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: UserID

#### HTTP Request

POST /api/v5/tradingBot/signal/sub-order

Request Example

POST /api/v5/tradingBot/signal/sub-order

body

{

"algoId"："1222",

"instId":"BTC-USDT-SWAP",

"side":"buy",

"ordType":"limit",

"px":"2.15",

"sz":"2"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instId | String | Yes | Instrument ID, e.g. BTC-USDT-SWAP |
| algoId | String | Yes | Algo ID |
| side | String | Yes | Order side, buy sell |
| 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 Spot and 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 | String | 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 `Spot and futures mode` and `Multi-currency margin`

### POST / Cancel sub order

Cancel an incomplete order.

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/tradingBot/signal/cancel-sub-order

Request Example

POST /api/v5/tradingBot/signal/cancel-sub-order

body

{

"algoId":"91664",

"signalOrdId":"590908157585625111",

"instId":"BTC-USDT-SWAP"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | 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 | String | 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: UserID

#### HTTP Request

GET /api/v5/tradingBot/signal/sub-orders

Request Example

# Get historical filled sub orders

GET /api/v5/tradingBot/signal/sub-orders?algoId=623833708424069120&algoOrdType=contract&state=filled

# Get designated sub order

GET /api/v5/tradingBot/signal/sub-orders?algoId=623833708424069120&algoOrdType=contract&signalOrdId=O632302662327996418

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | String | Yes | Algo ID |
| algoOrdType | String | Yes | Algo order type contract: Contract signal |
| state | String | Conditional | Sub order state live partially\_filled filled cancelled Either state or signalOrdId is required, if both are passed in, only state is valid. |
| signalOrdId | String | Conditional | Sub order ID |
| after | String | No | Pagination of data to return records earlier than the requested ordId |
| before | String | No | Pagination of data to return records newer than the requested ordId. |
| begin | String | No | Return records of ctime after than the requested timestamp (include), Unix timestamp format in milliseconds, e.g. 1597026383085 |
| end | String | No | Return records of ctime before than the requested timestamp (include), Unix timestamp format in milliseconds, e.g. 1597026383085 |
| limit | String | No | Number of results per request. The maximum is 100. The default is 100. |
| type | String | No | Sub order type  live filled Either type or clOrdId is required, if both are passed in, only clOrdId is valid. |
| clOrdId | String | No | Sub order client-supplied ID.  It will be deprecated soon |

Response Example

{

"code": "0",

"data": [

{

"accFillSz": "18",

"algoClOrdId": "",

"algoId": "623833708424069120",

"algoOrdType": "contract",

"avgPx": "1572.81",

"cTime": "1697024702320",

"ccy": "",

"clOrdId": "O632302662327996418",

"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": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| algoId | String | Algo ID |
| algoClOrdId | 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 Spot and 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 |
| 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 |

### GET / Signal bot event history

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/tradingBot/signal/event-history

Request Example

GET /api/v5/tradingBot/signal/event-history?algoId=623833708424069120

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | String | Yes | Algo ID |
| after | String | No | Pagination of data to return records eventCtime earlier than the requested timestamp, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| before | String | No | Pagination of data to return records eventCtime newer than the requested 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": [

{

"alertMsg": "{\"marketPosition\":\"short\",\"prevMarketPosition\":\"long\",\"action\":\"sell\",\"instrument\":\"ETHUSDT.P\",\"timestamp\":\"2023-10-16T10:50:00.000Z\",\"maxLag\":\"60\",\"investmentType\":\"base\",\"amount\":\"2\"}",

"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": [

{

"clOrdId": "O634100754731765763"

},

{

"clOrdId": "O634100754752737282"

}

]

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| algoId | String | Algo ID |
| eventType | String | Event type system\_action user\_action signal\_processing |
| eventCtime | String | Event timestamp of creation. Unix timestamp format in milliseconds, e.g. 1597026383085 |
| eventUtime | String | Event timestamp of update. Unix timestamp format in milliseconds, e.g. 1597026383085 |
| eventProcessMsg | String | Event process message |
| eventStatus | String | Event status success failure |
| triggeredOrdData | Array of object | 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](https://www.okx.com/help/vii-recurring-buy)  
The API endpoints of Recurring buy require authentication.

### POST / Place recurring buy order

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/tradingBot/recurring/order-algo

Request Example

POST /api/v5/tradingBot/recurring/order-algo

body

{

"stgyName": "BTC|ETH recurring buy monthly",

"amt":"100",

"recurringList":[

{

"ccy":"BTC",

"ratio":"0.2"

},

{

"ccy":"ETH",

"ratio":"0.8"

}

],

"period":"monthly",

"recurringDay":"1",

"recurringTime":"0",

"timeZone":"8", // UTC +8

"tdMode":"cross",

"investmentCcy":"USDT"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| stgyName | String | Yes | Custom name for trading bot, no more than 40 characters |
| recurringList | Array of object | Yes | Recurring buy info |
| > ccy | String | Yes | Recurring currency, e.g. BTC |
| > ratio | String | Yes | Proportion of recurring currency assets, e.g. "0.2" representing 20% |
| period | String | Yes | Period monthly weekly daily hourly |
| recurringDay | String | Conditional | Recurring buy date When the period is monthly, the value range is an integer of [1,28] When the period is weekly, the value range is an integer of [1,7] When the period is daily/hourly, the parameter is not required. |
| recurringHour | String | Conditional | Recurring buy by hourly 1/4/8/12, e.g. 4 represents "recurring buy every 4 hour" When the period is hourly, the parameter is required. |
| recurringTime | String | Yes | Recurring buy time, the value range is an integer of [0,23] When the period is hourly, 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 USDT/USDC |
| tdMode | String | Yes | Trading mode Margin mode: cross Non-Margin mode: cash |
| 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. |

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: UserID

#### HTTP Request

POST /api/v5/tradingBot/recurring/amend-order-algo

Request Example

POST /api/v5/tradingBot/recurring/amend-order-algo

body

{

"algoId":"448965992920907776",

"stgyName":"stg1"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | 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** |
| --- | --- | --- |
| 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 |

### 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: UserID

#### HTTP Request

POST /api/v5/tradingBot/recurring/stop-order-algo

Request Example

POST /api/v5/tradingBot/recurring/stop-order-algo

body

[

{

"algoId":"560472804207104000"

}

]

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| algoId | String | Yes | Algo ID |

Response Example

{

"code": "0",

"data": [

{

"algoClOrdId": "",

"algoId": "1839309556514557952",

"sCode": "0",

"sMsg": "",

"tag": ""

}

],

"msg": ""

}

#### 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 | (Deprecated) |

### GET / Recurring buy order list

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/tradingBot/recurring/orders-algo-pending

Request Example

GET /api/v5/tradingBot/recurring/orders-algo-pending

#### 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": "644497312047435776",

"algoOrdType": "recurring",

"amt": "100",

"cTime": "1699932133373",

"cycles": "6",

"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": "12",

"state": "running",

"stgyName": "stg1",

"tag": "",

"timeZone": "8",

"totalAnnRate": "0",

"totalPnl": "0",

"uTime": "1699952473152"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| algoId | 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 pause |
| stgyName | String | Custom name for trading bot, no more than 40 characters |
| recurringList | Array of object | Recurring buy info |
| > ccy | String | Recurring currency, e.g. BTC |
| > ratio | String | Proportion of recurring currency assets, e.g. "0.2" representing 20% |
| period | String | Period monthly weekly daily hourly |
| recurringDay | String | Recurring buy date When the period is monthly, the value range is an integer of [1,28] When the period is weekly, the value range is an integer of [1,7] |
| 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 |

### GET / Recurring buy order history

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/tradingBot/recurring/orders-algo-history

Request Example

GET /api/v5/tradingBot/recurring/orders-algo-history

#### 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",

"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"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| algoId | 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 stopped |
| stgyName | String | Custom name for trading bot, no more than 40 characters |
| recurringList | Array of object | Recurring buy info |
| > ccy | String | Recurring currency, e.g. BTC |
| > ratio | String | Proportion of recurring currency assets, e.g. "0.2" representing 20% |
| period | String | Period monthly weekly daily hourly |
| recurringDay | String | Recurring buy date When the period is monthly, the value range is an integer of [1,28] When the period is weekly, the value range is an integer of [1,7] |
| 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 |

### GET / Recurring buy order details

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/tradingBot/recurring/orders-algo-details

Request Example

GET /api/v5/tradingBot/recurring/orders-algo-details?algoId=644497312047435776

#### 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"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| algoId | 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 object | 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 |
| > px | String | Current market price, quote currency is investmentCcy |
| period | String | Period monthly weekly daily hourly |
| recurringDay | String | Recurring buy date When the period is monthly, the value range is an integer of [1,28] When the period is weekly, the value range is an integer of [1,7] |
| 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 |

### GET / Recurring buy sub orders

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/tradingBot/recurring/sub-orders

Request Example

GET /api/v5/tradingBot/recurring/sub-orders?algoId=560516615079727104

#### 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 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": [

{

"accFillSz": "0.045315",

"algoClOrdId": "",

"algoId": "560516615079727104",

"algoOrdType": "recurring",

"avgPx": "1765.4",

"cTime": "1679911222200",

"fee": "-0.0000317205",

"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": "-0.000000500682",

"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 recurring: recurring buy |
| 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 Non-Margin mode : cash |
| ordType | String | Sub order type market: Market 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 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 |

### 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

{

"op": "subscribe",

"args": [{

"channel": "algo-recurring-buy",

"instType": "SPOT"

}]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name algo-recurring-buy |
| > instType | String | Yes | Instrument type SPOT ANY |
| > algoId | String | No | Algo Order ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "algo-recurring-buy",

"instType": "SPOT"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"algo-recurring-buy\", \"instType\" : \"FUTURES\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name |
| > instType | String | Yes | Instrument type |
| > algoId | 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": "1699933314691",

"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"

}]

}

#### Response parameters when data is pushed.

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| arg | Object | Successfully subscribed channel |
| > channel | String | Channel name |
| > instType | String | Instrument type |
| > algoId | String | Algo Order ID |
| > uid | String | User ID |
| data | Array | Subscribed data |
| > algoId | 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 object | 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 |
| >> px | String | Current market price, quote currency is investmentCcy |
| > period | String | Period monthly weekly daily hourly |
| > recurringDay | String | Recurring buy date When the period is monthly, the value range is an integer of [1,28] When the period is weekly, the value range is an integer of [1,7] |
| > 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 |

## Copy Trading

Copy trading function is not opened.  
  
Lead trading API Workflow as follows:

**1. Apply to become a leading trader.**

* The procedure can refer to [How to become a lead trader](https://www.okx.com/help/11639154398221);
* You can know whether you are a lead trader by checking whether roleType or spotRoleType from [Get account configuration](https://www.okx.com/docs-v5/en/#trading-account-rest-api-get-account-configuration) is 1.

**2. Leading instruments:**

* [GET / Leading instruments](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-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 disenabled copy trading, you can still trade normally, but copy trading will not be triggered;
* [Amend leading instruments](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-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](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-place-order) endpoint, [Place multiple orders](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-place-multiple-orders) endpoint, [Place order channel](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-place-order), [Place multiple orders channel](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-place-multiple-orders), 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](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-place-order) endpoint, [Place multiple orders](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-place-multiple-orders) endpoint, [Place order channel](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-place-order), [Place multiple orders channel](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-place-multiple-orders), or close the position by [Close positions](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-close-positions) / [Close lead or copy position](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-post-close-lead-or-copy-position);
* [Close positions](https://www.okx.com/docs-v5/en/#order-book-trading-trade-post-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 or copy position](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-post-close-lead-or-copy-position) can only close a leading position once a time. It is required to pass subPosId which can get from [Get existing leading positions](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-existing-lead-or-copy-positions).

**5. TP/SL**

* TP/SL can be set by [Place algo order](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-mass-cancel-order) or [Place lead or copy stop order](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-post-place-lead-or-copy-stop-order);
* [Place algo order](https://www.okx.com/docs-v5/en/#order-book-trading-trade-ws-mass-cancel-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 or copy stop order](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-post-place-lead-or-copy-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](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-existing-lead-or-copy-positions).

### GET / Existing lead or copy positions

Retrieve lead or copy positions that are not closed.

Returns reverse chronological order with openTime

#### Rate limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

GET /api/v5/copytrading/current-subpositions

Request example

GET /api/v5/copytrading/current-subpositions?instId=BTC-USDT-SWAP

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instType | String | No | Instrument type SPOT SWAP It returns all types by default. |
| instId | String | No | Instrument ID, e.g. BTC-USDT-SWAP |
| uniqueCode | String | No | Unique code, only applicable to copy trading A combination of case-sensitive alphanumerics, all numbers and the length is 16 characters, e.g. 213E8C92DC61EFAC |
| subPosType | String | No | Data type. lead: lead trading, the default value copy: copy trading |
| after | String | No | Pagination of data to return records earlier than the requested subPosId. |
| before | String | No | Pagination of data to return records newer than the requested subPosId. |
| 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",

"slOrdPx": "",

"slTriggerPx": "",

"subPos": "1",

"subPosId": "649945658862370816",

"tpOrdPx": "",

"tpTriggerPx": "",

"uniqueCode": "25CD5A80241D6FE6",

"upl": "0.2807",

"uplRatio": "0.0222042921442527",

"availSubPos": "1"

},

{

"algoId": "",

"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 or copy position ID |
| posSide | String | Position side long  short  net (Long positions have positive subPos; short positions have negative subPos) |
| mgnMode | String | Margin mode. cross 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. |
| algoId | 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 |
| 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 or copy position history

Retrieve the completed lead or copy position of the last 3 months.  
Returns reverse chronological order with subPosId.

#### Rate limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

GET /api/v5/copytrading/subpositions-history

Request example

GET /api/v5/copytrading/subpositions-history?instId=BTC-USDT-SWAP

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instType | String | No | Instrument type SPOT SWAP It returns all types by default. |
| instId | String | No | Instrument ID, e.g. BTC-USDT-SWAP |
| subPosType | String | No | Data type. lead: lead trading, the default value copy: copy trading |
| after | String | No | Pagination of data to return records earlier than the requested subPosId. |
| before | String | No | Pagination of data to return records newer than the requested subPosId. |
| 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",

"openOrdId": "",

"openTime": "1701184638702",

"pnl": "0.6225",

"pnlRatio": "0.0166399358460306",

"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": "25CD5A80241D6FE6"

}

],

"msg": ""

}

#### Response parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| instId | String | Instrument ID, e.g. BTC-USDT-SWAP |
| subPosId | String | Lead or copy position ID |
| posSide | String | Position side long  short  net (long position has positive subPos; short position has negative subPos) |
| mgnMode | String | Margin mode. cross isolated |
| 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 |
| 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 |
| 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 or copy stop order

Set TP/SL for the current lead or copy position that are not closed.

#### Rate limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### 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 SPOT SWAP, the default value |
| subPosId | String | Yes | Lead or copy 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 -1 Only applicable to SPOT 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 -1 Only applicable to SPOT lead trader |
| tpTriggerPxType | String | No | Take-profit trigger price type   last: last price index: index price mark: mark price  Default is last |
| slTriggerPxType | String | No | Stop-loss trigger price type last: last price  index: index price  mark: mark price  Default is last |
| tag | String | No | Order tag A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 16 characters. |
| subPosType | String | No | Data type. lead: lead trading, the default value copy: copy trading |

Response example

{

"code": "0",

"data": [

{

"subPosId": "518560559046594560",

"tag":""

}

],

"msg": ""

}

#### Response parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| subPosId | String | Lead or copy position ID |
| tag | String | Order tag |

### POST / Close lead or copy position

You can only close a lead or copy position once a time.   
It is required to pass subPosId which can get from [Get existing leading positions](https://www.okx.com/docs-v5/en/#order-book-trading-copy-trading-get-existing-lead-or-copy-positions).

#### Rate limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

#### 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 | Instrument type SPOT SWAP, the default value |
| subPosType | String | No | Data type. lead: lead trading, the default value copy: copy trading |
| subPosId | String | Yes | Lead or copy position ID |
| ordType | String | No | Order type market：Market order, the default value limit：Limit order |
| px | String | No | Order price. Only applicable to limit order and SPOT lead trader  If the price is 0, the pending order will be canceled.  It is modifying order if you set px 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 or copy 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: UserID

#### HTTP request

GET /api/v5/copytrading/instruments

Request example

GET /api/v5/copytrading/instruments

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instType | String | No | Instrument type SPOT SWAP, 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. true or false |

### 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: UserID

#### 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 SPOT SWAP, 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": 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: UserID

#### HTTP request

GET /api/v5/copytrading/profit-sharing-details

Request example

GET /api/v5/copytrading/profit-sharing-details?limit=2

#### 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": "1723392000000",

"instType": "SWAP"

},

{

"ccy": "USDT",

"nickName": "Apple",

"profitSharingAmt": "0.00336",

"profitSharingId": "20",

"portLink": "",

"ts": "1723392000000",

"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: UserID

#### HTTP request

GET /api/v5/copytrading/total-profit-sharing

Request example

GET /api/v5/copytrading/total-profit-sharing

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instType | String | No | Instrument type SPOT 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: UserID

#### HTTP request

GET /api/v5/copytrading/unrealized-profit-sharing-details

Request example

GET /api/v5/copytrading/unrealized-profit-sharing-details

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instType | String | No | Instrument type SPOT 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: UserID

#### HTTP request

GET /api/v5/copytrading/total-unrealized-profit-sharing

Request example

GET /api/v5/copytrading/total-unrealized-profit-sharing

#### 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 / Apply for lead trading

Only ND broker sub-account whitelisted can apply for lead trader by this endpoint. It will be passed immediately.  
Please reach out to BD for help if you want to be whitelisted.  
For other accounts, e.g. ND main accounts and general main and sub-accounts, still need to apply on the web manually

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

POST /api/v5/copytrading/apply-lead-trading

Request example

POST /api/v5/copytrading/apply-lead-trading

{

"instType": "SWAP",

"instId": "BTC-USDT-SWAP",

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instType | String | No | Instrument type SWAP |
| instId | String | Yes | The lead instrument set at the first time. e.g. BTC-USDT-SWAP. If there are multiple instruments, separate them with commas. |

Response example

{

"code": "0",

"data": [

{

"result": true

}

],

"msg": ""

}

#### Response parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| result | Boolean | The result of setting  true |

### POST / Stop lead trading

It is used to stop lead trading for ND broker sub-account.

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

POST /api/v5/copytrading/stop-lead-trading

Request example

POST /api/v5/copytrading/stop-lead-trading

body

{

"instType": "SWAP",

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instType | String | No | Instrument type SWAP |

Response example

{

"code": "0",

"data": [

{

"result": true

}

],

"msg": ""

}

#### Response parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| result | Boolean | The result of setting  true |

### POST / Amend profit sharing ratio

It is used to amend profit sharing ratio.

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

POST /api/v5/copytrading/amend-profit-sharing-ratio

Request example

POST /api/v5/copytrading/amend-profit-sharing-ratio

body

{

"instType": "SWAP",

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instType | String | No | Instrument type SWAP |
| 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: UserID

#### HTTP request

GET /api/v5/copytrading/config

Request example

GET /api/v5/copytrading/config

#### 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 | String | Details |
| > instType | String | Instrument type SPOT SWAP |
| > roleType | String | Role type 0: General user 1: Leading trader 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: UserID

#### 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 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 |
| copyMgnMode | String | Yes | Copy margin mode cross: cross isolated: isolated copy: Use the same margin mode as lead trader when opening positions |
| copyInstIdType | String | Yes | 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 |
| instId | String | Conditional | Instrument ID.  If there are multiple instruments, separate them with commas. |
| copyMode | String | No | Copy mode fixed\_amount: set the same fixed amount for each order, and copyAmt is required； ratio\_copy: set amount as a multiple of the lead trader’s order value, and copyRatio is required  The default is fixed\_amount |
| 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 market\_close: immediately close at market price copy\_close：close when trader closes manual\_close: close manually The default is copy\_close |

Response example

{

"code": "0",

"data": [

{

"result": true

}

],

"msg": ""

}

#### Response parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| result | Boolean | The result of setting  true |

### POST / Amend copy settings

You need to use this endpoint to amend copy settings

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### 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 SWAP |
| 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 cross: cross isolated: isolated copy: Use the same margin mode as lead trader when opening positions |
| copyInstIdType | String | Yes | 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 |
| instId | String | Conditional | Instrument ID.  If there are multiple instruments, separate them with commas. |
| copyMode | String | No | Copy mode fixed\_amount: set the same fixed amount for each order, and copyAmt is required； ratio\_copy: set amount as a multiple of the lead trader’s order value, and copyRatio is required  The default is fixed\_amount |
| 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 market\_close: immediately close at market price copy\_close：close when trader closes manual\_close: close manually The default is copy\_close |

Response example

{

"code": "0",

"data": [

{

"result": true

}

],

"msg": ""

}

#### Response parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| result | Boolean | The result of setting  true |

### POST / Stop copying

You need to use this endpoint to stop copy trading

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### 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 SWAP |
| uniqueCode | String | Yes | Lead trader unique code A combination of case-sensitive alphanumerics, all numbers and the length is 16 characters, e.g. 213E8C92DC61EFAC |
| subPosCloseType | String | Yes | Action type for open positions, it is required if you have related copy position market\_close: immediately close at market price copy\_close：close when trader closes manual\_close: close manually |

Response example

{

"code": "0",

"data": [

{

"result": true

}

],

"msg": ""

}

#### Response parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| result | Boolean | The result of setting  true |

### GET / Copy settings

Retrieve the copy settings about certain lead trader.

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

GET /api/v5/copytrading/copy-settings

Request example

GET /api/v5/copytrading/copy-settings?instType=SWAP&uniqueCode=25CD5A80241D6FE6

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instType | String | No | Instrument type SWAP |
| 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": "ADA-USDT-SWAP"

},

{

"enabled": "1",

"instId": "YFII-USDT-SWAP"

}

],

"slRatio": "",

"slTotalAmt": "",

"subPosCloseType": "copy\_close",

"tpRatio": ""

}

],

"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 | 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 |

### GET / Multiple leverages

Retrieve leverages that belong to the lead trader and you.

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

GET /api/v5/copytrading/batch-leverage-info

Request example

GET /api/v5/copytrading/batch-leverage-info?mgnMode=isolated&uniqueCode=25CD5A80241D6FE6

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| mgnMode | String | Yes | Margin mode cross isolated |
| uniqueCode | String | Yes | Lead trader unique code A combination of case-sensitive alphanumerics, all numbers and the length is 16 characters, e.g. 213E8C92DC61EFAC |
| instId | String | No | Instrument ID.  If there are multiple instruments, separate them with commas. |

Response example

{

"code": "0",

"data": [

{

"instId": "ETC-USDT-SWAP",

"leadTraderLevers": [

{

"lever": "3",

"posSide": "long"

},

{

"lever": "3",

"posSide": "short"

}

],

"mgnMode": "isolated",

"myLevers": [

{

"lever": "3",

"posSide": "long"

},

{

"lever": "3",

"posSide": "short"

}

]

}

],

"msg": ""

}

#### Response parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| instId | String | Instrument ID |
| mgnMode | String | Margin mode isolated cross |
| leadTraderLevers | Array | Lead trader leverages |
| > lever | String | leverage |
| > posSide | String | Position side |
| myLevers | Array | My leverages |
| > lever | String | leverage |
| > posSide | String | Position side |

### POST / Set Multiple leverages

Set Multiple leverages  
Only applicable to copy trader

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

POST /api/v5/copytrading/batch-set-leverage

Request example

POST /api/v5/copytrading/batch-set-leverage

body

{

"instId": "BTC-USDT-SWAP",

"mgnMode": "isolated",

"lever": "5"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| mgnMode | String | Yes | Margin mode cross isolated |
| lever | String | Yes | Leverage |
| instId | String | Yes | Instrument ID.  If there are multiple instruments, separate them with commas. |

Response example

{

"code": "0",

"data": [

{

"failInstId": "",

"result": "0",

"succInstId": "BTC-USDT-SWAP"

}

],

"msg": ""

}

#### Response parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| succInstId | String | Instrument ID setted successfully |
| failInstId | String | Instrument ID setted unsuccessfully |
| result | String | Result.  0：All success 1:Some successes 2: All fail |

### GET / My lead traders

Retrieve my lead traders.

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

GET /api/v5/copytrading/current-lead-traders

Request example

GET /api/v5/copytrading/current-lead-traders?instType=SWAP

#### 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 |
| leadMode | String | Lead mode public private |

### GET / My history lead traders

Retrieve my history lead traders.

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

GET /api/v5/copytrading/lead-traders-history

Request example

GET /api/v5/copytrading/lead-traders-history?instType=SWAP

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instType | String | No | Instrument type SWAP, the default value |
| after | String | No | Pagination of data to return records earlier than the requested copyRelId. |
| before | String | No | Pagination of data to return records newer than the requested copyRelId. |
| limit | String | No | Number of results per request. Maximum is 100. Default is 100. |

Response example

{

"code": "0",

"data": [

{

"beginCopyTime": "1701185190222",

"ccy": "USDT",

"copyAmt": "20",

"copyMode": "fixed\_amount",

"copyNum": "0",

"copyRatio": "",

"copyRelId": "649753013401714688",

"copyState": "0",

"copyTotalAmt": "1000",

"copyTotalPnl": "0",

"endCopyTime": "1701185190800",

"leadMode": "public",

"nickName": "Angry-ATH-Trunk",

"portLink": "https://static.okx.com/cdn/okex/users/headimages/predefined/0006.png",

"profitSharingRatio": "0.02",

"uniqueCode": "C62F5565FC1677E1"

}

],

"msg": ""

}

#### Response parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| portLink | String | Portrait link |
| nickName | String | Nick name |
| uniqueCode | String | Lead trader unique code |
| copyNum | String | Number of times to copy order |
| copyTotalAmt | String | Copy total amount |
| copyTotalPnl | String | Copy total pnl |
| copyAmt | String | Copy amount per order in USDT |
| copyMode | String | Copy mode fixed\_amount ratio\_copy |
| copyRatio | String | Copy ratio per order. |
| 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 |
| endCopyTime | String | Stop copying time. Unix timestamp format in milliseconds, e.g.1597026383085 |
| copyRelId | String | Copy relation ID |
| copyState | String | Current copy state  0: non-copy, 1: copy |
| leadMode | String | Lead mode public private |

### 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

#### HTTP request

GET /api/v5/copytrading/public-config

Request example

GET /api/v5/copytrading/public-config?instType=SWAP

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instType | String | No | Instrument type SWAP, 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 fixed\_amount |
| minCopyAmt | String | Minimum copy amount per order in USDT when you are using copy mode fixed\_amount |
| maxCopyTotalAmt | String | Maximum copy total amount under the certain lead trader, the minimum is the same with minCopyAmt |
| minCopyRatio | String | Minimum ratio per order when you are using copy mode ratio\_copy |
| maxCopyRatio | String | Maximum ratio per order when you are using copy mode ratio\_copy |
| 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

GET /api/v5/copytrading/public-lead-traders?instType=SWAP

#### 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": [

{

"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 | 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 | 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 | Pnl ratios |
| >> beginTs | String | Begin time of pnl ratio on that day |
| >> 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

#### HTTP request

GET /api/v5/copytrading/public-weekly-pnl

Request example

GET /api/v5/copytrading/public-weekly-pnl?instType=SWAP&uniqueCode=D9ADEAB33AE9EABD

#### 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

#### HTTP request

GET /api/v5/copytrading/public-pnl

Request example

GET /api/v5/copytrading/public-pnl?instType=SWAP&uniqueCode=D9ADEAB33AE9EABD&lastDays=1

#### 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 of pnl ratio on that day |
| pnl | String | Pnl on that day |
| pnlRatio | String | Pnl ratio on that day |

### GET / Lead trader stats

Public endpoint. Key data related to lead trader performance.

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: IP

#### HTTP request

GET /api/v5/copytrading/public-stats

Request example

GET /api/v5/copytrading/public-stats?instType=SWAP&uniqueCode=D9ADEAB33AE9EABD&lastDays=1

#### 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

#### HTTP request

GET /api/v5/copytrading/public-preference-currency

Request example

GET /api/v5/copytrading/public-preference-currency?instType=SWAP&uniqueCode=CB4594A3BB5D3538

#### 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": "YFII",

"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

#### HTTP request

GET /api/v5/copytrading/public-current-subpositions

Request example

GET /api/v5/copytrading/public-current-subpositions?instType=SWAP&uniqueCode=D9ADEAB33AE9EABD

#### 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 subPosId. |
| before | String | No | Pagination of data to return records newer than the requested subPosId. |
| 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": "649582930998104064",

"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 long  short  net (Long positions have positive subPos; short positions have negative subPos) |
| mgnMode | String | Margin mode. cross 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

#### HTTP request

GET /api/v5/copytrading/public-subpositions-history

Request example

GET /api/v5/copytrading/public-subpositions-history?instType=SWAP&uniqueCode=9A8534AB09862774

#### 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 subPosId. |
| before | String | No | Pagination of data to return records newer than the requested subPosId. |
| 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 long  short  net (long position has positive subPos; short position has negative subPos) |
| mgnMode | String | Margin mode. cross isolated |
| 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

#### HTTP request

GET /api/v5/copytrading/public-copy-traders

Request example

GET /api/v5/copytrading/public-copy-traders?instType=SWAP&uniqueCode=D9ADEAB33AE9EABD

#### 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 |
| limit | String | No | Number of results per request. The maximum is 100; The default is 100 |

Response example

{

"code": "0",

"data": [

{

"ccy": "USDT",

"copyTotalPnl": "2060.12242",

"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 | String | 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 |

### GET / Lead trader ranks (private)

Private endpoint. Retrieve lead trader ranks.

For requests from the ND sub-account, under the same ND broker, this private endpoint can return ND lead trader information that the related public endpoint can't return.

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

GET /api/v5/copytrading/lead-traders

Request example

GET /api/v5/copytrading/lead-traders?instType=SWAP

#### 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": [

{

"chanType": "OKX",

"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": [

{

"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 | The rank information of lead traders |
| > chanType | String | Channel type OKX ND |
| > 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 | 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 | Pnl ratios |
| >> beginTs | String | Begin time of pnl ratio on that day |
| >> pnlRatio | String | Pnl ratio on that day |

### GET / Lead trader weekly pnl (private)

Private endpoint. Retrieve lead trader weekly pnl. Results are returned in counter chronological order.

For requests from the ND sub-account, under the same ND broker, uniqueCode is supported for ND lead trader unique code by this endpoint, but the related public endpoint does not support it.

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

GET /api/v5/copytrading/weekly-pnl

Request example

GET /api/v5/copytrading/weekly-pnl?instType=SWAP&uniqueCode=D9ADEAB33AE9EABD

#### 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 (private)

Private endpoint. Retrieve lead trader daily pnl. Results are returned in counter chronological order.

For requests from the ND sub-account, under the same ND broker, uniqueCode is supported for ND lead trader unique code by this endpoint, but the related public endpoint does not support it.

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

GET /api/v5/copytrading/pnl

Request example

GET /api/v5/copytrading/pnl?instType=SWAP&uniqueCode=D9ADEAB33AE9EABD&lastDays=1

#### 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 of pnl ratio on that week |
| pnl | String | Pnl on that week |
| pnlRatio | String | Pnl ratio on that week |

### GET / Lead trader stats (private)

Private endpoint. Key data related to lead trader performance.

For requests from the ND sub-account, under the same ND broker, uniqueCode is supported for ND lead trader unique code by this endpoint, but the related public endpoint does not support it.

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

GET /api/v5/copytrading/stats

Request example

GET /api/v5/copytrading/stats?instType=SWAP&uniqueCode=D9ADEAB33AE9EABD&lastDays=1

#### 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 (private)

Private endpoint. The most frequently traded crypto of this lead trader. Results are sorted by ratio from large to small.

For requests from the ND sub-account, under the same ND broker, uniqueCode is supported for ND lead trader unique code by this endpoint, but the related public endpoint does not support it.

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

GET /api/v5/copytrading/preference-currency

Request example

GET /api/v5/copytrading/preference-currency?instType=SWAP&uniqueCode=CB4594A3BB5D3538

#### 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": "YFII",

"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 (private)

Private endpoint. Get current leading positions of lead trader

For requests from the ND sub-account, under the same ND broker, uniqueCode is supported for ND lead trader unique code by this endpoint, but the related public endpoint does not support it.

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

GET /api/v5/copytrading/performance-current-subpositions

Request example

GET /api/v5/copytrading/performance-current-subpositions?instType=SWAP&uniqueCode=D9ADEAB33AE9EABD

#### 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 subPosId. |
| before | String | No | Pagination of data to return records newer than the requested subPosId. |
| limit | String | No | Number of results per request. Maximum is 500. 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": "649582930998104064",

"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 long  short  net (Long positions have positive subPos; short positions have negative subPos) |
| mgnMode | String | Margin mode. cross 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 (private)

Private endpoint. Retrieve the lead trader completed leading position of the last 3 months.  
Returns reverse chronological order with subPosId.

For requests from the ND sub-account, under the same ND broker, uniqueCode is supported for ND lead trader unique code by this endpoint, but the related public endpoint does not support it.

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

GET /api/v5/copytrading/performance-subpositions-history

Request example

GET /api/v5/copytrading/performance-subpositions-history?instType=SWAP&uniqueCode=9A8534AB09862774

#### 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 subPosId. |
| before | String | No | Pagination of data to return records newer than the requested subPosId. |
| 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 long  short  net (long position has positive subPos; short position has negative subPos) |
| mgnMode | String | Margin mode. cross isolated |
| 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 (private)

Private endpoint. Retrieve copy trader coming from certain lead trader. Return according to pnl from high to low

For requests from the ND sub-account, under the same ND broker, uniqueCode is supported for ND lead trader unique code by this endpoint, but the related public endpoint does not support it.

#### Rate limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

GET /api/v5/copytrading/copy-traders

Request example

GET /api/v5/copytrading/copy-traders?instType=SWAP&uniqueCode=D9ADEAB33AE9EABD

#### 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 |
| limit | String | No | Number of results per request. The maximum is 100; The default is 100 |

Response example

{

"code": "0",

"data": [

{

"ccy": "USDT",

"copyTotalPnl": "2060.12242",

"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 | String | 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 / Copy trading notification channel

As a copy trader, receive push notification of copy trading.

#### URL Path

/ws/v5/business (required login)

Request Example

{

"op": "subscribe",

"args": [{

"channel": "copytrading-notification",

"instType": "SWAP"

}]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name copytrading-notification |
| > instType | String | Yes | Instrument type SWAP |
| > instId | String | No | Instrument ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "copytrading-notification",

"instType": "SWAP"

},

"connId": "aa993428"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"copytrading-notification\", \"instType\" : \"FUTURES\"}]}",

"connId":"a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| 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-notification",

"instType": "SWAP",

"uid": "116488283046944768"

},

"data": [

{

"avgPx": "",

"ccy": "USDT",

"copyTotalAmt": "10",

"infoType": "8",

"instId": "ETH-USDT-SWAP",

"instType": "SWAP",

"lever": "",

"maxLeadTraderNum": "",

"minNotional": "",

"posSide": "",

"rmThold": "",

"side": "",

"slTotalAmt": "",

"slippageRatio": "",

"subPosId": "",

"uniqueCode": "716DDB411E9673F9"

}

]

}

#### 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 | Subscribed data |
| > instType | String | Instrument type |
| > infoType | String | Information type 1: open copy position successfully for complete fill  2: close copy position for complete fill 3: more than customized total stop loss amount  4: the lead trader cancels copy trader 5. copy trading failed, insufficient account balance 6. copy trading failed for fixed amount mode, copy amount less than the value of one contract 7. copy trading failed for ratio copy mode, the number of copying less than 1 cont 8. copy trading failed, more than customized copy total amount  9. copy trading failed due to slippage protection 12. fail to close copy position. |
| > subPosId | String | Copy position ID |
| > uniqueCode | String | Lead trader unique code |
| > instId | String | Instrument ID |
| > lever | String | leverage |
| > avgPx | String | Average filled price |
| > ccy | String | Currency |
| > side | String | Side buy sell |
| > posSide | String | Position side  long short net |
| > slTotalAmt | String | Total stop loss for the trader |
| > rmThold | String | Lead trader can remove copy trader if balance of copy trader less than this value. |
| > minNotional | String | A contract value in USDT. |
| > copyTotalAmt | String | Copy total amount |
| > slippageRatio | String | Slippage ratio |
| > maxLeadTraderNum | String | Maximum lead trading every day. |

### WS / Lead trading notification channel

The notification when failing to lead trade.

#### URL Path

/ws/v5/business (required login)

Request Example

{

"op": "subscribe",

"args": [{

"channel": "copytrading-lead-notification",

"instType": "SWAP"

}]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name copytrading-lead-notification |
| > instType | String | Yes | Instrument type SWAP |
| > instId | String | No | Instrument ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "copytrading-lead-notification",

"instType": "SWAP"

},

"connId": "aa993428"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"copytrading-lead-notification\", \"instType\" : \"FUTURES\"}]}",

"connId":"a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| 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 | Subscribed data |
| > instType | String | Instrument type |
| > infoType | String | Information type 1: lead trading failed due to touch max position limitation  2: lead trading failed due to touch the maximum daily number of lead trading  3: lead trading failed due to your USDT equity less than the minimum USDT equity of lead trading |
| > subPosId | String | Lead position ID |
| > uniqueCode | String | Lead trader unique code |
| > instId | String | Instrument ID |
| > side | String | Side buy sell |
| > posSide | String | Position side  long short net |
| > 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.

### GET / Tickers

Retrieve the latest price snapshot, best bid/ask price, and trading volume in the last 24 hours.

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: IP

#### HTTP Request

GET /api/v5/market/tickers

Request Example

GET /api/v5/market/tickers?instType=SWAP

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 SPOT SWAP FUTURES OPTION |
| uly | String | No | Underlying, e.g. BTC-USD  Applicable to FUTURES/SWAP/OPTION |
| instFamily | String | No | Instrument family Applicable to FUTURES/SWAP/OPTION |

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 currency.  If it is a derivatives contract, the value is the number of base currency.  If it is SPOT/MARGIN, the value is the quantity in quote currency. |
| vol24h | String | 24h trading volume, with a unit of contract.  If it is a derivatives contract, the value is the number of contracts.  If it is SPOT/MARGIN, 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. 1597026383085 |

### GET / Ticker

Retrieve the latest price snapshot, best bid/ask price, and trading volume in the last 24 hours.

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: IP

#### HTTP Request

GET /api/v5/market/ticker

Request Example

GET /api/v5/market/ticker?instId=BTC-USD-SWAP

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. BTC-USD-SWAP |

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"

}

]

}

#### 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 currency.  If it is a derivatives contract, the value is the number of base currency.  If it is SPOT/MARGIN, the value is the quantity in quote currency. |
| vol24h | String | 24h trading volume, with a unit of contract.  If it is a derivatives contract, the value is the number of contracts.  If it is SPOT/MARGIN, 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. 1597026383085. |

### GET / Order book

Retrieve order book of the instrument.

#### Rate Limit: 40 requests per 2 seconds

#### Rate limit rule: IP

#### HTTP Request

GET /api/v5/market/books

Request Example

GET /api/v5/market/books?instId=BTC-USDT

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. BTC-USDT |
| 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"

]

],

"bids": [

[

"41006.3",

"0.30178218",

"0",

"2"

]

],

"ts": "1629966436396"

}

]

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| asks | Array | Order book on sell side |
| bids | Array | 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.

#### Rate Limit: 10 requests per 2 seconds

#### Rate limit rule: IP

#### HTTP Request

GET /api/v5/market/books-full

Request Example

GET /api/v5/market/books-full?instId=BTC-USDT&sz=1

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instId | String | Yes | Instrument ID, e.g. BTC-USDT |
| sz | String | No | Order book depth per side. Maximum 5000, e.g. 5000 bids + 5000 asks  Default returns to 1 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 | Order book on sell side |
| bids | Array | 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.

### 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/candles

Request Example

GET /api/v5/market/candles?instId=BTC-USDT

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. BTC-USDT |
| bar | String | No | Bar size, the default is 1m e.g. [1m/3m/5m/15m/30m/1H/2H/4H]  Hong Kong time opening price k-line: [6H/12H/1D/2D/3D/1W/1M/3M] UTC time 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",

"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. 1597026383085 |
| o | String | Open price |
| h | String | highest price |
| l | String | Lowest price |
| c | String | Close price |
| vol | String | Trading volume, with a unit of contract.  If it is a derivatives contract, the value is the number of contracts.  If it is SPOT/MARGIN, the value is the quantity in base currency. |
| volCcy | String | Trading volume, with a unit of currency.  If it is a derivatives contract, the value is the number of base currency.  If it is SPOT/MARGIN, 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 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/market/history-candles

Request Example

GET /api/v5/market/history-candles?instId=BTC-USDT

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. BTC-USDT |
| 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. [1s/1m/3m/5m/15m/30m/1H/2H/4H]  Hong Kong time opening price k-line: [6H/12H/1D/2D/3D/1W/1M/3M] UTC time 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",

"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. 1597026383085 |
| o | String | Open price |
| h | String | Highest price |
| l | String | Lowest price |
| c | String | Close price |
| vol | String | Trading volume, with a unit of contract.  If it is a derivatives contract, the value is the number of contracts.  If it is SPOT/MARGIN, the value is the quantity in base currency. |
| volCcy | String | Trading volume, with a unit of currency.  If it is a derivatives contract, the value is the number of base currency.  If it is SPOT/MARGIN, 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

GET /api/v5/market/trades?instId=BTC-USDT

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. BTC-USDT |
| limit | String | No | Number of results per request. The maximum is 500; The default is 100 |

Response Example

{

"code": "0",

"msg": "",

"data": [

{

"instId": "BTC-USDT",

"side": "sell",

"sz": "0.00001",

"px": "29963.2",

"tradeId": "242720720",

"ts": "1654161646974"

},

{

"instId": "BTC-USDT",

"side": "sell",

"sz": "0.00001",

"px": "29964.1",

"tradeId": "242720719",

"ts": "1654161641568"

}

]

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| instId | String | Instrument ID |
| tradeId | String | Trade ID |
| px | String | Trade price |
| sz | String | Trade quantity  For spot trading, the unit is base currency |
| side | String | Trade side  buy  sell |
| ts | String | Trade time, Unix timestamp format in milliseconds, e.g. 1597026383085. |

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

GET /api/v5/market/history-trades?instId=BTC-USDT

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. BTC-USDT |
| type | String | No | Pagination Type  1: tradeId 2: timestamp The default is 1 |
| after | String | No | Pagination of data to return records earlier than the requested tradeId or ts. |
| before | String | No | Pagination of data to return records newer than the requested tradeId.  Do not support timestamp for pagination. The latest data will be returned when using before individually |
| limit | String | No | Number of results per request. The maximum and default both are 100 |

Response Example

{

"code": "0",

"msg": "",

"data": [

{

"instId": "BTC-USDT",

"side": "sell",

"sz": "0.00001",

"px": "29963.2",

"tradeId": "242720720",

"ts": "1654161646974"

},

{

"instId": "BTC-USDT",

"side": "sell",

"sz": "0.00001",

"px": "29964.1",

"tradeId": "242720719",

"ts": "1654161641568"

}

]

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| instId | String | Instrument ID |
| tradeId | String | Trade ID |
| px | String | Trade price |
| sz | String | Trade quantity |
| side | String | Trade side  buy  sell |
| ts | String | Trade time, Unix timestamp format in milliseconds, e.g. 1597026383085. |

### 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

GET /api/v5/market/option/instrument-family-trades?instFamily=BTC-USD

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instFamily | String | Yes | Instrument family, e.g. BTC-USD Applicable to OPTION |

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 | The list trade data |
| > instId | String | The Instrument ID |
| > tradeId | String | Trade ID |
| > px | String | Trade price |
| > sz | String | Trade quantity |
| > side | String | Trade side buy sell |
| > 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

GET /api/v5/public/option-trades?instFamily=BTC-USD

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instId | String | Conditional | Instrument ID, e.g. BTC-USD-221230-4000-C, Either instId or instFamily is required. If both are passed, instId 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": "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 |
| side | String | Trade side  buy  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. 1597026383085. |

### 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

GET /api/v5/market/platform-24-volume

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. 1597026383085 |

### 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

GET /api/v5/market/call-auction-details?instId=ONDO-USDC

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instId | String | Yes | Instrument ID, e.g. BTC-USDT |

Response Example

{

"code": "0",

"msg": "",

"data": [

{

"instId": "ONDO-USDC",

"unmatchedSz": "9988764",

"eqPx": "0.6",

"matchedSz": "44978",

"state": "continuous\_trading",

"auctionEndTime": "1726542000000",

"ts": "1726542000007"

}

]

}

#### 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. |
| state | String | Trading state of the symbol call\_auction continuous\_trading |
| ts | String | Data generation time. Unix timestamp in millieseconds. |

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.   
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

{

"op": "subscribe",

"args": [

{

"channel": "tickers",

"instId": "BTC-USDT"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name tickers |
| > instId | String | Yes | Instrument ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "tickers",

"instId": "BTC-USDT"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"tickers\", \"instId\" : \"LTC-USD-200327\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| 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": "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 | 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 currency.  If it is a derivatives contract, the value is the number of base currency.  If it is SPOT/MARGIN, the value is the quantity in quote currency. |
| > vol24h | String | 24h trading volume, with a unit of contract.  If it is a derivatives contract, the value is the number of contracts.  If it is SPOT/MARGIN, 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. 1597026383085 |

### WS / 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

{

"op": "subscribe",

"args": [

{

"channel": "candle1D",

"instId": "BTC-USDT"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name  candle3M candle1M candle1W  candle1D candle2D candle3D candle5D candle12H candle6H candle4H candle2H candle1H candle30m candle15m candle5m candle3m candle1m candle1s candle3Mutc candle1Mutc candle1Wutc candle1Dutc candle2Dutc candle3Dutc candle5Dutc candle12Hutc candle6Hutc |
| > instId | String | Yes | Instrument ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "candle1D",

"instId": "BTC-USDT"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"candle1D\", \"instId\" : \"BTC-USD-191227\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| 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": "candle1D",

"instId": "BTC-USDT"

},

"data": [

[

"1597026383085",

"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 | 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, with a unit of contract.  If it is a derivatives contract, the value is the number of contracts.  If it is SPOT/MARGIN, the value is the quantity in base currency. |
| > volCcy | String | Trading volume, with a unit of currency.  If it is a derivatives contract, the value is the number of base currency.  If it is SPOT/MARGIN, 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 |

### 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

{

"op": "subscribe",

"args": [

{

"channel": "trades",

"instId": "BTC-USDT"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name trades |
| > instId | String | Yes | Instrument ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "trades",

"instId": "BTC-USDT"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"trades\", \"instId\" : \"BTC-USD-191227\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| 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": "trades",

"instId": "BTC-USDT"

},

"data": [

{

"instId": "BTC-USDT",

"tradeId": "130639474",

"px": "42219.9",

"sz": "0.12060306",

"side": "buy",

"ts": "1630048897897",

"count": "3"

}

]

}

#### Push data parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| arg | Object | Successfully subscribed channel |
| > channel | String | Channel name |
| > instId | String | Instrument ID |
| data | Array | Subscribed data |
| > instId | String | Instrument ID, e.g. BTC-USDT |
| > tradeId | String | The last trade ID in the trades aggregation |
| > px | String | Trade price |
| > sz | String | Trade size |
| > side | String | Trade direction buy sell |
| > ts | String | Filled time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| > count | String | The count of trades aggregated |

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 `tradeId` 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 `tradeId` = 123 and `count` = 3, it means the message aggregates the trades of `tradeId` = 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.

### 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

{

"op": "subscribe",

"args": [

{

"channel": "trades-all",

"instId": "BTC-USDT"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name trades-all |
| > instId | String | Yes | Instrument ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "trades-all",

"instId": "BTC-USDT"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"trades-all\", \"instId\" : \"BTC-USD-191227\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| 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": "trades-all",

"instId": "BTC-USDT"

},

"data": [

{

"instId": "BTC-USDT",

"tradeId": "130639474",

"px": "42219.9",

"sz": "0.12060306",

"side": "buy",

"ts": "1630048897897"

}

]

}

#### Push data parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| arg | Object | Successfully subscribed channel |
| > channel | String | Channel name |
| > instId | String | Instrument ID |
| data | Array | Subscribed data |
| > instId | String | Instrument ID, e.g. BTC-USDT |
| > tradeId | String | Trade ID |
| > px | String | Trade price |
| > sz | String | Trade size |
| > side | String | Trade direction buy sell |
| > ts | String | Filled time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### WS / Order book channel

Retrieve order book data.

Use books for 400 depth levels, books5 for 5 depth levels, bbo-tbt tick-by-tick 1 depth level, books50-l2-tbt tick-by-tick 50 depth levels, and books-l2-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-l2-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-l2-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-l2-tbt -> books50-l2-tbt -> books -> books5.
* Users can not simultaneously subscribe to books-l2-tbt and books50-l2-tbt/books channels for the same trading symbol.
  + For more details, please refer to the changelog [2024-07-17](https://www.okx.com/docs-v5/log_en/#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](https://www.okx.com/docs-v5/en/#overview-websocket-login)

#### URL Path

/ws/v5/public

Request Example

{

"op": "subscribe",

"args": [

{

"channel": "books",

"instId": "BTC-USDT"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name books books5 bbo-tbt books50-l2-tbt books-l2-tbt |
| > instId | String | Yes | Instrument ID |

Response Example

{

"event": "subscribe",

"arg": {

"channel": "books",

"instId": "BTC-USDT"

},

"connId": "a4d3ae55"

}

Failure example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"books\", \"instId\" : \"BTC-USD-191227\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| 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.  snapshot: full  update: incremental |
| data | Array | Subscribed data |
| > asks | Array | Order book on sell side |
| > bids | Array | 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 |
| > prevSeqId | Integer | Sequence ID of the last sent message. Only applicable to books, books-l2-tbt, books50-l2-tbt |
| > 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, 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

{

"op": "subscribe",

"args": [{

"channel": "option-trades",

"instType": "OPTION",

"instFamily": "BTC-USD"

}]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name option-trades |
| > instType | String | Yes | Instrument type, OPTION |
| > instId | String | Conditional | Instrument ID, e.g. BTC-USD-221230-4000-C, Either instId or instFamily is required. If both are passed, instId will be used. |
| > instFamily | String | Conditional | Instrument family, e.g. BTC-USD |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "option-trades",

"instType": "OPTION",

"instFamily": "BTC-USD"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"option-trades\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name status |
| 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 | Subscribed data |
| > instId | String | Instrument ID |
| > instFamily | String | Instrument family |
| > tradeId | String | Trade ID |
| > px | String | Trade price |
| > sz | String | Trade quantity |
| > side | String | Trade side  buy  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. 1597026383085. |

### WS / Call auction details channel

Retrieve call auction details.

#### URL Path

/ws/v5/public

Request Example

{

"op": "subscribe",

"args": [{

"channel": "call-auction-details",

"instId": "ONDO-USDC"

}]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name  call-auction-details |
| > instId | String | Yes | Instrument ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "call-auction-details",

"instId": "ONDO-USDC"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"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** |
| --- | --- | --- | --- |
| 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": "call-auction-details",

"instId": "ONDO-USDC"

},

"data": [

{

"instId": "ONDO-USDC",

"unmatchedSz": "9988764",

"eqPx": "0.6",

"matchedSz": "44978",

"state": "continuous\_trading",

"auctionEndTime": "1726542000000",

"ts": "1726542000007"

}

]

}

#### Push data parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| arg | Object | Successfully subscribed channel |
| > channel | String | Channel name |
| > instId | String | Instrument ID |
| data | Array | 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 call\_auction continuous\_trading |
| > ts | String | Data generation time. Unix timestamp in millieseconds. |

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`.

# 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](https://share.hsforms.com/1mYdfKtJJR3CC03IyCeC6hg3a1fq).

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: UserID

#### HTTP Requests

GET /api/v5/rfq/counterparties

Request Example

GET /api/v5/rfq/counterparties

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 counterparts

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. LP 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](https://www.okx.com/help/demo-trading)

#### Rate Limit: 5 requests per 2 seconds; 150 requests per 12 hours

#### Rate limit rule: UserID

#### HTTP Requests

POST /api/v5/rfq/create-rfq

Request Example

POST /api/v5/rfq/create-rfq

{

"anonymous": true,

"counterparties":[

"Trader1",

"Trader2"

],

"allowPartialExecution":false,

"clRfqId":"rfq01",

"tag":"123456",

"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"

}

]

}

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 true or false.  If not specified, the default value is false.  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 true or false.  false by default. |
| 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 | Trade mode  Margin mode: cross isolated  Non-Margin mode: cash.  If not provided, tdMode will inherit default values set by the system shown below:  Spot and futures mode & SPOT: cash  Buy options in Spot and futures mode and Multi-currency Margin: isolated  Other cases: cross |
| > ccy | String | No | Margin currency.  Only applicable to cross MARGIN orders in Spot and futures mode. The parameter will be ignored in other scenarios. |
| > sz | String | Yes | The size of each leg |
| > lmtPx | String | No | Taker expected price for the RFQ  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. 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. The auto execution side depends on the leg side of the RFQ. For SPOT/MARGIN/FUTURES/SWAP, lmtPx will be in unit of the quote ccy. For OPTION, lmtPx will be in unit of settle ccy. The field will not be disclosed to counterparties. |
| > side | String | Yes | The direction of each leg. Valid values can be buy or sell. |
| > posSide | String | No | Position side.  The default is net in the net mode. It can only be long or short in the long/short mode.  If not specified, users in long/short mode always open new positions.  Only applicable to FUTURES/SWAP. |
| > tgtCcy | String | No | Defines the unit of the “sz” attribute.  Only applicable to instType = SPOT.  The valid enumerations are base\_ccy and quote\_ccy. When not specified, this is equal to base\_ccy by default. |

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"

}

]

}

]

}

#### 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. |
| > 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. |
| > 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. |
| > rfqId | 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 | Trade mode  Margin mode: cross isolated  Non-Margin mode: cash.  If not provided, tdMode will inherit default values set by the system shown below:  Spot and futures mode & SPOT: cash  Buy options in Spot and futures mode and Multi-currency Margin: isolated  Other cases: cross |
| >> ccy | String | Margin currency.  Only applicable to cross MARGIN orders in Spot and futures mode. The parameter will be ignored in other scenarios. |
| >> 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 | Position side.  The default is net in the net mode. If not specified, return "", which is equivalent to net.  It can only be long or short 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.  The valid enumerations are base\_ccy and quote\_ccy. When not specified this is equal to base\_ccy by default. |

### Cancel RFQ

Cancel an existing active RFQ that you have created previously.

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Requests

POST /api/v5/rfq/cancel-rfq

Request Example

POST /api/v5/rfq/cancel-rfq

{

"rfqId":"22535",

"clRfqId":"rfq001"

}

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: UserID

#### HTTP Requests

POST /api/v5/rfq/cancel-batch-rfqs

Request Example

POST /api/v5/rfq/cancel-batch-rfqs

{

"rfqIds":[

"2201",

"2202",

"2203"

],

"clRfqIds":[

"r1",

"r2",

"r3"

]

}

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 rfqIds or clRfqIds is required.  If both attributes are sent, rfqIds 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":"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, 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 all RFQs

Cancels all active RFQs.

#### Rate Limit: 2 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Requests

POST /api/v5/rfq/cancel-all-rfqs

Request Example

POST /api/v5/rfq/cancel-all-rfqs

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: UserID

#### HTTP Requests

POST /api/v5/rfq/execute-quote

Request Example

POST /api/v5/rfq/execute-quote

{

"rfqId":"22540",

"quoteId":"84073",

"legs":[

{

"sz":"25",

"instId":"BTC-USD-20220114-13250-C"

},

{

"sz":"25",

"instId":"BTC-USDT"

}

]

}

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 . |
| quoteId | 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: tgtCcy and side of each leg will be same as ones in the RFQ. px 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, 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 |
| > 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. |
| > quoteId | String | Quote ID. |
| > clQuoteId | 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 true. |
| > mTraderCode | String | A unique identifier of the maker. Empty if the anonymous parameter of the Quote is set to be true. |
| > 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 |
| >> tradeId | String | Last traded ID. |

### 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: UserID

#### HTTP Requests

GET /api/v5/rfq/maker-instrument-settings

Request Example

GET /api/v5/rfq/maker-instrument-settings

#### Request parameters

None

Response Example

{

"code": "0",

"msg": "",

"data":

[

{"instType": "OPTION",

"includeALL": true,

"data":

[

{

"uly": "BTC-USD",

"maxBlockSz": "10000",

"makerPxBand": "5"

},

{

"uly": "SOL-USD",

"maxBlockSz": "100000",

"makerPxBand": "15"

}

]

},

{"instType": "FUTURES",

"includeALL": false,

"data":

[

{

"uly": "BTC-USD",

"maxBlockSz": "10000",

"makerPxBand": "5"

},

{

"uly": "ETH-USDT",

"maxBlockSz": "100000",

"makerPxBand": "15"

}

]

},

{"instType:": "SWAP",

"includeALL": false,

"data":

[{

"uly": "BTC-USD",

"maxBlockSz": "10000",

"makerPxBand": "5"

},

{

"uly": "ETH-USDT"

}

]

},

{"instType:": "SPOT",

"includeALL": false,

"data":

[{

"instId": "BTC-USDT"

},

{

"instId": "TRX-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 | Return data of the request. |
| > instType | String | Type of instrument. Valid value can be FUTURES, OPTION, SWAP or SPOT. |
| > includeAll | Boolean | Receive all instruments or not under specific instType setting.  Valid value can be boolean (True/False). By default, the value will be false. |
| > data | Array of objects | Elements of the instType. |
| >> instFamily | String | Instrument family. Required for FUTURES, OPTION and SWAP only. |
| >> instId | String | Instrument ID. Required for SPOT only. |
| >> maxBlockSz | String | Max trade quantity for the product(s).  For FUTURES, OPTION and SWAP, the max quantity of the RFQ/Quote is in unit of contracts. For SPOT, 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

#### Rate limit rule: UserID

#### HTTP Requests

POST /api/v5/rfq/maker-instrument-settings

Request Example

POST /api/v5/rfq/maker-instrument-settings

[

{

"instType": "OPTION",

"data":

[{

"instFamily": "BTC-USD",

"maxBlockSz": "10000",

"makerPxBand": "5"

},

{

"instFamily": "SOL-USD",

"maxBlockSz": "100000",

"makerPxBand": "15"

}]

},

{

"instType": "FUTURES",

"data":

[{

"instFamily": "BTC-USD",

"maxBlockSz": "10000",

"makerPxBand": "5"

},

{

"instFamily": "ETH-USDT",

"maxBlockSz": "100000",

"makerPxBand": "15"

}]

},

{

"instType": "SWAP",

"data":

[{

"instFamily": "BTC-USD",

"maxBlockSz": "10000",

"makerPxBand": "5"

},

{

"instFamily": "ETH-USDT"

}]

},

{

"instType": "SPOT",

"data":

[{

"instId": "BTC-USDT"

},

{

"instId": "TRX-USDT"

}]

}

]

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 FUTURES, OPTION, SWAP or SPOT. |
| includeAll | Boolean | No | Receive all instruments or not under specific instType setting.  Valid value can be boolean (True/False). By default, the value will be false. |
| data | Array of objects | Yes | Elements of the instType. |
| > instFamily | String | Conditional | Instrument family. Required for FUTURES, OPTION and SWAP only. |
| > instId | String | Conditional | Instrument ID. Required for SPOT only. |
| > maxBlockSz | String | No | Max trade quantity for the product(s).  For FUTURES, OPTION and SWAP, the max quantity of the RFQ/Quote is in unit of contracts. For SPOT, 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":"",

"data":[

{

"result":true

}

]

}

#### 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. |
| > result | Boolean | Result of the request Valid value is true or false. |

### Reset MMP status

Reset the MMP status to be inactive.

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Requests

POST /api/v5/rfq/mmp-reset

Request Example

POST /api/v5/rfq/mmp-reset

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: UserID

#### HTTP Request

POST /api/v5/rfq/mmp-config

Request Example

POST /api/v5/rfq/mmp-config

body

{

"timeInterval":"5000",

"frozenInterval":"2000",

"countLimit": "100"

}

#### 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 |

### 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: UserID

#### HTTP Request

GET /api/v5/rfq/mmp-config

Request Example

GET /api/v5/rfq/mmp-config

#### 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 diabled |
| frozenInterval | String | Frozen period (ms). If it is "0", the trade will remain frozen until manually reset and mmpFrozenUntil will be "". |
| countLimit | String | Limit in number of execution attempts |
| mmpFrozen | Boolean | Whether MMP is currently triggered. true or false |
| 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: UserID

#### HTTP Requests

POST /api/v5/rfq/create-quote

Request Example

POST /api/v5/rfq/create-quote

{

"rfqId":"22539",

"clQuoteId":"q001",

"tag":"123456",

"quoteSide":"buy",

"anonymous": true,

"expiresIn":"30",

"legs":[

{

"px":"39450.0",

"sz":"200000",

"instId":"BTC-USDT-SWAP",

"tdMode":"cross",

"ccy":"USDT",

"side":"buy",

"posSide": "long"

},

{

"px":"39450.0",

"sz":"200000",

"instId":"BTC-USDT-SWAP",

"tdMode":"cross",

"ccy":"USDT",

"side":"buy",

"posSide": "long"

}

]

}

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 true or false. false by default. |
| quoteSide | String | Yes | The trading direction of the Quote. Its value can be buy or sell.  For example, if quoteSide is buy, 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: cross isolated  Non-Margin mode: cash.  If not provided, tdMode will inherit default values set by the system shown below:  Spot and futures mode mode & SPOT: cash  Buy options in Spot and futures mode and Multi-currency Margin: isolated  Other cases: cross |
| > ccy | String | No | Margin currency.  Only applicable to cross MARGIN orders in Spot and futures mode. 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. |
| > side | String | Yes | The direction of the leg. Valid values can be buy or sell. |
| > posSide | String | No | Position side.  The default is net in the net mode. It can only be long or short in the long/short mode.  If not specified, users in long/short mode always open new positions.  Only applicable to FUTURES/SWAP. |
| > tgtCcy | String | No | Defines the unit of the “sz” attribute.  Only applicable to instType = SPOT.  The valid enumerations are base\_ccy and quote\_ccy. When not specified this is equal to base\_ccy by default. |

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, 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 |
| > 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 active canceled pending\_fill filled expired or failed. |
| > reason | String | Reasons of state. Valid values can be mmp\_canceled. |
| > 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. |
| > quoteId | String | Quote ID. |
| > clQuoteId | 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 buy or sell. For example, if quoteSide is buy, 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. BTC-USDT-SWAP |
| >> tdMode | String | Trade mode  Margin mode: cross isolated  Non-Margin mode: cash.  If not provided, tdMode will inherit default values set by the system shown below:  Spot and futures mode & SPOT: cash  Buy options in Spot and futures mode and Multi-currency Margin: isolated  Other cases: cross |
| >> ccy | String | Margin currency.  Only applicable to cross MARGIN orders in Spot and futures mode. 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 net in the net mode. If not specified, return "", which is equivalent to net.  It can only be long or short 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.  The valid enumerations are base\_ccy and quote\_ccy. When not specified this is equal to base\_ccy by default. |

### 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: UserID

#### HTTP Requests

POST /api/v5/rfq/cancel-quote

Request Example

POST /api/v5/rfq/cancel-quote

{

"quoteId": "007",

"clQuoteId":"Bond007"

}

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** |
| --- | --- | --- | --- |
| quoteId | String | Conditional | Quote ID. |
| clQuoteId | String | Conditional | Client-supplied Quote ID. Either quoteId or clQuoteId is required. If both clQuoteId and quoteId are passed, quoteId will be treated as primary identifier. |
| rfqId | 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, 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 |
| > quoteId | String | Quote ID |
| > clQuoteId | String | Client-supplied Quote ID. |
| > sCode | String | The code of the event execution result, 0 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: UserID

#### HTTP Requests

POST /api/v5/rfq/cancel-batch-quotes

Request Example

POST /api/v5/rfq/cancel-batch-quotes

{

"quoteIds": ["1150","1151","1152"],

"clQuoteIds": ["q1","q2","q3"]

}

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** |
| --- | --- | --- | --- |
| quoteIds | Array of strings | Conditional | Quote IDs . |
| clQuoteIds | Array of strings | Conditional | Client-supplied Quote IDs. Either quoteIds or clQuoteIds is required.If both attributes are sent, quoteIds will be used as primary identifier. |

Success - All requested Quotes canceled

{

"code":"0",

"msg":"",

"data":[

{

"quoteId":"1150",

"clQuoteId":"q1",

"sCode":"0",

"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, 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 |
| > quoteId | String | Quote ID |
| > clQuoteId | String | Client-supplied Quote ID. |
| > sCode | String | The code of the event execution result, 0 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: UserID

#### HTTP Requests

POST /api/v5/rfq/cancel-all-quotes

Request Example

POST /api/v5/rfq/cancel-all-quotes

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, 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 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: UserID

#### HTTP Request

POST /api/v5/rfq/cancel-all-after

Request Example

POST /api/v5/rfq/cancel-all-after

{

"timeOut":"60"

}

#### 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: UserID

#### HTTP Requests

GET /api/v5/rfq/rfqs

Request Example

GET /api/v5/rfq/rfqs

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 clRfqId and rfqId are passed, rfqId will be treated as primary identifier |
| state | String | No | The status of the RFQ.  Valid values can be active canceled pending\_fill filled expired failed traded\_away.  traded\_away 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"

}

],

"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"

}

],

"cTime": "1650966723729",

"uTime": "1650966816577"

}

}

#### 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 failed traded\_away.  traded\_away only applies to Maker |
| > counterparties | Array of srings | 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 true. |
| > 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 true or false. false by default. |
| > legs | Array of objects | Legs of RFQ |
| >> instId | String | Instrument ID, e.g. BTC-USDT-SWAP |
| >> tdMode | String | Trade mode  Margin mode: cross isolated  Non-Margin mode: cash.  If not provided, tdMode will inherit default values set by the system shown below:  Spot and futures mode & SPOT: cash  Buy options in Spot and futures mode and Multi-currency Margin: isolated  Other cases: cross |
| >> ccy | String | Margin currency.  Only applicable to cross MARGIN orders in Spot and futures mode. The parameter will be ignored in other scenarios. |
| >> 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 | Position side.  The default is net in the net mode. If not specified, return "", which is equivalent to net.  It can only be long or short 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.  The valid enumerations are base\_ccy and quote\_ccy. When not specified this is equal to base\_ccy by default. |

### 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: UserID

#### HTTP Requests

GET /api/v5/rfq/quotes

Request Example

GET /api/v5/rfq/quotes

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** |
| --- | --- | --- | --- |
| rfqId | String | No | RFQ ID . |
| clRfqId | String | No | Client-supplied RFQ ID. If both clRfqId and rfqId are passed, rfqId will be be treated as primary identifier. |
| quoteId | String | No | Quote ID |
| clQuoteId | String | No | Client-supplied Quote ID. If both clQuoteId and quoteId are passed, quoteId will be treated as primary identifier |
| state | String | No | The status of the quote. Valid values can be active canceled pending\_fill filled expired or failed. |
| beginId | String | No | Start quote id the request to begin with. Pagination of data to return records newer than the requested quoteId, not including beginId |
| endId | String | No | End quote id the request to end with. Pagination of data to return records earlier than the requested quoteId, 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":[

{

"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"

}

],

"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, 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 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 active canceled pending\_fill filled expired or failed. |
| > reason | String | Reasons of state. Valid values can be mmp\_canceled. |
| > 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. |
| > quoteId | String | Quote ID. |
| > clQuoteId | 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. Empty If the anonymous parameter of the Quote is set to be true. |
| > 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 | Trade mode  Margin mode: cross isolated  Non-Margin mode: cash.  If not provided, tdMode will inherit default values set by the system shown below:  Spot and futures mode & SPOT: cash  Buy options in Spot and futures mode and Multi-currency Margin: isolated  Other cases: cross |
| >> ccy | String | Margin currency.  Only applicable to cross MARGIN orders in Spot and futures mode. 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 net in the net mode. If not specified, return "", which is equivalent to net.  It can only be long or short 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.  The valid enumerations are base\_ccy and quote\_ccy. When not specified this is equal to base\_ccy by default. |

### 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: UserID

#### HTTP Requests

GET /api/v5/rfq/trades

Request Example

GET /api/v5/rfq/trades

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 clRfqId and rfqId are passed, rfqId 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 clQuoteId and quoteId are passed, quoteId 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 withPagination 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. true: the default value. false. |

Response Example

{

"code": "0",

"msg": "",

"data": [

{

"rfqId": "123456",

"clRfqId": "",

"quoteId": "0T5342O",

"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"

}

],

"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"

}

],

"cTime": "1650966816550",

"tTraderCode": "SATS",

"mTraderCode": "MIKE"

}

]

}

#### 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 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. |
| > quoteId | String | Quote ID. |
| > clQuoteId | 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 true. |
| > mTraderCode | String | A unique identifier of the Maker. Empty if the anonymous parameter of the Quote is set to be true. |
| > 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. 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. 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 |
| >> tradeId | String | Last traded ID. |

### 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

GET /api/v5/market/block-tickers?instType=SWAP

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 SPOT SWAP FUTURES OPTION |
| uly | String | No | Underlying, e.g. BTC-USD  Applicable to FUTURES/SWAP/OPTION |
| instFamily | String | No | Instrument family, e.g. BTC-USD Applicable to FUTURES/SWAP/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 currency.  If it is a derivatives contract, the value is the number of base currency.  If it is SPOT/MARGIN, the value is the quantity in quote currency. |
| vol24h | String | 24h trading volume, with a unit of contract.  If it is a derivatives contract, the value is the number of contracts.  If it is SPOT/MARGIN, the value is the quantity in base currency. |
| ts | String | Block ticker data generation time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### 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

GET /api/v5/market/block-ticker?instId=LTC-USD-SWAP

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. BTC-USD-SWAP |

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 currency.  If it is a derivatives contract, the value is the number of base currency.  If it is SPOT/MARGIN, the value is the quantity in quote currency. |
| vol24h | String | 24h trading volume, with a unit of contract.  If it is a derivatives contract, the value is the number of contracts.  If it is SPOT/MARGIN, the value is the quantity in base currency. |
| ts | String | Block ticker data generation time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### 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 Requests

GET /api/v5/rfq/public-trades

Request Example

GET /api/v5/rfq/public-trades

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 blockTdId the request to begin with. Pagination of data to return records newer than the requested blockTdId, not including beginId. |
| endId | String | No | The last blockTdId the request to end with. Pagination of data to return records earlier than the requested blockTdId, 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",

"legs": [

{

"instId": "BTC-USD-210826",

"side": "sell",

"sz": "100",

"px": "11000",

"tradeId": "439161457415012354"

},

{

"instId": "BTC-USD-SWAP",

"side": "sell",

"sz": "100",

"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. |
| > blockTdId | String | Block trade ID. |
| > 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. |
| >> tradeId | String | Last traded ID. |

### Get public single-leg transactions of block trades

Retrieve the recent block trading transactions of an instrument. Descending order by tradeId. 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

GET /api/v5/public/block-trades?instId=BTC-USDT

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instId | String | Yes | Instrument ID, e.g. BTC-USDT |

Response Example

{

"code":"0",

"msg":"",

"data":[

{

"fillVol": "5",

"fwdPx": "26857.86591585",

"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 |
| tradeId | String | Trade ID |
| px | String | Trade price |
| sz | String | Trade quantity |
| side | String | Trade side  buy  sell |
| fillVol | String | Implied volatility  Only applicable to OPTION |
| fwdPx | String | Forward price  Only applicable to OPTION |
| idxPx | String | Index price  Applicable to FUTURES, SWAP, OPTION |
| markPx | String | Mark price  Applicable to FUTURES, SWAP, OPTION |
| ts | String | Trade time, Unix timestamp format in milliseconds, e.g. 1597026383085. |

Up to 500 most recent historical public transaction data can be retrieved.

## 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

{

"op": "subscribe",

"args": [

{

"channel": "rfqs"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name rfqs |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "rfqs"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"rfqs\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Operation subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name rfqs |
| 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 | 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 active, canceled, filled, expired traded\_away or failed.  traded\_away 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 True. |
| > 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 true or false.  false 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: cross isolated  Non-Margin mode: cash.  If not provided, tdMode will inherit default values set by the system shown below:  Spot and futures mode & SPOT: cash  Buy options in Spot and futures mode and Multi-currency Margin: isolated  Other cases: cross |
| >> ccy | String | Margin currency.  Only applicable to cross MARGIN orders in Spot and futures mode. 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 net in the net mode. If not specified, return "", which is equivalent to net.  It can only be long or short 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.  The valid enumerations are base\_ccy and quote\_ccy. When not specified this is equal to base\_ccy by default. |

state: pending\_fill is a kind of moment state, and this channel doesn't update it.

### 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

{

"op": "subscribe",

"args": [

{

"channel": "quotes"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name quotes |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "quotes"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"quotes\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Operation subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name quotes |
| 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 | 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 active canceled filled expired or failed. |
| > reason | String | Reasons of state. Valid values can be mmp\_canceled. |
| > 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. |
| > quoteId | String | Quote ID |
| > clQuoteId | 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 True. |
| > 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 | Trade mode  Margin mode: cross isolated  Non-Margin mode: cash.  If not provided, tdMode will inherit default values set by the system shown below:  Spot and futures mode & SPOT: cash  Buy options in Spot and futures mode and Multi-currency Margin: isolated  Other cases: cross |
| >> ccy | String | Margin currency.  Only applicable to cross MARGIN orders in Spot and futures mode. The parameter will be ignored in other scenarios. |
| >> 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 | Position side.  The default is net in the net mode. If not specified, return "", which is equivalent to net.  It can only be long or short 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.  The valid enumerations are base\_ccy and quote\_ccy. When not specified this is equal to base\_ccy by default. |

### 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

{

"op": "subscribe",

"args": [

{

"channel": "struc-block-trades"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name struc-block-trades |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "struc-block-trades"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"struc-block-trades\""}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| 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,

"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 | 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. |
| > quoteId | String | Quote ID. |
| > clQuoteId | 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. |
| > 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 anonymous mode of RFQ is True. |
| > mTraderCode | String | A unique identifier of the Maker. Empty If anonymous mode of Quote is True. |
| > 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. 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 base\_ccy and quote\_ccy. When not specified this is equal to base\_ccy by default. |
| >> fee | String | Fee. Negative number represents the user transaction fee charged by the platform. Positive fee represents rebate. |
| >> feeCcy | String | Fee currency |
| >> tradeId | String | Last traded ID. |

## 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

{

"op": "subscribe",

"args": [

{

"channel": "public-struc-block-trades"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name public-struc-block-trades |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "public-struc-block-trades"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"public-struc-block-trades\""}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Operation subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name public-struc-block-trades |
| 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",

"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 |
| > channel | String | Channel name |
| data | Array | Subscribed data |
| > cTime | String | The time the trade was executed. Unix timestamp in milliseconds. |
| > blockTdId | String | Block trade ID. |
| > 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 from the Takers perspective. Valid value can be buy or sell. |
| >> tradeId | String | Last traded ID. |

### 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

{

"op": "subscribe",

"args": [

{

"channel": "public-block-trades",

"instId": "BTC-USDT-SWAP"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | 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

{

"event": "subscribe",

"arg": {

"channel": "public-block-trades",

"instId": "BTC-USDT-SWAP",

"connId": "a4d3ae55"

}

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"args\":[{ \"channel\" : \"public-block-trades\""}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| 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 |
| 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",

"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 | 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 size. |
| > side | String | Trade direction, buy, sell, from taker perspective. |
| > fillVol | String | Implied volatility  Only applicable to OPTION |
| > fwdPx | String | Forward price  Only applicable to options |
| > idxPx | String | Index price  Applicable to FUTURES, SWAP, OPTION |
| > markPx | String | Mark price  Applicable to FUTURES, SWAP, OPTION |
| > ts | String | Filled time, Unix timestamp format in milliseconds, e.g. 1597026383085. |

### 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

{

"op": "subscribe",

"args": [{

"channel": "block-tickers",

"instId": "BTC-USDT"

}]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name block-tickers |
| > instId | String | Yes | Instrument ID e.g. BTC-USDT-SWAP |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "block-tickers",

"instId": "LTC-USD-200327"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"block-tickers\", \"instId\" : \"LTC-USD-200327\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name block-tickers |
| > 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 |
| > channel | String | Channel name |
| > instId | String | Instrument ID |
| data | Array | Subscribed data |
| > instId | String | Instrument ID |
| > instType | String | Instrument type |
| > volCcy24h | String | 24h trading volume, with a unit of currency.  If it is a derivatives contract, the value is the number of base currency.  If it is SPOT/MARGIN, the value is the quantity in quote currency. |
| > vol24h | String | 24h trading volume, with a unit of contract.  If it is a derivatives contract, the value is the number of contracts.  If it is SPOT/MARGIN, the value is the quantity in base currency. |
| > ts | String | Block ticker data generation time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

# 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-ordersWebSocket 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-tradesWebSocket 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 statedenoted 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-tradesWebSocket 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: UserID

#### HTTP Request

POST /api/v5/sprd/order

Request Example

# place order for a spread

POST /api/v5/sprd/order

body

{

"sprdId":"BTC-USDT\_BTC-USDT-SWAP",

"clOrdId":"b15",

"side":"buy",

"ordType":"limit",

"px":"2.15",

"sz":"2"

}

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-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. The unit is USD for inverse spreads, and the corresponding baseCcy for linear and hybrid spreads. |
| px | String | Yes | Order price. Only applicable to limit, post\_only, ioc |

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: UserID

#### HTTP Request

POST /api/v5/sprd/cancel-order

Request Example

POST /api/v5/sprd/cancel-order

body

{

"ordId":"2510789768709120"

}

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** |
| --- | --- | --- | --- |
| 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 |

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: UserID

#### HTTP Request

POST /api/v5/sprd/mass-cancel

Request Example

POST /api/v5/sprd/mass-cancel

body

{

"sprdId": "BTC-USDT\_BTC-USDT-SWAP"

}

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 true, false |

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: UserID

#### HTTP Request

POST /api/v5/sprd/amend-order

Request Example

POST /api/v5/sprd/amend-order

body

{

"ordId":"2510789768709120",

"newSz":"2"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| 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  Either newSz or newPx is required.  When amending a partially-filled order, the newSz should include the amount that has been filled. |
| newPx | String | Conditional | New price after amendment  Either newSz or newPx 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: UserID

#### HTTP Request

GET /api/v5/sprd/order

Request Example

GET /api/v5/sprd/order?ordId=2510789768709120

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='1905309079888199680')

print(result)

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| 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. 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** |
| --- | --- | --- |
| 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 |
| ordType | String | Order type market: Market order  limit: Limit order  post\_only: Post-only order  ioc: 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  canceled  live  partially\_filled  filled |
| cancelSource | String | Source of the order cancellation.Valid values and the corresponding meanings are:  0: Order canceled by system  1: Order canceled by user  14: Order canceled: IOC order was partially canceled due to incompletely filled 15: Order canceled: The order price is beyond the limit 20: Cancel all after triggered  31: The post-only order will take liquidity in maker orders 32: Self trade prevention 34: Order failed to settle due to insufficient margin  35: Order cancellation due to insufficient margin from another order |
| uTime | String | Update time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| cTime | String | Creation time, Unix timestamp format in milliseconds, e.g. 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: UserID

#### HTTP Request

GET /api/v5/sprd/orders-pending

Request Example

GET /api/v5/sprd/orders-pending

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 market: Market order  limit: Limit order  post\_only: Post-only order  ioc: Immediate-or-cancel order |
| state | String | No | State  live  partially\_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 |
| 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 |
| ordType | String | Order type market: Market order  limit: Limit order  post\_only: Post-only order  ioc: 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  live  partially\_filled |
| cancelSource | String | Source of the order cancellation.Valid values and the corresponding meanings are:  0: Order canceled by system  1: Order canceled by user  14: Order canceled: IOC order was partially canceled due to incompletely filled 15: Order canceled: The order price is beyond the limit 20: Cancel all after triggered  31: The post-only order will take liquidity in maker orders 32: Self trade prevention  34: Order failed to settle due to insufficient margin  35: Order cancellation due to insufficient margin from another order |
| uTime | String | Update time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| cTime | String | Creation time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### 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: UserID

#### HTTP Request

GET /api/v5/sprd/orders-history

Request Example

GET /api/v5/sprd/orders-history

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 market: Market order  limit: limit order  post\_only: Post-only order  ioc: Immediate-or-cancel order |
| state | String | No | State  canceled  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. 1597026383085. Date older than 7 days will be truncated. |
| 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",

"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 |
| ordType | String | Order type market: Market order  limit: limit order  post\_only: Post-only order  ioc: 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, inluding pendingSettleSz |
| 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  canceled  filled |
| cancelSource | String | Source of the order cancellation. Valid values and the corresponding meanings are:  0: Order canceled by system  1: Order canceled by user  14: Order canceled: IOC order was partially canceled due to incompletely filled 15: Order canceled: The order price is beyond the limit 20: Cancel all after triggered  31: The post-only order will take liquidity in maker orders 32: Self trade prevention 34: Order failed to settle due to insufficient margin  35: Order cancellation due to insufficient margin from another order |
| uTime | String | Update time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| cTime | String | Creation time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### 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: UserID

#### HTTP Request

GET /api/v5/sprd/orders-history-archive

Request Example

GET /api/v5/sprd/orders-history-archive

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| sprdId | String | No | spread ID, e.g. |
| ordType | String | No | Order type market: Market order  limit: limit order  post\_only: Post-only order  ioc: Immediate-or-cancel order |
| state | String | No | State  canceled  filled |
| instType | String | No | Instrument type SPOT FUTURES SWAP  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. 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",

"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": "canceled",

"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 |
| ordType | String | Order type market: Market order  limit: limit order  post\_only: Post-only order  ioc: 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, inluding pendingSettleSz |
| 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  canceled  filled |
| cancelSource | String | Source of the order cancellation. Valid values and the corresponding meanings are:  0: Order canceled by system  1: Order canceled by user  14: Order canceled: IOC order was partially canceled due to incompletely filled 15: Order canceled: The order price is beyond the limit 20: Cancel all after triggered  31: The post-only order will take liquidity in maker orders 32: Self trade prevention 34: Order failed to settle due to insufficient margin  35: Order cancellation due to insufficient margin from another order |
| uTime | String | Update time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| cTime | String | Creation time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### 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: UserID

#### HTTP Request

GET /api/v5/sprd/trades

Request Example

GET /api/v5/sprd/trades

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. |
| tradeId | 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 tradeId, not including beginId |
| endId | String | No | End trade ID the request to end with. Pagination of data to return records earlier than the requested tradeId, not including endId |
| 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",

"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 |
| tradeId | 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, 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 | 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 buy or sell. |
| > 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

#### HTTP Request

GET /api/v5/sprd/spreads

Request Example

GET /api/v5/sprd/spreads?instId=BTC-USDT

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. |
| sprdId | String | No | The spread ID |
| state | string | No | Spreads which are available to trade, suspened or expired. Valid values include live, suspend and expired. |

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": "1703836800000",

"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 linear, inverse, hybrid |
| state | String | Current state of the spread. Valid values include live, expired, suspend. |
| 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. 1597026383085 |
| expTime | String | Expiry time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| uTime | String | The timestamp the spread was last updated. Unix timestamp format in milliseconds, e.g. 1597026383085 |
| 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 buy and sell. |

### Get order book (Public)

Retrieve the order book of the spread.

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: IP

#### HTTP Request

GET /api/v5/sprd/books

Request Example

GET /api/v5/sprd/books?sprdId=BTC-USDT\_BTC-USDT-SWAP

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 | Order book on sell side |
| bids | Array | 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

GET /api/v5/market/sprd-ticker?sprdId=BTC-USDT\_BTC-USDT-SWAP

#### 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

#### HTTP Request

GET /api/v5/sprd/public-trades

Request Example

GET /api/v5/sprd/public-trades?sprdId=BTC-USDT\_BTC-USDT-SWAP

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 |
| tradeId | String | Trade ID |
| px | String | Trade price |
| sz | String | Trade quantity |
| side | String | Trade side of the taker.  buy  sell |
| ts | String | Trade time, Unix timestamp format in milliseconds, e.g. 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

GET /api/v5/market/sprd-candles?sprdId=BTC-USDT\_BTC-USDT-SWAP

#### 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]  Hong Kong time opening price k-line:[6H/12H/1D/2D/3D/1W/1M/3M]  UTC time 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 |
| confirm | String | The state of candlesticks.  0 represents that it is uncompleted  1 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

GET /api/v5/market/sprd-history-candles?sprdId=BTC-USDT\_BTC-USDT-SWAP

#### 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]  Hong Kong time opening price k-line:[6H/12H/1D/2D/3D/1W/1M/3M]  UTC time 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.  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]

### 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: UserID

#### HTTP Request

POST /api/v5/sprd/cancel-all-after

Request Example

POST /api/v5/sprd/cancel-all-after

{

"timeOut":"30"

}

#### 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: UserID

Rate limit is shared with the Nitro Spread `Place order` REST API endpoints

Request Example

{

"id": "1512",

"op": "sprd-order",

"args": [

{

"sprdId":"BTC-USDT\_BTC-USDT-SWAP",

"clOrdId":"b15",

"side":"buy",

"ordType":"limit",

"px":"2.15",

"sz":"2"

}

]

}

#### 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 | 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": "5XXXX",

"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 | 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: UserID

Rate limit is shared with the `Amend order` REST API endpoints

Request Example

{

"id":"1512",

"op":"sprd-amend-order",

"args":[

{

"ordId":"2510789768709120",

"newSz":"2"

}

]

}

#### 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 | 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 |
| > 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 newSz or newPx 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 | 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 server. The result of the amend is subject to the status pushed by the order channel or the order status query.

### WS / Cancel order

Cancel an incomplete order

#### URL Path

/ws/v5/business (required login)

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: UserID

Rate limit is shared with the Nitro Spread `Cancel order` REST API endpoints

Request Example

{

"id": "1514",

"op": "sprd-cancel-order",

"args": [

{

"ordId": "2510789768709120"

}

]

}

#### 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 | 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": "5XXXX",

"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 | 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: UserID

Request Example

{

"id": "1512",

"op": "sprd-mass-cancel",

"args": [{

"sprdId": "BTC-USDT\_BTC-USDT-SWAP"

}]

}

#### 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 | Yes | Request parameters |
| > 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 | Data |
| > result | Boolean | Result of the request true, false |

## WebSocket Private Channel

* Production Trading URL: wss://ws.okx.com:8443/ws/v5/business
* AWS Production Trading URL: wss://wsaws.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

{

"op": "subscribe",

"args": [

{

"channel": "sprd-orders",

"sprdId": "BTC-USDT\_BTC-USDT-SWAP"

}

]

}

Request Example:

{

"op": "subscribe",

"args": [

{

"channel": "sprd-orders",

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name sprd-orders |
| > sprdId | String | No | Spread ID |

Successful Response Example : single

{

"event": "subscribe",

"arg": {

"channel": "sprd-orders",

"sprdId": "BTC-USDT\_BTC-UST-SWAP"

},

"connId": "a4d3ae55"

}

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "sprd-orders"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"sprd-orders\", \"instType\" : \"FUTURES\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Required** | **Type** | **Description** |
| --- | --- | --- | --- |
| 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 |

Push Data Example: single

{

"arg": {

"channel": "sprd-orders",

"sprdId": "BTC-USDT\_BTC-USDT-SWAP",

"uid": "614488474791936"

},

"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",

"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 | 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 market: Market order  limit: limit order  post\_only: Post-only order  ioc: Immediate-or-cancel order |
| > 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:  canceled  live  partially\_filled  filled |
| > cancelSource | String | Source of the order cancellation.Valid values and the corresponding meanings are:  0: Order canceled by system  1: Order canceled by user  14: Order canceled: IOC order was partially canceled due to incompletely filled 15: Order canceled: The order price is beyond the limit 20: Cancel all after triggered  31: The post-only order will take liquidity in maker orders 32: Self trade prevention  34: Order failed to settle due to insufficient margin  35: Order cancellation due to insufficient margin from another order |
| > 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  -1: failure  0: success "" will be returned if there is no order amendment. |

### Trades channel

All updates relating to User's Trades are sent through the sprd-trades WebSocket Notifications channel.

This is a private channel and consumable solely by the authenticated user.

Updates received through the sprd-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.

#### URL Path

/ws/v5/business (required login)

Request Example : single

{

"op": "subscribe",

"args": [

{

"channel": "sprd-trades",

"sprdId": "BTC-USDT\_BTC-USDT-SWAP"

}

]

}

Request Example:

{

"op": "subscribe",

"args": [

{

"channel": "sprd-trades",

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name sprd-trades |
| > sprdId | String | No | Spread ID |

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| 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

{

"arg": {

"channel": "sprd-trades",

"sprdId": "BTC-USDT\_BTC-USDT-SWAP",

"uid": "614488474791936"

},

"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": ""

}

]

}

#### 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 | Subscribed data |
| > sprdId | String | spread ID |
| > tradeId | String | Trade ID |
| > ordId | String | Order ID |
| > clOrdId | String | Client Order ID as assigned by the client |
| > 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 buy or sell. |
| >> 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 "" |

## WebSocket Public Channel

* Production Trading URL: wss://ws.okx.com:8443/ws/v5/business
* AWS Production Trading URL: wss://wsaws.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-l2-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-l2-tbt -> sprd-books5.

#### URL Path

/ws/v5/business

Request Example: sprd-books5

{

"op": "subscribe",

"args": [

{

"channel": "sprd-books5",

"sprdId": "BTC-USDT\_BTC-USDT-SWAP"

}

]

}

Request Example: sprd-books-l2-tbt

{

"op": "subscribe",

"args": [

{

"channel": "sprd-books-l2-tbt",

"sprdId": "BTC-USDT\_BTC-USDT-SWAP"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name |
| > sprdId | String | Yes | spread ID |

Successful Response Example: sprd-books5

{

"event": "subscribe",

"arg": {

"channel": "sprd-books5",

"sprdId": "BTC-USDT\_BTC-USDT-SWAP"

},

"connId": "a4d3ae55"

}

Successful Response Example: sprd-books-l2-tbt

{

"event":"subscribe",

"arg":{

"channel":"sprd-books-l2-tbt",

"sprdId":"BTC-USDT\_BTC-USDT-SWAP"

},

"connId":"214fdd24"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"args\":[{ \"channel\" : \"sprd-books5\", \"sprdId\" : \"BTC-USD\_BTC-USD-191227\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name |
| > sprdId | String | Yes | spread ID |
| msg | String | No | Error message |
| code | String | No | Error code |
| 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 |
| > channel | String | Channel name |
| > sprdId | String | spread ID |
| action | String | Push data action, incremental data or full snapshot. snapshot: full update: incremental |
| data | Array | Subscribed data |
| > asks | Array | Order book on sell side |
| > bids | Array | 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 sprd-books-l2-tbt. |
| > prevSeqId | Integer | Sequence ID of the last sent message. Only applicable to sprd-books-l2-tbt. |
| > 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

{

"op": "subscribe",

"args": [

{

"channel": "sprd-public-trades",

"sprdId": "BTC-USDT\_BTC-USDT-SWAP"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name sprd-public-trades |
| > sprdId | String | Yes | spread ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "sprd-public-trades",

"sprdId": "BTC-USDT\_BTC-USDT-SWAP"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"sprd-public-trades\", \"instId\" : \"BTC-USD-191227\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| 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

{

"arg": {

"channel": "sprd-public-trades",

"sprdId": "BTC-USDT\_BTC-USDT-SWAP"

},

"data": [

{

"sprdId": "BTC-USDT\_BTC-USDT-SWAP",

"tradeId": "2499206329160695808",

"px": "-10",

"sz": "0.001",

"side": "sell",

"ts": "1726801105519"

}

]

}

#### Push data parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| arg | Object | Successfully subscribed channel |
| > channel | String | Channel name |
| > sprdId | String | spread ID |
| data | Array | Subscribed data |
| > sprdId | String | spread ID, e.g. |
| > tradeId | String | Trade ID |
| > px | String | Trade price |
| > sz | String | Trade size |
| > 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

{

"op": "subscribe",

"args": [

{

"channel": "sprd-tickers",

"sprdId": "BTC-USDT\_BTC-USDT-SWAP"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name sprd-tickers |
| > sprdId | String | Yes | spread ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "sprd-tickers",

"sprdId": "BTC-USDT\_BTC-USDT-SWAP"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"sprd-tickers\", \"instId\" : \"LTC-USD-200327\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| 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

{

"arg": {

"channel": "sprd-tickers",

"sprdId": "BTC-USDT\_BTC-USDT-SWAP"

},

"data": [

{

"sprdId": "BTC-USDT\_BTC-USDT-SWAP",

"last": "4",

"lastSz": "0.01",

"askPx": "19.7",

"askSz": "5.79",

"bidPx": "5.9",

"bidSz": "5.79",

"open24h": "-7",

"high24h": "19.6",

"low24h": "-7",

"vol24h": "9.87",

"ts": "1715247061026"

}

]

}

#### Push data parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| arg | Object | Successfully subscribed channel |
| > channel | String | Channel name |
| > sprdId | String | spread ID |
| data | Array | 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

{

"op":"subscribe",

"args":[

{

"channel":"sprd-candle1D",

"sprdId":"BTC-USDT\_BTC-USDT-SWAP"

}

]

}

#### Request parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation, subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| channel | String | Yes | Channel name  sprd-candle3M sprd-candle1M  sprd-candle1W  sprd-candle1D sprd-candle2D sprd-candle3D sprd-candle5D  sprd-candle12H sprd-candle6H sprd-candle4H sprd-candle2H sprd-candle1H  sprd-candle30m sprd-candle15m sprd-candle5m sprd-candle3m sprd-candle1m  sprd-candle3Mutc sprd-candle1Mutc sprd-candle1Wutc sprd-candle1Dutc sprd-candle2Dutc sprd-candle3Dutc sprd-candle5Dutc sprd-candle12Hutc sprd-candle6Hutc |
| sprdId | String | Yes | Spread ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "sprd-candle1D",

"sprdId": "BTC-USDT\_BTC-USDT-SWAP"

}

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"sprd-candle1D\", \"instId\" : \"BTC-USD-191227\"}]}"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| 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 |

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 | 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.

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: IP + instrumentType

#### HTTP Request

GET /api/v5/public/instruments

Request Example

GET /api/v5/public/instruments?instType=SPOT

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 SPOT: Spot MARGIN: Margin SWAP: Perpetual Futures FUTURES: Expiry Futures OPTION: Option |
| uly | String | Conditional | Underlying  Only applicable to FUTURES/SWAP/OPTION.If instType is OPTION, either uly or instFamily is required. |
| instFamily | String | Conditional | Instrument family Only applicable to FUTURES/SWAP/OPTION. If instType is OPTION, either uly or instFamily is required. |
| instId | String | No | Instrument ID |

Response Example

{

"code":"0",

"msg":"",

"data":[

{

"alias": "",

"auctionEndTime": "",

"baseCcy": "BTC",

"category": "1",

"ctMult": "",

"ctType": "",

"ctVal": "",

"ctValCcy": "",

"expTime": "",

"instFamily": "",

"instId": "BTC-USDT",

"instType": "SPOT",

"lever": "10",

"listTime": "1606468572000",

"lotSz": "0.00000001",

"maxIcebergSz": "9999999999.0000000000000000",

"maxLmtAmt": "1000000",

"maxLmtSz": "9999999999",

"maxMktAmt": "1000000",

"maxMktSz": "",

"maxStopSz": "",

"maxTriggerSz": "9999999999.0000000000000000",

"maxTwapSz": "9999999999.0000000000000000",

"minSz": "0.00001",

"optType": "",

"quoteCcy": "USDT",

"settleCcy": "",

"state": "live",

"ruleType": "normal",

"stk": "",

"tickSz": "0.1",

"uly": ""

}

]

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| instType | String | Instrument type |
| instId | String | Instrument ID, e.g. BTC-USD-SWAP |
| uly | String | Underlying, e.g. BTC-USD  Only applicable to MARGIN/FUTURES/SWAP/OPTION |
| instFamily | String | Instrument family, e.g. BTC-USD  Only applicable to MARGIN/FUTURES/SWAP/OPTION |
| category | String | Currency category. Note: this parameter is already deprecated |
| baseCcy | String | Base currency, e.g. BTC inBTC-USDT  Only applicable to SPOT/MARGIN |
| quoteCcy | String | Quote currency, e.g. USDT in BTC-USDT  Only applicable to SPOT/MARGIN |
| settleCcy | String | Settlement and margin currency, e.g. BTC  Only applicable to FUTURES/SWAP/OPTION |
| ctVal | String | Contract value  Only applicable to FUTURES/SWAP/OPTION |
| ctMult | String | Contract multiplier  Only applicable to FUTURES/SWAP/OPTION |
| ctValCcy | String | Contract value currency  Only applicable to FUTURES/SWAP/OPTION |
| optType | String | Option type, C: Call P: put  Only applicable to OPTION |
| stk | String | Strike price  Only applicable to OPTION |
| listTime | String | Listing time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| auctionEndTime | String | The end time of call auction, Unix timestamp format in milliseconds, e.g. 1597026383085  Only applicable to SPOT that are listed through call auctions, return "" in other cases |
| expTime | String | Expiry time  Applicable to SPOT/MARGIN/FUTURES/SWAP/OPTION. For FUTURES/OPTION, it is natural delivery/exercise time. It is the instrument offline time when there is SPOT/MARGIN/FUTURES/SWAP/ manual offline. Update once change. |
| lever | String | Max Leverage,  Not applicable to SPOT, OPTION |
| tickSz | String | Tick size, e.g. 0.0001 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 SPOT/MARGIN, the value is the quantity in base currency. |
| minSz | String | Minimum order size If it is a derivatives contract, the value is the number of contracts. If it is SPOT/MARGIN, the value is the quantity in base currency. |
| ctType | String | Contract type linear: linear contract inverse: inverse contract  Only applicable to FUTURES/SWAP |
| alias | String | Alias this\_week next\_week this\_month next\_month quarter next\_quarter Only applicable to FUTURES  **Not recommended for use, users are encouraged to rely on the expTime field to determine the delivery time of the contract** |
| state | String | Instrument status live  suspend preopen. e.g. There will be preopen before the Futures and Options new contracts state is live. test: Test pairs, can't be traded |
| ruleType | String | Trading rule types normal: normal trading pre\_market: pre-market trading |
| 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 SPOT/MARGIN, the value is the quantity in base currency. |
| 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 SPOT/MARGIN, the value is the quantity in USDT. |
| maxLmtAmt | String | Max USD amount for a single limit order |
| maxMktAmt | String | Max USD amount for a single market order  Only applicable to SPOT/MARGIN |
| 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 SPOT/MARGIN, the value is the quantity in base currency.  The minimum order quantity of a single TWAP order is minSz\*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 SPOT/MARGIN, the value is the quantity in base currency. |
| 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 SPOT/MARGIN, the value is the quantity in base currency. |
| 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 SPOT/MARGIN, the value is the quantity in USDT. |

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 auctionEndTime  
For spot symbols listed through a call auction, listTime represents the start time of the auction, and auctionEndTime indicates the end of the auction and the start of continuous trading. For other scenarios, listTime will mark the beginning of continuous trading, and auctionEndTime 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 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 + (instrumentType + uly)

#### HTTP Request

GET /api/v5/public/delivery-exercise-history

Request Example

GET /api/v5/public/delivery-exercise-history?instType=OPTION&uly=BTC-USD

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",

uly="BTC-USD"

)

print(result)

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instType | String | Yes | Instrument type FUTURES OPTION |
| uly | String | Conditional | Underlying, only applicable to FUTURES/OPTION Either uly or instFamily is required. If both are passed, instFamily will be used. |
| instFamily | String | Conditional | Instrument family, only applicable to FUTURES/OPTION Either uly or instFamily is required. If both are passed, instFamily will be used. |
| 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 |
| limit | String | No | Number of results per request. The maximum is 100; The default is 100 |

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. 1597026383085 |
| details | String | Delivery/exercise details |
| > insId | String | Delivery/exercise contract ID |
| > px | String | Delivery/exercise price |
| > type | String | Type  delivery  exercised  expired\_otm:Out of the money |

### Get open interest

Retrieve the total open interest for contracts on OKX.

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: IP +instrumentID

#### HTTP Request

GET /api/v5/public/open-interest

Request Example

GET /api/v5/public/open-interest?instType=SWAP

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 SWAP FUTURES OPTION |
| uly | String | Conditional | Underlying Applicable to FUTURES/SWAP/OPTION.  If instType is OPTION, either uly or instFamily is required. |
| instFamily | String | Conditional | Instrument family Applicable to FUTURES/SWAP/OPTION If instType is OPTION, either uly or instFamily is required. |
| instId | String | No | Instrument ID, e.g. BTC-USDT-SWAP Applicable to FUTURES/SWAP/OPTION |

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 |
| 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. 1597026383085 |

### Get funding rate

Retrieve funding rate.

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: IP +instrumentID

#### HTTP Request

GET /api/v5/public/funding-rate

Request Example

GET /api/v5/public/funding-rate?instId=BTC-USD-SWAP

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. BTC-USD-SWAP  only applicable to SWAP |

Response Example

{

"code": "0",

"data": [

{

"fundingRate": "0.0000792386885340",

"fundingTime": "1703088000000",

"instId": "BTC-USDT-SWAP",

"instType": "SWAP",

"method": "next\_period",

"maxFundingRate": "0.00375",

"minFundingRate": "-0.00375",

"nextFundingRate": "0.0002061194322149",

"nextFundingTime": "1703116800000",

"premium": "0.0001233824646391",

"settFundingRate": "0.0001418433662153",

"settState": "settled",

"ts": "1703070685309"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| instType | String | Instrument type SWAP |
| instId | String | Instrument ID, e.g. BTC-USD-SWAP |
| method | String | Funding rate mechanism  current\_period  next\_period |
| fundingRate | String | Current funding rate |
| nextFundingRate | String | Forecasted funding rate for the next period  The nextFundingRate will be "" if the method is current\_period |
| fundingTime | String | Settlement time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| nextFundingTime | String | Forecasted funding time for the next period , Unix timestamp format in milliseconds, e.g. 1597026383085 |
| 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 |
| settState | String | Settlement state of funding rate  processing  settled |
| settFundingRate | String | If settState = processing, it is the funding rate that is being used for current settlement cycle.  If settState = settled, it is the funding rate that is being used for previous settlement cycle |
| premium | String | Premium between the mid price of perps market and the index price |
| ts | String | Data return time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

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 retrieve data from the last 3 months.

#### Rate Limit: 10 requests per 2 seconds

#### Rate limit rule: IP +instrumentID

#### HTTP Request

GET /api/v5/public/funding-rate-history

Request Example

GET /api/v5/public/funding-rate-history?instId=BTC-USD-SWAP

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. BTC-USD-SWAP  only applicable to SWAP |
| before | String | No | Pagination of data to return records newer than the requested fundingTime |
| after | String | No | Pagination of data to return records earlier than the requested fundingTime |
| limit | String | No | Number of results per request. The maximum is 100; The default is 100 |

Response Example

{

"code":"0",

"msg":"",

"data":[

{

"fundingRate": "0.0000746604960499",

"fundingTime": "1703059200000",

"instId": "BTC-USD-SWAP",

"instType": "SWAP",

"method": "next\_period",

"realizedRate": "0.0000746572360545"

},

{

"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 SWAP |
| instId | String | Instrument ID, e.g. BTC-USD-SWAP |
| fundingRate | String | Predicted funding rate |
| realizedRate | String | Actual funding rate |
| fundingTime | String | Settlement time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| method | String | Funding rate mechanism  current\_period  next\_period |

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 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

GET /api/v5/public/price-limit?instId=BTC-USDT-SWAP

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. BTC-USDT-SWAP |

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. BTC-USDT-SWAP |
| 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. 1597026383085 |
| enabled | Boolean | Whether price limit is effective  true: the price limit is effective  false: 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 +uly

#### HTTP Request

GET /api/v5/public/opt-summary

Request Example

GET /api/v5/public/opt-summary?uly=BTC-USD

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(

uly="BTC-USD",

)

print(result)

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| uly | String | Conditional | Underlying, only applicable to OPTION Either uly or instFamily is required. If both are passed, instFamily will be used. |
| instFamily | String | Conditional | Instrument family, only applicable to OPTION Either uly or instFamily is required. If both are passed, instFamily will be used. |
| 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": "0.000018173851073258973",

"vegaBS": "0.7089307622132419"

},

{

"askVol": "1.7968814062499998",

"bidVol": "0",

"delta": "-0.014668822072611904",

"deltaBS": "-0.01426678984554619",

"fwdPx": "39016.8143629068452065",

"gamma": "0.49483062407551576",

"gammaBS": "0.000011933182397798109",

"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": "0.000018173851073258973",

"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 |
| ts | String | Data update time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### 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 +instId

#### HTTP Request

GET /api/v5/public/estimated-price

Request Example

GET /api/v5/public/estimated-price?instId=BTC-USDT-191227

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-USDT-220916",

)

print(result)

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instId | String | Yes | Instrument ID, e.g. BTC-USD-200214  only applicable to FUTURES/OPTION |

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 FUTURES OPTION |
| instId | String | Instrument ID, e.g. BTC-USDT-SWAP |
| settlePx | String | Estimated delivery price |
| ts | String | Data return 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

GET /api/v5/public/discount-rate-interest-free-quota?ccy=BTC

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 |  |

Response Example

{

"code": "0",

"data": [

{

"amt": "0",

"ccy": "BTC",

"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 |
| amt | String | Interest-free quota |
| discountLv | String |  |
| minDiscountRate | String | Minimum discount rate when it exceeds the maximum amount of the last tier. |
| details | Array | New discount details. |
| > discountRate | String | Discount rate |
| > maxAmt | String | Tier - upper bound.  The unit is the currency like BTC. "" means positive infinity |
| > minAmt | String | Tier - lower bound.  The unit is the currency like BTC. The minimum is 0 |
| > tier | String | Tiers |
| > liqPenaltyRate | String | Liquidation penalty rate |
| > disCcyEq | String | Discount equity in currency for quick calculation if your equity is themaxAmt |

### 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

GET /api/v5/public/time

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 +instrumentID

#### HTTP Request

GET /api/v5/public/mark-price

Request Example

GET /api/v5/public/mark-price?instType=SWAP

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 MARGIN SWAP FUTURES OPTION |
| uly | String | No | Underlying Applicable to FUTURES/SWAP/OPTION |
| instFamily | String | No | Instrument family Applicable to FUTURES/SWAP/OPTION |
| instId | String | No | Instrument ID, e.g. 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 MARGIN SWAP FUTURES OPTION |
| instId | String | Instrument ID, e.g. BTC-USD-200214 |
| markPx | String | Mark price |
| ts | String | Data return time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### Get position tiers

Retrieve position tiers information, maximum leverage depends on your borrowings and margin ratio.

#### Rate Limit: 10 requests per 2 seconds

#### Rate limit rule: IP

#### HTTP Request

GET /api/v5/public/position-tiers

Request Example

GET /api/v5/public/position-tiers?tdMode=cross&instType=SWAP&instFamily=BTC-USDT

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 MARGIN SWAP FUTURES OPTION |
| tdMode | String | Yes | Trade mode Margin mode cross isolated |
| uly | String | Conditional | Single underlying or multiple underlyings (no more than 3) separated with comma. If instType is SWAP/FUTURES/OPTION, either uly or instFamily is required. If both are passed, instFamily will be used. |
| instFamily | String | Conditional | Single instrument familiy or multiple instrument families (no more than 5) separated with comma. If instType is SWAP/FUTURES/OPTION, either uly or instFamily is required. If both are passed, instFamily will be used. |
| 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 SWAP,FUTURES,OPTION |
| ccy | String | Conditional | Margin currency Only applicable to cross MARGIN. It will return borrowing amount for Multi-currency margin and Portfolio margin when ccy 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 FUTURES/SWAP/OPTION |
| instFamily | String | Instrument family Applicable to FUTURES/SWAP/OPTION |
| 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 ccy 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 ccy takes effect. |
| mmr | String | 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 instId takes effect) |
| baseMaxLoan | String | Base currency borrowing amount (only applicable to leverage and the case when instId takes effect) |

### Get interest rate and loan quota

Retrieve interest rate

#### Rate Limit: 2 requests per 2 seconds

#### Rate limit rule: IP

#### HTTP Request

GET /api/v5/public/interest-rate-loan-quota

Request Example

GET /api/v5/public/interest-rate-loan-quota

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": [

{

"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"

}

],

"regular": [

{

"irDiscount": "",

"loanQuotaCoef": "1",

"level": "Lv1"

},

{

"irDiscount": "",

"loanQuotaCoef": "2",

"level": "Lv2"

}

]

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| basic | Array | Basic interest rate |
| > ccy | String | Currency |
| > rate | String | Daily rate |
| > quota | String | Max borrow |
| vip | Array | Interest info for vip users |
| > level | String | VIP Level, e.g. VIP1 |
| > loanQuotaCoef | String | Loan quota coefficient. Loan quota = quota \* level |
| > irDiscount | String | (Deprecated) |
| regular | Array | Interest info for regular users |
| > level | String | Regular user Level, e.g. Lv1 |
| > loanQuotaCoef | String | Loan quota coefficient. Loan quota = quota \* level |
| > irDiscount | String | (Deprecated) |

### Get underlying

#### Rate Limit: 20 requests per 2 seconds

#### Rate limit rule: IP

#### HTTP Request

GET /api/v5/public/underlying

Request Example

GET /api/v5/public/underlying?instType=FUTURES

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 | Instrument type SWAP FUTURES OPTION |

Response Example

{

"code":"0",

"msg":"",

"data":[

[

"LTC-USDT",

"BTC-USDT",

"ETC-USDT"

]

]

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| uly | Array | Underlying |

### Get insurance fund

Get insurance fund balance information

#### Rate Limit: 10 requests per 2 seconds

#### Rate limit rule: IP

#### HTTP Request

GET /api/v5/public/insurance-fund

Request Example

GET /api/v5/public/insurance-fund?instType=SWAP&uly=BTC-USD

import okx.PublicData as PublicData

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

publicDataAPI = PublicData.PublicAPI(flag=flag)

# Get insurance fund balance information

result = publicDataAPI.get\_insurance\_fund(

instType="SWAP",

uly="BTC-USD"

)

print(result)

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instType | String | Yes | Instrument type MARGIN SWAP FUTURES OPTION |
| type | String | No | Type regular\_update  liquidation\_balance\_deposit bankruptcy\_loss platform\_revenue  adl: ADL historical data  The default is all type |
| uly | String | Conditional | Underlying Required for FUTURES/SWAP/OPTION Either uly or instFamily is required. If both are passed, instFamily will be used. |
| instFamily | String | Conditional | Instrument family Required for FUTURES/SWAP/OPTION Either uly or instFamily is required. If both are passed, instFamily will be used. |
| ccy | String | Conditional | Currency, only applicable to MARGIN |
| before | String | No | Pagination of data to return records newer than the requested ts |
| after | String | No | Pagination of data to return records earlier than the requested ts |
| limit | String | No | Number of results per request. The maximum is 100; The default is 100 |

Response Example

{

"code": "0",

"data": [

{

"details": [

{

"adlType": "",

"amt": "",

"balance": "1343.1308",

"ccy": "ETH",

"decRate": "",

"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 insurance fund, in USD |
| instFamily | String | Instrument family Applicable to FUTURES/SWAP/OPTION |
| instType | String | Instrument type |
| details | Array of objects | Insurance fund data |
| > balance | String | The balance of insurance fund |
| > amt | String | The change in the balance of insurance fund  Applicable when type is liquidation\_balance\_deposit, bankruptcy\_loss or platform\_revenue |
| > ccy | String | The currency of insurance fund |
| > type | String | The type of insurance fund |
| > maxBal | String | Maximum insurance fund balance in the past eight hours  Only applicable when type is adl |
| > maxBalTs | String | Timestamp when insurance fund balance reached maximum in the past eight hours, Unix timestamp format in milliseconds, e.g. 1597026383085  Only applicable when type is adl |
| > decRate | String | Real-time insurance fund decline rate (compare balance and maxBal)  Only applicable when type is adl |
| > adlType | String | ADL related events  rate\_adl\_start: ADL begins due to high insurance fund decline rate  bal\_adl\_start: ADL begins due to insurance fund balance falling  pos\_adl\_start：ADL begins due to the volume of liquidation orders falls to a certain level (only applicable to premarket symbols)  adl\_end: ADL ends  Only applicable when type is adl |
| > ts | String | The update timestamp of insurance fund. Unix timestamp format in milliseconds, e.g. 1597026383085 |

The enumeration value `regular\_update` of type field is used to present up-to-minute insurance fund change. The amt field will be used to present the difference of insurance 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

GET /api/v5/public/convert-contract-coin?instId=BTC-USD-SWAP&px=35000&sz=0.888

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 FUTURES/SWAP/OPTION |
| 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 coin usds: USDT/USDC only applicable to USDⓈ-margined contracts from FUTURES/SWAP |
| opType | String | No | Order type open: round down sz when opening positions  close: round sz to the nearest when closing positions  The default is close  Applicable to FUTURES SWAP |

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

GET /api/v5/public/instrument-tick-bands?instType=OPTION

#### 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 | String | 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/v5/public/premium-history

Request Example

GET /api/v5/public/premium-history?instId=BTC-USDT-SWAP

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instId | String | Yes | Instrument ID, e.g. BTC-USDT-SWAP Applicable to SWAP |
| 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 100. The default is 100. |

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. BTC-USDT-SWAP |
| premium | String | Premium between the mid price of perps market and the index price |
| ts | String | Data generation time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### 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

GET /api/v5/market/index-tickers?instId=BTC-USDT

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 USD/USDT/BTC/USDC as the quote currency. |
| instId | String | Conditional | Index, e.g. BTC-USD Either quoteCcy or instId is required. |

Response Example

{

"code": "0",

"msg": "",

"data": [

{

"instId": "BTC-USDT",

"idxPx": "43350",

"high24h": "43649.7",

"sodUtc0": "43444.1",

"open24h": "43640.8",

"low24h": "43261.9",

"sodUtc8": "43328.7",

"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. 1597026383085 |

### 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

GET /api/v5/market/index-candles?instId=BTC-USD

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. BTC-USD |
| 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]  Hong Kong time opening price k-line: [6H/12H/1D/1W/1M/3M] UTC time opening price k-line: [6Hutc/12Hutc/1Dutc/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",

"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. 1597026383085 |
| 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 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

GET /api/v5/market/history-index-candles?instId=BTC-USD

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instId | String | Yes | Index, e.g. BTC-USD |
| 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]  Hong Kong time opening price k-line: [6H/12H/1D/1W/1M] UTC time opening price k-line: [/6Hutc/12Hutc/1Dutc/1Wutc/1Mutc] |
| 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",

"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. 1597026383085 |
| 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 mark price candlesticks

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

GET /api/v5/market/mark-price-candles?instId=BTC-USD-SWAP

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. BTC-USD-SWAP |
| 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]  Hong Kong time opening price k-line: [6H/12H/1D/1W/1M/3M] UTC time opening price k-line: [6Hutc/12Hutc/1Dutc/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",

"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. 1597026383085 |
| 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: 10 requests per 2 seconds

#### Rate limit rule: IP

#### HTTP Request

GET /api/v5/market/history-mark-price-candles

Request Example

GET /api/v5/market/history-mark-price-candles?instId=BTC-USD-SWAP

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| instId | String | Yes | Instrument ID, e.g. BTC-USD-SWAP |
| 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]  Hong Kong time opening price k-line: [6H/12H/1D/1W/1M] UTC time opening price k-line: [6Hutc/12Hutc/1Dutc/1Wutc/1Mutc] |
| 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",

"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. 1597026383085 |
| 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 oracle

Get the crypto price of signing using Open Oracle smart contract.

#### Rate Limit: 1 request per 5 seconds

#### Rate limit rule: IP

#### HTTP Request

GET /api/v5/market/open-oracle

Request Example

GET /api/v5/market/open-oracle

import okx.MarketData as MarketData

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

marketDataAPI = MarketData.MarketAPI(flag=flag)

# Get the crypto price of signing using Open Oracle smart contract

result = marketDataAPI.get\_oracle()

print(result)

Response Example

{

"code":"0",

"msg":"",

"data":[

{

"messages":[

"0x000000000000000000000000000000000000000000000000000000000000008000000000000000000000000000000000000000000000000000000000616d3b1400000000000000000000000000000000000000000000000000000000000000c00000000000000000000000000000000000000000000000000000000e70528b800000000000000000000000000000000000000000000000000000000000000006707269636573000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000034254430000000000000000000000000000000000000000000000000000000000"

],

"prices":{

"BTC":"62014"

},

"signatures":[

"0xf18330e59eaf42373c2c40f1f9e24113ba21d4ed734dd3ed3bc1d12290fa74ba5623fca1113c5d245a1202dc065e333615b90f810f12132ce4a1ecacb8c6b24a000000000000000000000000000000000000000000000000000000000000001b"

],

"timestamp":"1634548500"

}

]

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| messages | String | ABI-encoded values [kind, timestamp, key, value], where kind equals 'prices', timestamp is the time when price was obtained, key is the asset ticker (e.g. btc) and value is the asset price. |
| prices | String | Readable asset prices |
| signatures | String | Ethereum-compatible ECDSA signatures for each message |
| timestamp | String | Time of latest datapoint, Unix timestamp, e.g. 1597026387 |

OKX Oracle public key is 85615b076615317c80f14cbad6501eec031cd51c

### 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

GET /api/v5/market/exchange-rate

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

GET /api/v5/market/index-components?index=BTC-USD

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 BTC-USDT |

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": "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. 1597026383085 |
| components | String | 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

GET /api/v5/public/economic-calendar

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| region | string | No | Country, region or entity  afghanistan, albania, algeria, andorra, angola, antigua\_and\_barbuda, argentina, armenia, aruba, australia, austria, azerbaijan, bahamas, bahrain, bangladesh, barbados, belarus, belgium, belize, benin, bermuda, bhutan, bolivia, bosnia\_and\_herzegovina, botswana, brazil, brunei, bulgaria, burkina\_faso, burundi, cambodia, cameroon, canada, cape\_verde, cayman\_islands, central\_african\_republic, chad, chile, china, colombia, comoros, congo, costa\_rica, croatia, cuba, cyprus, czech\_republic, denmark, djibouti, dominica, dominican\_republic, east\_timor, ecuador, egypt, el\_salvador, equatorial\_guinea, eritrea, estonia, ethiopia, euro\_area, european\_union, faroe\_islands, fiji, finland, france, g20, g7, gabon, gambia, georgia, germany, ghana, greece, greenland, grenada, guatemala, guinea, guinea\_bissau, guyana, hungary, haiti, honduras, hong\_kong, hungary, imf, indonesia, iceland, india, indonesia, iran, iraq, ireland, isle\_of\_man, israel, italy, ivory\_coast, jamaica, japan, jordan, kazakhstan, kenya, kiribati, kosovo, kuwait, kyrgyzstan, laos, latvia, lebanon, lesotho, liberia, libya, liechtenstein, lithuania, luxembourg, macau, macedonia, madagascar, malawi, malaysia, maldives, 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 | Level of importance  1: low  2: medium  3: high |
| 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": "1700121600000",

"dateSpan": "0",

"event": "Harmonised Inflation Rate YoY",

"forecast": "7.8%",

"importance": "1",

"prevInitial": "",

"previous": "9%",

"refDate": "1698710400000",

"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 |
| 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 |

## 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](https://www.okx.com/docs-v5/log_en/#2022-12-06))

#### URL Path

/ws/v5/public

Request Example

{

"op": "subscribe",

"args": [

{

"channel": "instruments",

"instType": "SPOT"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name instruments |
| > instType | String | Yes | Instrument type SPOT MARGIN SWAP FUTURES OPTION |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "instruments",

"instType": "SPOT"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"instruments\", \"instType\" : \"FUTURES\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name |
| > instType | String | Yes | Instrument type SPOT MARGIN SWAP FUTURES OPTION |
| 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",

"ctMult": "",

"ctType": "",

"ctVal": "",

"ctValCcy": "",

"expTime": "",

"instFamily": "",

"instId": "BTC-USDT",

"instType": "SPOT",

"lever": "10",

"listTime": "1606468572000",

"lotSz": "0.00000001",

"maxIcebergSz": "9999999999.0000000000000000",

"maxLmtAmt": "1000000",

"maxLmtSz": "9999999999",

"maxMktAmt": "1000000",

"maxMktSz": "",

"maxStopSz": "",

"maxTriggerSz": "9999999999.0000000000000000",

"maxTwapSz": "9999999999.0000000000000000",

"minSz": "0.00001",

"optType": "",

"quoteCcy": "USDT",

"settleCcy": "",

"state": "live",

"ruleType": "normal",

"stk": "",

"tickSz": "0.1",

"uly": ""

}

]

}

#### Push data parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| arg | Object | Subscribed channel |
| > channel | String | Channel name |
| > instType | String | Instrument type |
| data | Array | Subscribed data |
| > instType | String | Instrument type |
| > instId | String | Instrument ID, e.g. BTC-UST |
| > uly | String | Underlying, e.g. BTC-USD  Only applicable to FUTURES/SWAP/OPTION |
| > instFamily | String | Instrument family, e.g. BTC-USD  Only applicable to FUTURES/SWAP/OPTION |
| > category | String | Currency category. Note: this parameter is already deprecated |
| > baseCcy | String | Base currency, e.g. BTC in BTC-USDT  Only applicable to SPOT/MARGIN |
| > quoteCcy | String | Quote currency, e.g. USDT in BTC-USDT  Only applicable to SPOT/MARGIN |
| > settleCcy | String | Settlement and margin currency, e.g. BTC  Only applicable to FUTURES/SWAP/OPTION |
| > ctVal | String | Contract value |
| > ctMult | String | Contract multiplier |
| > ctValCcy | String | Contract value currency |
| > optType | String | Option type C: Call P: Put Only applicable to OPTION |
| > stk | String | Strike price Only applicable to OPTION |
| > listTime | String | Listing time Only applicable to FUTURES/SWAP/OPTION |
| > auctionEndTime | String | The end time of call auction, Unix timestamp format in milliseconds, e.g. 1597026383085  Only applicable to SPOT that are listed through call auctions, return "" in other cases |
| > expTime | String | Expiry time Applicable to SPOT/MARGIN/FUTURES/SWAP/OPTION. For FUTURES/OPTION, 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 SPOT/OPTION, used to distinguish between MARGIN and SPOT. |
| > tickSz | String | Tick size, e.g. 0.0001 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 SPOT/MARGIN, the value is the quantity in base currency |
| > minSz | String | Minimum order size If it is a derivatives contract, the value is the number of contracts. If it is SPOT/MARGIN, the value is the quantity in base currency |
| > ctType | String | Contract type linear: linear contract inverse: inverse contract Only applicable to FUTURES/SWAP |
| > alias | String | Alias this\_week next\_week this\_month next\_month quarter next\_quarter Only applicable to FUTURES  **Not recommended for use, users are encouraged to rely on the expTime field to determine the delivery time of the contract** |
| > state | String | Instrument status live suspend expired preopen. e.g. There will be preopen before the Futures and Options new contracts state is live.  test: Test pairs, can't be traded |
| > state | String | Instrument status live suspend expired preopen e.g. Futures and options contracts rollover from generation to trading start; certain symbols before they go live test: Test pairs, can't be traded |
| > ruleType | String | Trading rule types normal: normal trading pre\_market: pre-market trading |
| > 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 SPOT/MARGIN, the value is the quantity in base currency. |
| > 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 SPOT/MARGIN, the value is the quantity in USDT. |
| > 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 SPOT/MARGIN, the value is the quantity in base currency. |
| > 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 SPOT/MARGIN, the value is the quantity in base currency. |
| > 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 SPOT/MARGIN, the value is the quantity in base currency. |
| > 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 SPOT/MARGIN, the value is the quantity in USDT. |

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 auctionEndTime  
For spot symbols listed through a call auction, listTime represents the start time of the auction, and auctionEndTime indicates the end of the auction and the start of continuous trading. For other scenarios, listTime will mark the beginning of continuous trading, and auctionEndTime 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

{

"op": "subscribe",

"args": [

{

"channel": "open-interest",

"instId": "LTC-USD-SWAP"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name open-interest |
| > instId | String | Yes | Instrument ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "open-interest",

"instId": "LTC-USD-SWAP"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"open-interest\", \"instId\" : \"LTC-USD-SWAP\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| 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": "open-interest",

"instId": "LTC-USD-SWAP"

},

"data": [

{

"instType": "SWAP",

"instId": "LTC-USD-SWAP",

"oi": "5000",

"oiCcy": "555.55",

"ts": "1597026383085"

}

]

}

#### Push data parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| arg | Object | Successfully subscribed channel |
| > channel | String | Channel name |
| > instId | String | Instrument ID |
| data | Array | Subscribed data |
| > instType | String | Instrument type |
| > instId | String | Instrument ID, e.g. LTC-USD-SWAP |
| > oi | String | Open interest, in units of contracts. |
| > oiCcy | String | Open interest, in currency units |
| > ts | String | The time when the data was updated, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### Funding rate channel

Retrieve funding rate. Data will be pushed in 30s to 90s.

#### URL Path

/ws/v5/public

Request Example

{

"op": "subscribe",

"args": [

{

"channel": "funding-rate",

"instId": "BTC-USD-SWAP"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name funding-rate |
| > instId | String | Yes | Instrument ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "funding-rate",

"instId": "BTC-USD-SWAP"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"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** |
| --- | --- | --- | --- |
| 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":[

{

"fundingRate":"0.0001875391284828",

"fundingTime":"1700726400000",

"instId":"BTC-USD-SWAP",

"instType":"SWAP",

"method": "next\_period",

"maxFundingRate":"0.00375",

"minFundingRate":"-0.00375",

"nextFundingRate":"0.0002608059239328",

"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 | Subscribed data |
| > instType | String | Instrument type, SWAP |
| > instId | String | Instrument ID, e.g. BTC-USD-SWAP |
| > method | String | Funding rate mechanism  current\_period  next\_period |
| > fundingRate | String | Current funding rate |
| > fundingTime | String | Funding time of the upcoming settlement, Unix timestamp format in milliseconds, e.g. 1597026383085. |
| > nextFundingRate | String | Forecasted funding rate for the next period |
| > nextFundingTime | String | Forecasted funding time for the next period, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| > 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 |
| > settState | String | Settlement state of funding rate  processing  settled |
| > settFundingRate | String | If settState = processing, it is the funding rate that is being used for current settlement cycle.  If settState = settled, it is the funding rate that is being used for previous settlement cycle |
| > premium | String | Premium between the mid price of perps market and the index price |
| > ts | String | Data return time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

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

{

"op": "subscribe",

"args": [

{

"channel": "price-limit",

"instId": "LTC-USD-190628"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name price-limit |
| > instId | String | Yes | Instrument ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "price-limit",

"instId": "LTC-USD-190628"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"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** |
| --- | --- | --- | --- |
| 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": "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 | Subscribed data |
| > instType | String | Instrument type |
| > instId | String | Instrument ID, e.g. BTC-USDT |
| > 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. 1597026383085 |
| > enabled | Boolean | Whether price limit is effective  true: the price limit is effective  false: 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

{

"op": "subscribe",

"args": [

{

"channel": "opt-summary",

"instFamily": "BTC-USD"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name opt-summary |
| > instFamily | String | Yes | Instrument family |

Response Example

{

"event": "subscribe",

"arg": {

"channel": "opt-summary",

"instFamily": "BTC-USD"

},

"connId": "a4d3ae55"

}

Failure example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"opt-summary\", \"uly\" : \"BTC-USD\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| 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": "0.0000000001",

"theta": "0.0000032334",

"lever": "8.465747567",

"markVol": "0.3675503331",

"bidVol": "0",

"askVol": "1.1669998535",

"realVol": "",

"deltaBS": "-0.9999672034",

"gammaBS": "0.0000000002",

"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 | Subscribed data |
| > instType | String | Instrument type, OPTION |
| > instId | String | Instrument ID |
| > 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 priceo 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 |
| > ts | String | Price update time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### Estimated delivery/exercise price channel

Retrieve the estimated delivery/exercise price of SWAP, FUTURES and OPTION contracts.

Only the estimated delivery/exercise price will be pushed an hour before delivery/exercise, and will be pushed if there is any price change.

#### URL Path

/ws/v5/public

Request Example

{

"op": "subscribe",

"args": [

{

"channel": "estimated-price",

"instType": "FUTURES",

"instFamily": "BTC-USD"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name estimated-price |
| > instType | String | Yes | Instrument type OPTION FUTURES SWAP |
| > instFamily | String | Conditional | Instrument family Either instFamily or instId is required. |
| > instId | String | Conditional | Instrument ID Either instFamily or instId is required. |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "estimated-price",

"instType": "FUTURES",

"instFamily": "BTC-USD"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"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** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name |
| > instType | String | Yes | Instrument type OPTION FUTURES SWAP |
| > 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": "BTC-USD"

},

"data": [

{

"instType": "FUTURES",

"instId": "BTC-USD-170310",

"settlePx": "200",

"ts": "1597026383085"

}

]

}

#### Push data parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| arg | Object | Successfully subscribed channel |
| > channel | String | Channel name |
| > instType | String | Instrument type FUTURES OPTION SWAP |
| > instFamily | String | Instrument family |
| > instId | String | Instrument ID |
| data | Array | Subscribed data |
| > instType | String | Instrument type |
| > instId | String | Instrument ID, e.g. BTC-USD-170310 |
| > settlePx | String | Estimated delivery/exercise price |
| > ts | String | Data update time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### 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

{

"op": "subscribe",

"args": [

{

"channel": "mark-price",

"instId": "LTC-USD-190628"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name mark-price |
| > instId | String | Yes | Instrument ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "mark-price",

"instId": "LTC-USD-190628"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"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** |
| --- | --- | --- | --- |
| 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": "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 | 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

{

"op": "subscribe",

"args": [

{

"channel": "index-tickers",

"instId": "BTC-USDT"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name index-tickers |
| > instId | String | Yes | Index with USD, USDT, BTC, USDC as the quote currency, e.g. BTC-USDT |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "index-tickers",

"instId": "BTC-USDT"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"index-tickers\", \"instId\" : \"BTC-USDT\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name index-tickers |
| > instId | String | Yes | Index with USD, USDT, BTC, USDC as the quote currency, e.g. BTC-USDT |
| 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. BTC-USDT. |
| data | Array | 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. 1597026383085 |

### 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

{

"op": "subscribe",

"args": [

{

"channel": "mark-price-candle1D",

"instId": "BTC-USD-190628"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name  mark-price-candle3M  mark-price-candle1M  mark-price-candle1W  mark-price-candle1D  mark-price-candle2D  mark-price-candle3D  mark-price-candle5D  mark-price-candle12H  mark-price-candle6H  mark-price-candle4H  mark-price-candle2H  mark-price-candle1H  mark-price-candle30m  mark-price-candle15m  mark-price-candle5m  mark-price-candle3m  mark-price-candle1m  mark-price-candle1Yutc  mark-price-candle3Mutc  mark-price-candle1Mutc  mark-price-candle1Wutc  mark-price-candle1Dutc  mark-price-candle2Dutc  mark-price-candle3Dutc  mark-price-candle5Dutc  mark-price-candle12Hutc  mark-price-candle6Hutc |
| > instId | String | Yes | Instrument ID |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "mark-price-candle1D",

"instId": "BTC-USD-190628"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"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** |
| --- | --- | --- | --- |
| 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": "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 | 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 |
| > 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

{

"op": "subscribe",

"args": [

{

"channel": "index-candle30m",

"instId": "BTC-USD"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name  index-candle3M  index-candle1M  index-candle1W  index-candle1D  index-candle2D  index-candle3D  index-candle5D  index-candle12H  index-candle6H  index-candle4H  index -candle2H  index-candle1H  index-candle30m  index-candle15m  index-candle5m  index-candle3m  index-candle1m  index-candle3Mutc  index-candle1Mutc  index-candle1Wutc  index-candle1Dutc  index-candle2Dutc  index-candle3Dutc  index-candle5Dutc  index-candle12Hutc  index-candle6Hutc |
| > instId | String | Yes | Index, e.g. BTC-USD |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "index-candle30m",

"instId": "BTC-USD"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"index-candle30m\", \"instId\" : \"BTC-USD\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | subscribe unsubscribe |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name |
| > instId | String | No | Index, e.g. BTC-USD |
| 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"

},

"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 | 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 |
| > 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

{

"op": "subscribe",

"args": [

{

"channel": "liquidation-orders",

"instType": "SWAP"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name liquidation-orders |
| > instType | String | Yes | Instrument type SWAP FUTURES MARGIN OPTION |

Response Example

{

"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": "IOST-USDT-SWAP",

"instType": "SWAP",

"uly": "IOST-USDT"

}

]

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| arg | Object | Successfully subscribed channel |
| > channel | String | Channel name |
| > instId | String | Instrument ID |
| data | Array | Subscribed data |
| > instType | String | Instrument type |
| > instId | String | Instrument ID, e.g. BTC-USD-SWAP |
| > uly | String | Underlying, only applicable to FUTURES/SWAP/OPTION |
| > details | Array | Liquidation details |
| >> side | String | Order side, buy sell, only applicable to FUTURES/SWAP |
| >> posSide | String | Position side, long short Combination of side and posSide, sell/long : Close long ; buy/short:Close short, only applicable to FUTURES/SWAP |
| >> bkPx | String | Bankruptcy price. The price of the transaction with the system's liquidation account, only applicable to FUTURES/SWAP |
| >> sz | String | Quantity of liquidation, only applicable to MARGIN/FUTURES/SWAP |
| >> bkLoss | String | Bankruptcy loss |
| >> ccy | String | Liquidation currency, only applicable to MARGIN |
| >> ts | String | Liquidation time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### ADL warning channel

Auto-deleveraging warning channel.

In the normal state, data will be pushed once every minute to display the balance of insurance fund and etc.

In the warning state or when there is ADL risk (warning/adl), data will be pushed every second to display information such as the real-time decline rate of insurance fund.

For more ADL details, please refer to [Introduction to Auto-deleveraging](https://www.okx.com/help/iv-introduction-to-auto-deleveraging-adl)

#### URL Path

/ws/v5/public

Request Example

{

"op": "subscribe",

"args": [{

"channel": "adl-warning",

"instType": "FUTURES",

"instFamily": "BTC-USDT"

}]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name adl-warning |
| > instType | String | Yes | Instrument type SWAP FUTURES OPTION |
| > instFamily | String | No | Instrument family |

Successful Response Example

{

"event":"subscribe",

"arg":{

"channel":"adl-warning",

"instType":"FUTURES",

"instFamily":"BTC-USDT"

},

"connId":"48d8960a"

}

Failure Response Example

{

"event":"error",

"msg":"Illegal request: { \"event\": \"subscribe\", \"arg\": { \"channel\": \"adl-warning\", \"instType\": \"FUTURES\", \"instFamily\": \"BTC-USDT\" } }",

"code":"60012",

"connId":"48d8960a"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name adl-warning |
| > 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":[

{

"decRate":"",

"maxBal":"",

"adlRecRate":"0.25",

"adlRecBal":"8000.0",

"bal":"280784384.9564228289548144",

"instType":"FUTURES",

"adlRate":"0.5",

"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 adl-warning |
| > instType | String | Instrument type |
| > instFamily | String | Instrument family |
| data | Array | Subscribed data |
| > instType | String | Instrument type |
| > instFamily | String | Instrument family |
| > state | String | state  normal  warning  adl |
| > bal | String | Real-time insurance fund balance |
| > ccy | String | The corresponding currency of insurance fund balance |
| > maxBal | String | Maximum insurance fund balance in the past eight hours   Applicable when state is warning or adl |
| > maxBalTs | String | Timestamp when insurance fund balance reached maximum in the past eight hours, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| > decRate | String | Real-time insurance fund decline rate (compare bal and maxBal)   Applicable when state is warning or adl |
| > adlType | String | ADL related events  rate\_adl\_start: ADL begins due to high insurance fund decline rate  bal\_adl\_start: ADL begins due to insurance fund balance falling  pos\_adl\_start：ADL begins due to the volume of liquidation orders falls to a certain level (only applicable to premarket symbols)  adl\_end: ADL ends |
| > adlBal | String | Insurance fund balance that triggers ADL |
| > adlRate | String | Insurance fund decline rate that triggers ADL |
| > adlRecBal | String | Insurance fund balance that turns off ADL |
| > adlRecRate | String | Insurance fund decline rate that turns off ADL |
| > ts | String | Data push time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### 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

{

"op": "subscribe",

"args": [

{

"channel": "economic-calendar"

}

]

}

#### Request parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name economic-calendar |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "economic-calendar"

}

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"economic-calendar\", \"instId\" : \"LTC-USD-190628\"}]}"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Event subscribe unsubscribe error |
| 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 |
| data | Array | 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 |
| > date | string | Estimated release time of the value of actual field, millisecond format of Unix timestamp, e.g. 1597026383085 |
| > 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 |
| > importance | string | Level of importance 1: low  2: medium  3: high |
| > 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

GET /api/v5/rubik/stat/trading-data/support-coin

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 string | Currency supported by derivatives trading data |
| option | Array of string | Currency supported by option trading data |
| spot | Array of string | 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 + instrumentID

#### HTTP Request

GET /api/v5/rubik/stat/contracts/open-interest-history

Request example

GET /api/v5/rubik/stat/contracts/open-interest-history?instId=BTC-USDT-SWAP

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 FUTURES, SWAP |
| period | string | No | Bar size, the default is 5m, e.g. [5m/15m/30m/1H/2H/4H]  Hong Kong time opening price k-line: [6H/12H/1D/2D/3D/5D/1W/1M/3M]  UTC time opening price k-line: [6Hutc/12Hutc/1Dutc/2Dutc/3Dutc/5Dutc/1Wutc/1Mutc/3Mutc] |
| end | string | No | Pagination of data to return records earlier than the requested ts |
| begin | string | No | return records newer than the requested ts |
| limit | string | No | Number of results per request. The maximum is 100. The default is 100. |

Response example

{

"code":"0",

"msg":"",

"data":[

[

"1701417600000", // timestamp

"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. 1597026383085 |
| 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

GET /api/v5/rubik/stat/taker-volume?ccy=BTC&instType=SPOT

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 SPOT CONTRACTS |
| begin | String | No | Begin time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| end | String | No | End time, Unix timestamp format in milliseconds, e.g. 1597026383011 |
| period | String | No | Period, the default is 5m, e.g. [5m/1H/1D]  5m granularity can only query data within two days at most 1H granularity can only query data within 30 days at most  1D granularity can only query data within 180 days at most |

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 + instrumentID

#### HTTP Request

GET /api/v5/rubik/stat/taker-volume-contract

Request example

GET /api/v5/rubik/stat/taker-volume-contract?instId=BTC-USDT-SWAP

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 FUTURES, SWAP |
| period | string | No | Bar size, the default is 5m, e.g. [5m/15m/30m/1H/2H/4H]  Hong Kong time opening price k-line:[6H/12H/1D/2D/3D/5D/1W/1M/3M]  UTC time opening price k-line: [6Hutc/12Hutc/1Dutc/2Dutc/3Dutc/5Dutc/1Wutc/1Mutc/3Mutc] |
| unit | string | No | The unit of buy/sell volume, the default is 1  0: Crypto  1: Contracts  2: U |
| end | string | No | return records earlier than the requested ts |
| begin | string | No | return records newer than the requested ts |
| limit | string | No | Number of results per request. The maximum is 100. The default is 100. |

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. 1597026383085 |
| 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 lending 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

GET /api/v5/rubik/stat/margin/loan-ratio?ccy=BTC

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. 1597026383085 |
| end | String | No | End time, e.g. 1597026383085 |
| period | String | No | Period m: Minute, H: Hour, D: Day the default is 5m, e.g. [5m/1H/1D]  5m granularity can only query data within two days at most 1H granularity can only query data within 30 days at most 1D 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 + instrumentID

#### HTTP Request

GET /api/v5/rubik/stat/contracts/long-short-account-ratio-contract-top-trader

Request Example

GET /api/v5/rubik/stat/contracts/long-short-account-ratio-contract-top-trader?instId=BTC-USDT-SWAP

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 FUTURES, SWAP |
| period | string | No | Bar size, the default is 5m, e.g. [5m/15m/30m/1H/2H/4H]  Hong Kong time opening price k-line: [6H/12H/1D/2D/3D/5D/1W/1M/3M]  UTC time opening price k-line: [6Hutc/12Hutc/1Dutc/2Dutc/3Dutc/5Dutc/1Wutc/1Mutc/3Mutc] |
| end | string | No | return records earlier than the requested ts |
| begin | string | No | return records newer than the requested ts |
| limit | string | No | Number of results per request. The maximum is 100. The default is 100. |

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. 1597026383085 |
| 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

#### Rate limit rule: IP + instrumentID

#### HTTP Request

GET /api/v5/rubik/stat/contracts/long-short-position-ratio-contract-top-trader

Request example

GET /api/v5/rubik/stat/contracts/long-short-position-ratio-contract-top-trader?instId=BTC-USDT-SWAP

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, eg: BTC-USDT-SWAP  Only applicable to FUTURES, SWAP |
| period | string | No | Bar size, the default is 5m, e.g. [5m/15m/30m/1H/2H/4H]  Hong Kong time opening price k-line: [6H/12H/1D/2D/3D/5D/1W/1M/3M]  UTC time opening price k-line: [6Hutc/12Hutc/1Dutc/2Dutc/3Dutc/5Dutc/1Wutc/1Mutc/3Mutc] |
| end | string | No | return records earlier than the requested ts |
| begin | string | No | return records newer than the requested ts |
| limit | string | No | Number of results per request. The maximum is 100. The default is 100. |

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. 1597026383085 |
| 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.

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 + instrumentID

#### HTTP Request

GET /api/v5/rubik/stat/contracts/long-short-account-ratio-contract

Request example

GET /api/v5/rubik/stat/contracts/long-short-account-ratio-contract?instId=BTC-USDT-SWAP

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 FUTURES, SWAP |
| period | string | No | Bar size, the default is 5m, e.g. [5m/15m/30m/1H/2H/4H]  Hong Kong time opening price k-line:[6H/12H/1D/2D/3D/5D/1W/1M/3M]  UTC time opening price k-line: [6Hutc/12Hutc/1Dutc/2Dutc/3Dutc/5Dutc/1Wutc/1Mutc/3Mutc] |
| end | string | No | return records earlier than the requested ts |
| begin | string | No | return records newer than the requested ts |
| limit | string | No | Number of results per request. The maximum is 100. The default is 100. |

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. 1597026383085 |
| 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

GET /api/v5/rubik/stat/contracts/long-short-account-ratio?ccy=BTC

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. 1597026383085 |
| end | String | No | End time, e.g. 1597026383011 |
| period | String | No | Period, the default is 5m, e.g. [5m/1H/1D]  5m granularity can only query data within two days at most 1H granularity can only query data within 30 days at most  1D 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: 5 requests per 2 seconds

#### Rate limit rule: IP

#### HTTP Request

GET /api/v5/rubik/stat/contracts/open-interest-volume

Request Example

GET /api/v5/rubik/stat/contracts/open-interest-volume?ccy=BTC

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. 1597026383085 |
| end | String | No | End time, e.g. 1597026383011 |
| period | String | No | Period, the default is 5m, e.g. [5m/1H/1D]  5m granularity can only query data within two days at most 1H granularity can only query data within 30 days at most  1D 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: 5 requests per 2 seconds

#### Rate limit rule: IP

#### HTTP Request

GET /api/v5/rubik/stat/option/open-interest-volume

Request Example

GET /api/v5/rubik/stat/option/open-interest-volume?ccy=BTC

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.

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: IP

#### HTTP Request

GET /api/v5/rubik/stat/option/open-interest-volume-ratio

Request Example

GET /api/v5/rubik/stat/option/open-interest-volume-ratio?ccy=BTC

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

GET /api/v5/rubik/stat/option/open-interest-volume-expiry?ccy=BTC

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]

### Get open interest and volume (strike)

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

GET /api/v5/rubik/stat/option/open-interest-volume-strike?ccy=BTC&expTime=20210901

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 YYYYMMdd, e.g. 20210623 |
| 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",

"10000",

"0",

"0.5",

"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 (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,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

GET /api/v5/rubik/stat/option/taker-block-volume?ccy=BTC

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": [

"1630512000000",

"8.55",

"67.3",

"16.05",

"16.3",

"126.4",

"40.7"

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ts | String | Timestamp |
| 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: UserID

#### HTTP Request

GET /api/v5/asset/currencies

Request Example

GET /api/v5/asset/currencies

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. BTC or BTC,ETH. |

Response Example

{

"code": "0",

"msg": "",

"data": [

{

"burningFeeRate": "",

"canDep": true,

"canInternal": true,

"canWd": true,

"ccy": "BTC",

"chain": "BTC-Bitcoin",

"ctAddr": "",

"depEstOpenTime": "",

"depQuotaFixed": "",

"depQuoteDailyLayer2": "",

"fee": "0.00005",

"logoLink": "https://static.coinall.ltd/cdn/oksupport/asset/currency/icon/btc20230419112752.png",

"mainNet": true,

"maxFee": "0.00005",

"maxFeeForCtAddr": "",

"maxWd": "500",

"minDep": "0.0005",

"minDepArrivalConfirm": "1",

"minFee": "0.00005",

"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. BTC |
| 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. USDT-ERC20, USDT-TRC20 |
| ctAddr | String | Contract address |
| canDep | Boolean | The availability to deposit from chain  false: not available  true: available |
| canWd | Boolean | The availability to withdraw to chain  false: not available  true: available |
| canInternal | Boolean | The availability to internal transfer  false: not available  true: available |
| depEstOpenTime | String | Estimated opening time for deposit, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| wdEstOpenTime | String | Estimated opening time for withdraw, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| minDep | String | The minimum deposit amount of currency in a single transaction |
| minWd | String | The minimum on-chain withdrawal amount of currency in a single transaction |
| minInternal | String | The minimum internal transfer amount of currency in a single transaction No maximum internal transfer limit in a single transaction, subject to the withdrawal limit in the past 24 hours(wdQuota). |
| maxWd | String | The maximum amount of currency on-chain withdrawal 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 on-chain withdrawal and internal transfer), unit in USD |
| usedWdQuota | String | The amount of currency withdrawal used in the past 24 hours, unit in USD |
| fee | String | The fixed withdrawal fee Apply to on-chain withdrawal |
| minFee | String | (Deprecated) |
| maxFee | String | (Deprecated) |
| minFeeForCtAddr | String | (Deprecated) |
| maxFeeForCtAddr | String | (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 on-chain withdrawal |
| mainNet | Boolean | If current chain is main net, then it will return true, otherwise it will return false |
| needTag | Boolean | Whether tag/memo information is required for withdrawal, e.g. EOS will return true |
| 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 USD Return empty string if there is no deposit limit |
| usedDepQuotaFixed | String | The used amount of fixed deposit quota, unit in USD Return empty string if there is no deposit limit |
| depQuoteDailyLayer2 | String | The layer2 network daily deposit limit |

### Get balance

Retrieve the funding account balances of all the assets and the amount that is available or on hold.

Only asset information of a currency with a balance greater than 0 will be returned.

#### Rate Limit: 6 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/asset/balances

Request Example

GET /api/v5/asset/balances

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 balane

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. BTC or BTC,ETH. |

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-tradable assets

#### Rate Limit: 6 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/asset/non-tradable-assets

Request Example

GET /api/v5/asset/non-tradable-assets

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. BTC or BTC,ETH. |

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. CELT |
| 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.  false: not available true: 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. USDT |
| 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: UserID

#### HTTP Request

GET /api/v5/asset/asset-valuation

Request Example

GET /api/v5/asset/asset-valuation

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](https://www.okx.com/docs-v5/en/#sub-account-rest-api-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.)

Failure of the request does not mean the transfer has failed. Recommend to call "Get funds transfer state" to confirm the status.

#### Rate Limit: 2 requests per second

#### Rate limit rule: UserID + Currency

#### HTTP Request

POST /api/v5/asset/transfer

Request Example

# Transfer 1.5 USDT from funding account to Trading account when current account is master-account

POST /api/v5/asset/transfer

body

{

"ccy":"USDT",

"amt":"1.5",

"from":"6",

"to":"18"

}

# Transfer 1.5 USDT from funding account to subAccount when current account is master-account

POST /api/v5/asset/transfer

body

{

"ccy":"USDT",

"type":"1",

"amt":"1.5",

"from":"6",

"to":"6",

"subAcct":"mini"

}

# Transfer 1.5 USDT from funding account to subAccount when current account is sub-account

POST /api/v5/asset/transfer

body

{

"ccy":"USDT",

"type":"4",

"amt":"1.5",

"from":"6",

"to":"6",

"subAcct":"mini"

}

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 | Transfer type 0: transfer within account 1: master account to sub-account (Only applicable to API Key from master account) 2: sub-account to master account (Only applicable to API Key from master account) 3: sub-account to master account (Only applicable to APIKey from sub-account) 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](https://www.okx.com/docs-v5/en/#sub-account-rest-api-set-permission-of-transfer-out)) The default is 0. If you want to make transfer between sub-accounts by master account API key, refer to [Master accounts manage the transfers between sub-accounts](https://www.okx.com/docs-v5/en/#sub-account-rest-api-master-accounts-manage-the-transfers-between-sub-accounts) |
| ccy | String | Yes | Transfer currency, e.g. USDT |
| amt | String | Yes | Amount to be transferred |
| from | String | Yes | The remitting account 6: Funding account 18: Trading account |
| to | String | Yes | The beneficiary account 6: Funding account 18: Trading account |
| subAcct | String | Conditional | Name of the sub-account When type is 1/2/4, this parameter is required. |
| loanTrans | Boolean | No | Whether or not borrowed coins can be transferred out under Spot mode/Multi-currency margin/Portfolio margin true: borrowed coins can be transferred out false: borrowed coins cannot be transferred out the default is false |
| omitPosRisk | String | No | Ignore position risk Default is false Applicable to Portfolio margin |
| clientId | String | No | 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": [

{

"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: UserID

#### HTTP Request

GET /api/v5/asset/transfer-state

Request Example

GET /api/v5/asset/transfer-state?transId=1&type=1

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 | Transfer type 0: transfer within account  1: master account to sub-account (Only applicable to API Key from master account)  2: sub-account to master account (Only applicable to API Key from master account) 3: sub-account to master account (Only applicable to APIKey from sub-account) 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) The default is 0. For Custody accounts, can choose not to pass this parameter or pass 0. |

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 | Transfer type 0: transfer within account 1: master account to sub-account (Only applicable to API Key from master account)  2: sub-account to master account (Only applicable to APIKey from master account) 3: sub-account to master account (Only applicable to APIKey from sub-account) 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) |
| from | String | The remitting account 6: Funding account 18: Trading account |
| to | String | The beneficiary account 6: Funding account 18: Trading account |
| subAcct | String | Name of the sub-account |
| instId | String | deprecated |
| toInstId | 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: UserID

#### HTTP Request

GET /api/v5/asset/bills

Request Example

GET /api/v5/asset/bills

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 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 61: [Convert] Exchange between crypto 68: Fee rebate (by rebate card) 72: Token received 73: Token given away 74: Token refunded 75: Subscription to savings 76: Redemption to savings 77: Jumpstart distribute 78: Jumpstart lock up 80: DEFI/Staking purchase 82: DEFI/Staking redemption 83: Staking yield 84: Violation fee 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 136: ETH 2.0 staking system account increase ETH (for on-chain operation) 137: ETH 2.0 Subscription 138: ETH 2.0 Swapping 139: ETH 2.0 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 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 |
| 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 ts, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| before | String | No | Pagination of data to return records newer than the requested ts, 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",

"msg": "",

"data": [{

"billId": "12344",

"ccy": "BTC",

"clientId": "",

"balChg": "2",

"bal": "12",

"type": "1",

"ts": "1597026383085"

}]

}

#### 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 |
| ts | String | Creation time, Unix timestamp format in milliseconds, e.g.1597026383085 |

### Get deposit address

Retrieve the deposit addresses of currencies, including previously-used addresses.

#### Rate Limit: 6 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/asset/deposit-address

Request Example

GET /api/v5/asset/deposit-address?ccy=BTC

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. BTC |

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. TONCOIN attached tag name is comment, the return will be {'comment':'123456'} |
| ccy | String | Currency, e.g. BTC |
| chain | String | Chain name, e.g. USDT-ERC20, USDT-TRC20 |
| to | String | The beneficiary account 6: Funding account 18: 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 true 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](https://www.okx.com/docs-v5/en/#funding-account-websocket-deposit-info-channel).

#### Rate Limit: 6 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/asset/deposit-history

Request Example

GET /api/v5/asset/deposit-history

# Query deposit history from 2022-06-01 to 2022-07-01

GET /api/v5/asset/deposit-history?ccy=BTC&after=1654041600000&before=1656633600000

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. BTC |
| 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 3: internal transfer 4: deposit from chain |
| state | String | No | Status of deposit  0: waiting for confirmation 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 |
| after | String | No | Pagination of data to return records earlier than the requested ts, Unix timestamp format in milliseconds, e.g. 1654041600000 |
| before | String | No | Pagination of data to return records newer than the requested ts, Unix timestamp format in milliseconds, e.g. 1656633600000 |
| limit | string | No | Number of results per request. The maximum is 100; The default is 100 |

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, email address, account name, and will return "" in other cases |
| areaCodeFrom | String | If from 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. 1655251200000 |
| state | String | Status of deposit 0: Waiting for confirmation 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 |
| 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 |

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)

#### Rate Limit: 6 requests per second

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/asset/withdrawal

Request Example

# on-chain withdrawal

POST /api/v5/asset/withdrawal

body

{

"amt":"1",

"dest":"4",

"ccy":"BTC",

"chain":"BTC-Bitcoin",

"toAddr":"17DKe3kkkkiiiiTvAKKi2vMPbm1Bz3CMKw"

}

# internal withdrawal

POST /api/v5/asset/withdrawal

body

{

"amt":"10",

"dest":"3",

"ccy":"USDT",

"areaCode":"86",

"toAddr":"15651000000"

}

# Specific entity users need to provide receiver's info

POST /api/v5/asset/withdrawal

body

{

"amt":"1",

"dest":"4",

"ccy":"BTC",

"chain":"BTC-Bitcoin",

"toAddr":"17DKe3kkkkiiiiTvAKKi2vMPbm1Bz3CMKw",

"rcvrInfo":{

"walletType":"exchange",

"exchId":"did:ethr:0xfeb4f99829a9acdf52979abee87e83addf22a7e1",

"rcvrFirstName":"Bruce",

"rcvrLastName":"Wayne"

}

}

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="TXtvfb7cdrn6VX9H49mgio8bUxZ3DGfvYF",

amt="100",

dest="4",

chain="USDT-TRC20"

)

print(result)

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | Yes | Currency, e.g. USDT |
| amt | String | Yes | 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](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-currencies). For internal transfer, transaction fee is always 0. |
| dest | String | Yes | Withdrawal method 3: internal transfer 4: on-chain withdrawal |
| toAddr | String | Yes | toAddr should be a trusted address/account.  If your dest is 4, some crypto currency addresses are formatted as 'address:tag', e.g. 'ARDOR-7JF3-8F2E-QUWZ-CAN7F:123456' If your dest is 3,toAddr should be a recipient address which can be email, phone or login account name (account name is only for sub-account). |
| chain | String | Conditional | Chain name There are multiple chains under some currencies, such as USDT has USDT-ERC20, USDT-TRC20 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 on-chain withdrawal. You can get supported chain name by the endpoint of [Get currencies](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-currencies). |
| areaCode | String | Conditional | Area code for the phone number, e.g. 86 If toAddr is a phone number, this parameter is required. Apply to internal transfer |
| rcvrInfo | Object | Conditional | Recipient information For the specific entity users to do on-chain withdrawal/lightning withdrawal, this information is required. |
| > walletType | String | Yes | Wallet Type exchange: Withdraw to exchange wallet private: Withdraw to private wallet If withdrawal to the exchange wallet, relevant information about the recipient must be provided. For the exchange wallet belongs to business recipient, rcvrFirstName may input the company name, rcvrLastName may input "N/A", location info may input the registered address of the company. Withdrawal to a private wallet does not require providing recipient information. |
| > exchId | String | Conditional | Exchange ID You can query supported exchanges through the endpoint of [Get exchange list (public)](https://www.okx.com/docs-v5/en/#funding-account-rest-api-get-exchange-list-public) If the exchange is not in the exchange list, fill in '0' in this field.  Apply to walletType = exchange |
| > rcvrFirstName | String | Conditional | Receiver's first name, e.g. Bruce Apply to walletType = exchange |
| > rcvrLastName | String | Conditional | Receiver's last name, e.g. Wayne Apply to walletType = exchange |
| > rcvrCountry | String | Conditional | The recipient's country, e.g. United States You must enter an English country name or a two letter country code (ISO 3166-1). Please refer to the Country Name and Country Code in the country information table below. Apply to walletType = exchange |
| > rcvrCountrySubDivision | String | Conditional | State/Province of the recipient, e.g. California Apply to walletType = exchange |
| > rcvrTownName | String | Conditional | The town/city where the recipient is located, e.g. San Jose Apply to walletType = exchange |
| > rcvrStreetName | String | Conditional | Recipient's street address, e.g. Clementi Avenue 1 Apply to walletType = exchange |
| clientId | String | No | 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": [{

"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. USDT-ERC20, USDT-TRC20 |
| amt | String | Withdrawal amount |
| wdId | String | Withdrawal ID |
| clientId | String | Client-supplied ID A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters. |

#### 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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: UserID

#### HTTP Request

POST /api/v5/asset/cancel-withdrawal

Request Example

POST /api/v5/asset/cancel-withdrawal

body {

"wdId":"1123456"

}

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](https://www.okx.com/docs-v5/en/#funding-account-websocket-withdrawal-info-channel).

#### Rate Limit: 6 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/asset/withdrawal-history

Request Example

GET /api/v5/asset/withdrawal-history

# Query withdrawal history from 2022-06-01 to 2022-07-01

GET /api/v5/asset/withdrawal-history?ccy=BTC&after=1654041600000&before=1656633600000

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | No | Currency, e.g. BTC |
| 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 3: Internal transfer 4: On-chain withdrawal |
| state | String | No | Status of withdrawal   Stage 1 : Pending withdrawal10: Waiting transfer 0: Waiting withdrawal 4/5/6/8/9/12: Waiting manual review 7: Approved   Stage 2 : Withdrawal in progress (Applicable to on-chain withdrawals, internal transfers do not have this stage)1: Broadcasting your transaction to chain 15: Pending transaction validation 16: Due to local laws and regulations, your withdrawal may take up to 24 hours to arrive -3: Canceling   Final stage-2: Canceled  -1: Failed 2: Success |
| after | String | No | Pagination of data to return records earlier than the requested ts, Unix timestamp format in milliseconds, e.g. 1654041600000 |
| before | String | No | Pagination of data to return records newer than the requested ts, Unix timestamp format in milliseconds, e.g. 1656633600000 |
| limit | String | No | Number of results per request. The maximum is 100; The default is 100 |

Response Example

{

"code": "0",

"msg": "",

"data": [

{

"chain": "ETH-Ethereum",

"fee": "0.007",

"feeCcy": "ETH",

"ccy": "ETH",

"clientId": "",

"amt": "0.029809",

"txId": "0x35c\*\*\*\*\*\*b360a174d",

"from": "156\*\*\*\*359",

"areaCodeFrom": "86",

"to": "0xa30d1fab\*\*\*\*\*\*\*\*7CF18C7B6C579",

"areaCodeTo": "",

"state": "2",

"ts": "1655251200000",

"nonTradableAsset": false,

"wdId": "15447421"

}

]

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ccy | String | Currency |
| chain | String | Chain name, e.g. USDT-ERC20, USDT-TRC20 |
| nonTradableAsset | Boolean | Whether it is a non-tradable asset or not true: non-tradable asset, false: tradable asset |
| amt | String | Withdrawal amount |
| ts | String | Time the withdrawal request was submitted, Unix timestamp format in milliseconds, e.g. 1655251200000. |
| from | String | Withdrawal account  It can be email/phone/sub-account name |
| areaCodeFrom | String | Area code for the phone number If from 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 to is a phone number, this parameter returns the area code for the phone number |
| 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 {'comment':'123456'} |
| txId | 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. USDT |
| state | String | Status of withdrawal |
| wdId | String | Withdrawal ID |
| clientId | String | Client-supplied ID |

### 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: UserID

#### HTTP Request

GET /api/v5/asset/deposit-withdraw-status

Request Example

# For deposit

GET /api/v5/asset/deposit-withdraw-status?txId=xxxxxx&to=1672734730284&ccy=USDT&chain=USDT-ERC20

# For withdrawal

GET /api/v5/asset/deposit-withdraw-status?wdId=200045249

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| wdId | String | Conditional | Withdrawal ID, use to retrieve withdrawal status  Required to input one and only one of wdId and txId |
| txId | String | Conditional | Hash record of the deposit, use to retrieve deposit status  Required to input one and only one of wdId and txId |
| ccy | String | Conditional | Currency type, e.g. USDT  Required when retrieving deposit status with txId |
| to | String | Conditional | To address, the destination address in deposit  Required when retrieving deposit status with txId |
| chain | String | Conditional | Currency chain information, e.g. USDT-ERC20  Required when retrieving deposit status with txId |

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 UTC+8. 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 txId 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

GET /api/v5/asset/exchange-list

#### 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. 1xbet |
| exchId | String | Exchange ID, e.g. did:ethr:0xfeb4f99829a9acdf52979abee87e83addf22a7e1 |

### Apply for monthly statement (last year)

Apply for monthly statement in the past year.

#### Rate Limit: 20 requests per month

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/asset/monthly-statement

Request Example

POST /api/v5/asset/monthly-statement

body

{

"month":"Jan"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| month | String | No | Month,last month by default. Valid value is Jan, Feb, Mar, Apr,May, Jun, Jul,Aug, Sep,Oct,Nov,Dec |

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. 1597026383085 |

### GET monthly statement (last year)

Retrieve monthly statement in the past year.

#### Rate Limit: 10 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/asset/monthly-statement

Request Example

GET /api/v5/asset/monthly-statement?month=Jan

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| month | String | Yes | Month, valid value is Jan, Feb, Mar, Apr,May, Jun, Jul,Aug, Sep,Oct,Nov,Dec |

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. 1597026383085 |
| state | String | Download link status  "finished" "ongoing" |

### Get convert currencies

#### Rate Limit: 6 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/asset/convert/currencies

Request Example

GET /api/v5/asset/convert/currencies

#### 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: UserID

#### HTTP Request

GET /api/v5/asset/convert/currency-pair

Request Example

GET /api/v5/asset/convert/currency-pair?fromCcy=USDT&toCcy=BTC

#### Response parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| fromCcy | String | Yes | Currency to convert from, e.g. USDT |
| toCcy | String | Yes | Currency to convert to, e.g. BTC |

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. BTC-USDT |
| baseCcy | String | Base currency, e.g. BTC in BTC-USDT |
| baseCcyMax | String | Maximum amount of base currency |
| baseCcyMin | String | Minimum amount of base currency |
| quoteCcy | String | Quote currency, e.g. USDT in BTC-USDT |
| 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: UserID

#### Rate Limit: 1 request per 5 seconds

#### Rate limit rule: Instrument

#### HTTP Request

POST /api/v5/asset/convert/estimate-quote

Request Example

POST /api/v5/asset/convert/estimate-quote

body

{

"baseCcy": "ETH",

"quoteCcy": "USDT",

"side": "buy",

"rfqSz": "30",

"rfqSzCcy": "USDT"

}

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| baseCcy | String | Yes | Base currency, e.g. BTC in BTC-USDT |
| quoteCcy | String | Yes | Quote currency, e.g. USDT in BTC-USDT |
| side | String | Yes | Trade side based on baseCcy buy sell |
| 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. 1597026383085 |
| ttlMs | String | Validity period of quotation in milliseconds |
| clQReqId | String | Client Order ID as assigned by the client |
| quoteId | String | Quote ID |
| baseCcy | String | Base currency, e.g. BTC in BTC-USDT |
| quoteCcy | String | Quote currency, e.g. USDT in BTC-USDT |
| side | String | Trade side based on baseCcy |
| 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](https://www.okx.com/docs-v5/en/#funding-account-rest-api-estimate-quote) before convert trade.

Only assets in the trading account supported convert.

#### Rate Limit: 10 requests per second

#### Rate limit rule: UserID

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

POST /api/v5/asset/convert/trade

body

{

"baseCcy": "ETH",

"quoteCcy": "USDT",

"side": "buy",

"sz": "30",

"szCcy": "USDT",

"quoteId": "quoterETH-USDT16461885104612381"

}

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| quoteId | String | Yes | Quote ID |
| baseCcy | String | Yes | Base currency, e.g. BTC in BTC-USDT |
| quoteCcy | String | Yes | Quote currency, e.g. USDT in BTC-USDT |
| side | String | Yes | Trade side based on baseCcy buy sell |
| 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** |
| --- | --- | --- |
| tradeId | String | Trade ID |
| quoteId | String | Quote ID |
| clTReqId | String | Client Order ID as assigned by the client |
| state | String | Trade state fullyFilled: success rejected: failed |
| instId | String | Currency pair, e.g. BTC-USDT |
| baseCcy | String | Base currency, e.g. BTC in BTC-USDT |
| quoteCcy | String | Quote currency, e.g. USDT in BTC-USDT |
| side | String | Trade side based on baseCcy buy sell |
| 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. 1597026383085 |

### Get convert history

#### Rate Limit: 6 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/asset/convert/history

Request Example

GET /api/v5/asset/convert/history

#### 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 ts, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| before | String | No | Pagination of data to return records newer than the requested ts, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| limit | String | No | Number of results per request. The maximum is 100. The default is 100. |
| tag | String | No | Order tag Applicable to broker user If the convert trading used tag, 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 |
| state | String | Trade state fullyFilled : success  rejected : failed |
| instId | String | Currency pair, e.g. BTC-USDT |
| baseCcy | String | Base currency, e.g. BTC in BTC-USDT |
| quoteCcy | String | Quote currency, e.g. USDT in BTC-USDT |
| side | String | Trade side based on baseCcy buy sell |
| 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. 1597026383085 |

### Get deposit payment methods

To display all the available fiat deposit payment methods

#### Rate Limit: 3 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/fiat/deposit-payment-methods

Request Example

GET /api/v5/fiat/deposit-payment-methods?ccy=TRY

body

{

"ccy" : "TRY",

}

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | Yes | ISO-4217 3 digit currency 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": "VakıfBank",

"bankCode": "TVBATR2AXXX",

"state": "active"

},

{

"paymentAcctId": "2",

"acctNum": "TR740001592093703829602622",

"recipientName": "John Doe",

"bankName": "FBHLTRISXXX",

"bankCode": "",

"state": "active"

}

]

}

]

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ccy | String | The currency code |
| paymentMethod | String | The payment method associated with the currency SEPA TR\_BANKS PIX |
| feeRate | String | The fee rate for each deposit, expressed as a percentage e.g. 0.02 represents 2 percent fee for each transaction. |
| 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, "" if not configured |
| accounts | Array of Object | An array containing information about payment accounts associated with the currency and method. |
| > paymentAccId | 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 | The state of the account, e.g. active |

### Get withdrawal payment methods

To display all the available fiat withdrawal payment methods

#### Rate Limit: 3 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/fiat/withdrawal-payment-methods

Request Example

GET /api/v5/fiat/withdrawal-payment-methods?ccy=TRY

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | Yes | ISO-4217 3 digit currency 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": "VakıfBank",

"bankCode": "TVBATR2AXXX",

"state": "active"

},

{

"paymentAcctId": "2",

"acctNum": "TR740001592093703829603024",

"recipientName": "John Doe",

"bankName": "Şekerbank",

"bankCode": "SEKETR2AXXX",

"state": "active"

}

]

}

]

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ccy | String | The currency code |
| paymentMethod | String | The payment method associated with the currency SEPA TR\_BANKS PIX |
| feeRate | String | The fee rate for each deposit, expressed as a percentage  e.g. 0.02 represents 2 percent fee for each transaction. |
| 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, "" 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 | The state of the account, e.g. active |

### Create Withdrawal Order

Initiate a fiat withdrawal request (Authenticated endpoint, Only for API keys with "Withdrawal" access)

#### Rate Limit: 3 requests per second

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/fiat/create-withdrawal

Request Example

POST /api/v5/fiat/create-withdrawal

body

{

"paymentAcctId":"412323",

"ccy" : "TRY",

"amt" : "10000",

"paymentMethod" : "TR\_BANKS",

"clientId" : "194a6975e98246538faeb0fab0d502df"

}

#### 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  TR\_BANKS  SEPA  PIX |
| clientId | String | Yes | Client-supplied ID, A combination of case-sensitive alphanumerics, all numbers, or all letters of up to 32 characters  eg. 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** |
| --- | --- | --- |
| amt | String | The requested amount for the transaction |
| cTime | String | The creation time of the transaction |
| clientId | String | The client ID associated with the transaction |
| ccy | String | The currency of the transaction |
| fee | String | The transaction fee |
| ordId | String | The unique order Id |
| paymentAcctId | String | The Id of the payment account used |
| paymentMethod | String | Payment Method  SEPA  TR\_BANKS  PIX |
| state | String | The State of the transaction |
| uTime | String | The update time of the transaction |

### Cancel Withdrawal Order

Cancel a pending fiat withdrawal order, currently only applicable to TRY

#### Rate Limit: 3 requests per second

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/fiat/cancel-withdrawal

Request Example

POST /api/v5/fiat/cancel-withdrawal

body

{

"ordId":"124041201450544699"

}

#### 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 |
| state | String | The state of the transaction eg. canceled |

### Get Withdrawal Order History

Get fiat withdrawal order history

#### Rate Limit: 3 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/fiat/withdrawal-order-history

Request Example

GET /api/v5/fiat/withdrawal-order-history

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | No | ISO-4217 3 digit currency code |
| paymentMethod | String | No | Payment Method  SEPA  TR\_BANKS  PIX |
| state | String | No | State of the order:  completed failed pending canceled inqueue processing |
| after | String | No | Filter with a begin timestamp. Unix timestamp format in milliseconds (inclusive) |
| before | String | No | Filter with an end timestamp. Unix timestamp format in milliseconds (inclusive) |
| limit | String | No | Number of results per request. Maximum and default is 100 |

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": "124041201450544690",

"paymentMethod": "TR\_BANKS",

"paymentAcctId": "20",

"amt": "5000",

"fee": "0",

"ccy": "TRY",

"state": "completed",

"clientId": "164a6975e48946538faeb0fab0d414fg"

}

]

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| amt | String | Final amount of the transaction |
| cTime | String | Creation time of the transaction |
| clientId | String | Client Id of the transaction |
| ccy | String | Currency of the transaction |
| fee | String | Transaction fee |
| ordId | String | Unique Order Id |
| paymentAcctId | String | ID of the payment account used |
| paymentMethod | String | Payment method type |
| state | String | State of the transaction eg. completed, failed, pending, canceled, inqueue |
| uTime | String | Update time of the transaction |

### Get Withdrawal Order Detail

Get fiat withdraw order detail

#### Rate Limit: 3 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/fiat/withdrawal

Request Example

GET /api/v5/fiat/withdrawal?ordId=024041201450544699

body

{

"ordId":"024041201450544699"

}

#### 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** |
| --- | --- | --- |
| amt | String | Amount of the transaction |
| ccy | String | The currency of the transaction |
| cTime | String | The creation time of the transaction |
| clientId | String | The original request ID associated with the transaction |
| fee | String | The transaction fee |
| ordId | String | The unique orderId |
| paymentAcctId | String | The ID of the payment account used |
| paymentMethod | String | The payment method type |
| state | String | The state of the transaction |
| uTime | String | The update time of the transaction |

### Get Deposit Order History

Get fiat deposit order history

#### Rate Limit: 3 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/fiat/deposit-order-history

Request Example

GET /api/v5/fiat/deposit-order-history

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | No | ISO-4217 3 digit currency code |
| paymentMethod | String | No | Payment Method  SEPA  TR\_BANKS  PIX |
| state | String | No | State of the order:  completed failed pending canceled inqueue processing |
| after | String | No | Filter with a begin timestamp. Unix timestamp format in milliseconds (inclusive) |
| before | String | No | Filter with an end timestamp. Unix timestamp format in milliseconds (inclusive) |
| 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** |
| --- | --- | --- |
| amt | String | Final amount of the transaction |
| cTime | String | Creation time of the transaction |
| clientId | String | Client Id of the transaction |
| ccy | String | Currency of the transaction |
| fee | String | Transaction fee |
| ordId | String | Unique Order Id |
| paymentAcctId | String | ID of the payment account used |
| paymentMethod | String | Payment method type |
| paymentMethodId | String | Payment method ID |
| state | String | State of the transaction completed failed pending canceled inqueue |
| uTime | String | Update time of the transaction |

### Get Deposit Order Detail

Get fiat deposit order detail

#### Rate Limit: 3 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/fiat/deposit

Request Example

GET /api/v5/fiat/deposit?ordId=024041201450544699

body

{

"ordId":"024041201450544699",

}

#### 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** |
| --- | --- | --- |
| amt | String | Amount of the transaction |
| ccy | String | The currency of the transaction |
| cTime | String | The creation time of the transaction |
| clientId | String | The original request ID associated with the transaction. If it's a deposit, it's most likely an empty string (""). |
| fee | String | The transaction fee |
| ordId | String | The unique orderId |
| paymentAcctId | String | The ID of the payment account used |
| paymentMethod | String | The payment method type |
| state | String | The state of the transaction |
| uTime | String | The update time of the transaction |

## 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

{

"op": "subscribe",

"args": [

{

"channel": "deposit-info"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name deposit-info |
| > ccy | String | No | Currency, e.g. BTC |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "deposit-info"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"deposit-info\""}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Operation subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name deposit-info |
| > ccy | String | No | Currency, e.g. BTC |
| 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": "TEhFAqpuHa3LY\*\*\*\*\*8ByNoGnrmexeGMw",

"ts": "1674103661123",

"txId": "bc5376817\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*dbb0d729f6b",

"uid": "289320\*\*\*\*60975104"

}]

}

#### Push data parameters

| **Parameters** | **Types** | **Description** |
| --- | --- | --- |
| arg | Object | Successfully subscribed channel |
| > channel | String | Channel name |
| > uid | String | User Identifier |
| > ccy | String | Currency, e.g. BTC |
| data | Array | 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 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  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 |
| > 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

{

"op": "subscribe",

"args": [

{

"channel": "withdrawal-info"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | Operation subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name withdrawal-info |
| > ccy | String | No | Currency, e.g. BTC |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "withdrawal-info"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"withdrawal-info\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | Operation subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name withdrawal-info |
| > ccy | String | No | Currency, e.g. BTC |
| 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,

"pTime": "1674103268578",

"pmtId": "",

"state": "0",

"subAcct": "test",

"tag": "",

"to": "TN8CKTQMnpWfT\*\*\*\*\*\*8KipbJ24ErguhF",

"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. BTC |
| data | Array | 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, e.g. USDT-ERC20, USDT-TRC20 |
| > nonTradableAsset | String | Whether it is a non-tradable asset or not true: non-tradable asset, false: tradable asset |
| > amt | String | Withdrawal amount |
| > ts | String | Time the withdrawal request was submitted, Unix timestamp format in milliseconds, e.g. 1655251200000. |
| > from | String | Withdrawal account It can be email/phone/sub-account name |
| > areaCodeFrom | String | Area code for the phone number If from 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 to is a phone number, this parameter returns the area code for the phone number |
| > tag | String | Some currencies require a tag for withdrawals |
| > pmtId | 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. TONCOIN attached tag name is comment, the return will be {'comment':'123456'} |
| > txId | 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. USDT |
| > state | String | Status of withdrawal   Stage 1 : Pending withdrawal10: Waiting transfer 0: Waiting withdrawal 4/5/6/8/9/12: Waiting manual review 7: Approved   Stage 2 : Withdrawal in progress (Applicable to on-chain withdrawals, internal transfers do not have this stage)1: Broadcasting your transaction to chain 15: Pending transaction validation 16: Due to local laws and regulations, your withdrawal may take up to 24 hours to arrive -3: Canceling   Final stage-2: Canceled  -1: Failed 2: Success |
| > wdId | String | Withdrawal ID |
| > clientId | String | Client-supplied ID |

# Sub-account

The API endpoints of sub-account require authentication.

## REST API

### Get sub-account list

Applies to master accounts only

#### Rate limit：2 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

GET /api/v5/users/subaccount/list

Request sample

GET /api/v5/users/subaccount/list

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"

}

]

}

#### 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 string | 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 |

### Reset the API Key of a sub-account

Applies to master accounts only and master accounts API Key must be linked to IP addresses. Only API keys with Trade privilege can call this endpoint.

#### Rate limit：1 request per second

#### Rate limit rule: UserID

#### HTTP request

POST /api/v5/users/subaccount/modify-apikey

Request sample

POST /api/v5/users/subaccount/modify-apikey

body

{

"subAcct":"yongxu",

"apiKey":"49e1b84b-6dee-4894-80ee-ce9eb7ad614f",

"ip":"1.1.1.1"

}

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 read\_only: Read trade: 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 ip is set to "", then no IP addresses is linked to the APIKey. |

Returned results

{

"code": "0",

"msg": "",

"data": [{

"subAcct": "yongxu",

"label": "v5",

"apiKey": "arg13sdfgs",

"perm": "read,trade",

"ip": "1.1.1.1",

"ts": "1597026383085"

}]

}

#### Response parameters

| **Parameter name** | **Type** | **Description** |
| --- | --- | --- |
| subAcct | String | Sub-account name |
| apiKey | String | Sub-accountAPI public key |
| label | String | Sub-account API Key label |
| perm | String | Sub-account API Key permissions read\_only: Read trade: Trade |
| ip | String | Sub-account API Key IP addresses that linked with API Key |
| ts | String | Creation time |

### 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: UserID

#### HTTP request

GET /api/v5/account/subaccount/balances

Request sample

GET /api/v5/account/subaccount/balances?subAcct=test1

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

{

"code": "0",

"data": [

{

"adjEq": "10679688.0460531643092577",

"borrowFroz": "",

"details": [

{

"availBal": "",

"availEq": "9930359.9998",

"cashBal": "9930359.9998",

"ccy": "USDT",

"crossLiab": "0",

"disEq": "9439737.0772999514",

"eq": "9930359.9998",

"eqUsd": "9933041.196999946",

"smtSyncEq": "0",

"spotCopyTradingEq": "0",

"fixedBal": "0",

"frozenBal": "0",

"imr": "",

"interest": "0",

"isoEq": "0",

"isoLiab": "0",

"liab": "0",

"maxLoan": "10000",

"mgnRatio": "",

"mmr": "",

"notionalLever": "",

"ordFrozen": "0",

"twap": "0",

"uTime": "1620722938250",

"upl": "0",

"uplLiab": "0",

"borrowFroz": "",

"spotIsoBal": "0",

"spotBal": "",

"openAvgPx": "",

"accAvgPx": "",

"spotUpl": "",

"spotUplRatio": "",

"totalPnl": "",

"totalPnlRatio": ""

},

{

"availBal": "",

"availEq": "33.6799714158199414",

"cashBal": "33.2009985",

"ccy": "BTC",

"crossLiab": "0",

"disEq": "1239950.9687532129092577",

"eq": "33.771820625136023",

"eqUsd": "1239950.9687532129092577",

"smtSyncEq": "0",

"spotCopyTradingEq": "0",

"fixedBal": "0",

"frozenBal": "0.0918492093160816",

"imr": "",

"interest": "0",

"isoEq": "0",

"isoLiab": "0",

"liab": "0",

"maxLoan": "1453.92289531493594",

"mgnRatio": "",

"mmr": "",

"notionalLever": "",

"ordFrozen": "0",

"twap": "0",

"uTime": "1620722938250",

"upl": "0.570822125136023",

"uplLiab": "0",

"borrowFroz": "",

"spotIsoBal": "0",

"spotBal": "",

"openAvgPx": "",

"accAvgPx": "",

"spotUpl": "",

"spotUplRatio": "",

"totalPnl": "",

"totalPnlRatio": ""

}

],

"imr": "3372.2942371050594217",

"isoEq": "0",

"mgnRatio": "70375.35408747017",

"mmr": "134.8917694842024",

"notionalUsd": "33722.9423710505978888",

"ordFroz": "0",

"totalEq": "11172992.1657531589092577",

"uTime": "1623392334718",

"upl": "-7.571730042000012"

}

],

"msg": ""

}

#### Response parameters

| **Parameters** | **Types** | **Description** |
| --- | --- | --- |
| uTime | String | The latest time to get account information, millisecond format of Unix timestamp, e.g. 1597026383085 |
| totalEq | String | The total amount of equity in USD |
| isoEq | String | Isolated margin equity in USD Applicable to Spot and futures mode/Multi-currency margin/Portfolio margin |
| adjEq | String | Adjusted / Effective equity in USD  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 Multi-currency margin/Portfolio margin |
| ordFroz | String | Margin frozen for pending cross orders in USD  Applicable to Multi-currency margin |
| imr | String | Initial margin requirement in USD  The sum of initial margins of all open positions and pending orders under cross-margin mode in USD.  Applicable to Multi-currency margin/Portfolio margin |
| mmr | String | Maintenance margin requirement in USD  The sum of maintenance margins of all open positions and pending orders under cross-margin mode in USD.  Applicable to Multi-currency margin/Portfolio margin |
| borrowFroz | String | Potential borrowing IMR of the account in USD  Only applicable to Multi-currency margin/Portfolio margin. It is "" for other margin modes. |
| mgnRatio | String | Margin ratio in USD.  Applicable to Multi-currency margin/Portfolio margin |
| notionalUsd | String | Notional value of positions in USD  Applicable to Multi-currency margin/Portfolio margin |
| upl | String | Cross-margin info of unrealized profit and loss at the account level in USD Applicable to Multi-currency margin/Portfolio margin |
| details | Array | 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. 1597026383085 |
| > isoEq | String | Isolated margin equity of currency Applicable to Spot and futures mode/Multi-currency margin/Portfolio margin |
| > fixedBal | String | Frozen balance for Dip Sniper and Peak Sniper |
| > availEq | String | Available equity of currency  The balance that can be used on margin or futures/swap trading.  Applicable to Spot and futures mode/Multi-currency margin/Portfolio margin |
| > disEq | String | Discount equity of currency in USD |
| > availBal | String | Available balance of currency |
| > frozenBal | String | Frozen balance of currency |
| > ordFrozen | String | Margin frozen for open orders Applicable to Spot mode/Spot and futures mode/Multi-currency margin |
| > liab | String | Liabilities of currency Applicable to Multi-currency margin/Portfolio margin |
| > upl | String | The sum of the unrealized profit & loss of all margin and derivatives positions of currency.  Applicable to Spot and futures mode/Multi-currency margin/Portfolio margin |
| > uplLiab | String | Liabilities due to Unrealized loss of currency Applicable to Multi-currency margin/Portfolio margin |
| > crossLiab | String | Cross Liabilities of currency Applicable to Multi-currency margin/Portfolio margin |
| > isoLiab | String | Isolated Liabilities of currency Applicable to Multi-currency margin/Portfolio margin |
| > mgnRatio | String | Cross margin ratio of currency  The index for measuring the risk of a certain asset in the account.  Applicable to Spot and futures mode and when there is cross position |
| > imr | String | Cross initial margin requirement at the currency level Applicable to Spot and futures mode and when there is cross position |
| > mmr | String | Cross maintenance margin requirement at the currency level Applicable to Spot and futures mode and when there is cross position |
| > interest | String | Interest of currency Applicable to Multi-currency margin/Portfolio margin |
| > twap | String | System's forced repayment(TWAP) indicator Divided into multiple levels from 0 to 5, the larger the number, the more likely the auto repayment will be triggered. Applicable to Multi-currency margin/Portfolio margin |
| > maxLoan | String | Max loan of currency Applicable to cross of Multi-currency margin/Portfolio margin |
| > eqUsd | String | Equity USD of currency |
| > borrowFroz | String | Potential borrowing IMR of currency in USD  Only applicable to Multi-currency margin/Portfolio margin. It is "" for other margin modes. |
| > notionalLever | String | Leverage of currency Applicable to Spot and futures mode |
| > spotIsoBal | String | SPOT isolated balance. only applicable to copy trading |
| > 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. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |
| > openAvgPx | Array | Spot average cost price. The unit is USD. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |
| > accAvgPx | Array | Spot accumulated cost price. The unit is USD. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |
| > spotUpl | String | Spot unrealized profit and loss. The unit is USD. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |
| > spotUplRatio | String | Spot unrealized profit and loss ratio. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |
| > totalPnl | String | Spot accumulated profit and loss. The unit is USD. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |
| > totalPnlRatio | String | Spot accumulated profit and loss ratio. [Clicking knows more](https://www.okx.com/help/i-introduction-of-spot) |

"" 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: UserID

#### HTTP request

GET /api/v5/asset/subaccount/balances

Request sample

GET /api/v5/asset/subaccount/balances?subAcct=test1

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. BTC or BTC,ETH. |

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: UserID

#### HTTP request

GET /api/v5/account/subaccount/max-withdrawal

Request Example

GET /api/v5/account/subaccount/max-withdrawal?subAcct=test1

#### 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. BTC or BTC,ETH. |

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 Multi-currency margin) |
| maxWdEx | String | Max withdrawal (including borrowed assets under Multi-currency margin) |
| spotOffsetMaxWd | String | Max withdrawal under Spot-Derivatives risk offset mode (excluding borrowed assets under Portfolio margin)  Applicable to Portfolio margin |
| spotOffsetMaxWdEx | String | Max withdrawal under Spot-Derivatives risk offset mode (including borrowed assets under Portfolio margin)  Applicable to Portfolio margin |

### Get history of sub-account transfer

Applies to master accounts only.

#### Rate limit：6 requests per second

#### Rate limit rule: UserID

#### HTTP request

GET /api/v5/asset/subaccount/bills

Request sample

GET /api/v5/asset/subaccount/bills

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 0: Transfers from master account to sub-account 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. 1597026383085 |
| 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. 1597026383085 |
| 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. 1597026383085 |

### 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: UserID

#### HTTP request

GET /api/v5/asset/subaccount/managed-subaccount-bills

Request sample

GET /api/v5/asset/subaccount/managed-subaccount-bills

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | No | Currency, e.g BTC |
| type | String | No | Transfer type 0: Transfers from master account to sub-account 1: 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. 1597026383085 |
| before | String | No | Query the data after the requested bill ID creation time (exclude), 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": [{

"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: UserID

#### HTTP request

POST /api/v5/asset/subaccount/transfer

Request sample

POST /api/v5/asset/subaccount/transfer

body

{

"ccy":"USDT",

"amt":"1.5",

"from":"6",

"to":"6",

"fromSubAccount":"test-1",

"toSubAccount":"test-2"

}

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 Multi-currency margin/Portfolio margin The default is false |
| omitPosRisk | String | No | Ignore position risk Default is false Applicable to Portfolio margin |

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: UserID

#### HTTP Request

POST /api/v5/users/subaccount/set-transfer-out

Request Example

POST /api/v5/users/subaccount/set-transfer-out

body

{

"subAcct": "Test001,Test002",

"canTransOut": true

}

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 true. false: cannot transfer out  true: 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.  false: cannot transfer out  true: 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: UserID

#### HTTP request

GET /api/v5/users/entrust-subaccount-list

Request sample

GET /api/v5/users/entrust-subaccount-list

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](https://www.okx.com/earn/onchain-earn)

### GET / Offers

#### Rate Limit: 3 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/finance/staking-defi/offers

Request Example

GET /api/v5/finance/staking-defi/offers

#### 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 |

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. BTC |
| productId | String | Product ID |
| 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 0 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 | Current target currency information available for investment |
| > ccy | String | Investment currency, e.g. BTC |
| > bal | String | Available balance to invest |
| > minAmt | String | Minimum subscription amount |
| > maxAmt | String | Maximum available subscription amount |
| earningData | Array of object | Earning data |
| > ccy | String | Earning currency, e.g. BTC |
| > earningType | String | Earning type 0: Estimated earning 1: Cumulative earning |
| state | String | Product state purchasable: Purchasable sold\_out: Sold out Stop: Suspension of subscription |
| redeemPeriod | Array of string | Redemption Period, format in [min time,max time] H: Hour, D: Day e.g. ["1H","24H"] represents redemption period is between 1 Hour and 24 Hours. ["14D","14D"] represents redemption period is 14 days. |
| fastRedemptionDailyLimit | String | Fast redemption daily limit If fast redemption is not supported, it will return ''. |

### POST / Purchase

#### Rate Limit: 2 requests per second

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/finance/staking-defi/purchase

Request Example

# Invest 100ZIL 30-day staking protocol

POST /api/v5/finance/staking-defi/purchase

body

{

"productId":"1234",

"investData":[

{

"ccy":"ZIL",

"amt":"100"

}

],

"term":"30"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| productId | String | Yes | Product ID |
| investData | Array | Yes | Investment data |
| > ccy | String | Yes | Investment currency, e.g. BTC |
| > 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: UserID

#### HTTP Request

POST /api/v5/finance/staking-defi/redeem

Request Example

# Early redemption of investment

POST /api/v5/finance/staking-defi/redeem

body

{

"ordId":"754147",

"protocolType":"defi",

"allowEarlyRedeem":true

}

#### 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: UserID

#### HTTP Request

POST /api/v5/finance/staking-defi/cancel

Request Example

POST /api/v5/finance/staking-defi/cancel

body

{

"ordId":"754147",

"protocolType":"defi"

}

#### 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: UserID

#### HTTP Request

GET /api/v5/finance/staking-defi/orders-active

Request Example

GET /api/v5/finance/staking-defi/orders-active

#### 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",

"msg": "",

"data": [

{

"ordId":"123456",

"state":"1",

"ccy": "GLMR",

"protocol": "glimmar", //Staking

"protocolType":"staking",

"term":"15",

"apy":"0.5496",

"investData":[

{

"ccy":"GLMR",

"amt":"100"

}

],

"earningData": [

{

"ccy": "GLMR",

"earningType":"1", // Daily distribution

"earnings":"3"

}

],

"purchasedTime":"1597026383085",

"tag": "",

"estSettlementTime": "",

"cancelRedemptionDeadline": "",

"fastRedemptionData": []

},

{

"ordId":"123457",

"state":"1",

"ccy": "USDT",

"protocol": "compond", //DEFI-loan

"protocolType":"defi",

"term":"0",

"apy":"0.12",

"investData":[

{

"ccy":"USDT",

"amt":"20",

"minAmt":"1",

"maxAmt":""

}

],

"earningData": [

{

"ccy": "USDT",

"earningType":"0", //Redeem distribution

"earnings":"3" //Estimated earning

},

{

"ccy": "COMP",

"earningType":"1", //Daily distribution

"earnings":"3" //Cumulative earning

}

],

"purchasedTime":"1597026383085",

"tag": "",

"estSettlementTime": "",

"cancelRedemptionDeadline": "",

"fastRedemptionData": []

},

{

"ordId":"123458",

"state":"1",

"ccy": "ETH",

"protocol": "sushiswap", //DEFI

"protocolType":"defi",

"term":"0",

"apy":"0.12",

"investData":[

{

"ccy":"USDT",

"amt":"100"

},

{

"ccy":"ETH",

"amt":"0.03"

}

],

"earningData": [

{

"ccy": "SUSHI",

"earningType":"1", // Daily distribution

"earnings":"3" // Cumulative earning

}

],

"purchasedTime":"1597026383085",

"tag": "",

"estSettlementTime": "",

"cancelRedemptionDeadline": "",

"fastRedemptionData": []

},

{

"ordId":"123458",

"state":"3",

"ccy": "LON",

"protocol": "tokenlon", //DEFI-pos

"protocolType":"defi",

"earningCcy": ["LON"],

"term":"7",

"apy":"0.12",

"investData":[

{

"ccy":"LON",

"amt":"1"

}

],

"earningData": [

{

"ccy": "LON",

"earningType":"0", //Redeem distribution

"earnings":"3" //Cumulative earning

}

],

"purchasedTime":"1597026383085",

"tag": "",

"estSettlementTime": "",

"cancelRedemptionDeadline": "",

"fastRedemptionData": []

}

]

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ccy | String | Currency, e.g. BTC |
| ordId | String | Order ID |
| productId | String | Product ID |
| state | String | Order state 8: Pending  13: Cancelling  9: Onchain  1: Earning  2: Redeeming |
| 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 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 object | Investment data |
| > ccy | String | Investment currency, e.g. BTC |
| > amt | String | Invested amount |
| earningData | Array of object | 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 object | 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: UserID

#### HTTP Request

GET /api/v5/finance/staking-defi/orders-history

Request Example

GET /api/v5/finance/staking-defi/orders-history

#### 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 |
| after | String | No | Pagination of data to return records earlier than the requested ID. The value passed is the corresponding ordId |
| before | String | No | Pagination of data to return records newer than the requested ID. The value passed is the corresponding ordId |
| limit | String | No | Number of results per request. The default is 100. The maximum is 100. |

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. BTC |
| ordId | String | Order ID |
| productId | String | Product ID |
| state | String | Order state 3: Completed (including canceled and redeemed) |
| 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 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 | Investment data |
| > ccy | String | Investment currency, e.g. BTC |
| > amt | String | Invested amount |
| earningData | Array | Earning data |
| > ccy | String | Earning currency, e.g. BTC |
| > earningType | String | Earning type 0: Estimated earning 1: Cumulative earning |
| > realizedEarnings | String | Cumulative earning of redeemed orders This field is only valid when the order is in redemption state |
| purchasedTime | String | Order purchased time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| redeemedTime | String | Order redeemed time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| 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](https://www.okx.com/earn/eth)

### GET / Product info

#### Rate Limit: 3 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/finance/staking-defi/eth/product-info

Request Example

GET /api/v5/finance/staking-defi/eth/product-info

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: UserID

#### HTTP Request

POST /api/v5/finance/staking-defi/eth/purchase

Request Example

POST /api/v5/finance/staking-defi/eth/purchase

body

{

"amt":"100"

}

#### 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: UserID

#### HTTP Request

POST /api/v5/finance/staking-defi/eth/redeem

Request Example

POST /api/v5/finance/staking-defi/eth/redeem

body

{

"amt": "10"

}

#### 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.

### 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: UserID

#### HTTP Request

GET /api/v5/finance/staking-defi/eth/balance

Request Example

GET /api/v5/finance/staking-defi/eth/balance

#### 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. BETH |
| 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. 1597026383085 |

### GET / Purchase&Redeem history

#### Rate Limit: 6 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/finance/staking-defi/eth/purchase-redeem-history

Request Example

GET /api/v5/finance/staking-defi/eth/purchase-redeem-history?type=purchase

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| type | String | Yes | Type purchase redeem |
| status | String | No | Status pending success failed |
| after | String | No | Pagination of data to return records earlier than the requestTime. The value passed is the corresponding timestamp |
| before | String | No | Pagination of data to return records newer than the requestTime. The value passed is the corresponding timestamp |
| limit | String | No | Number of results per request. The default is 100. The maximum is 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 purchase redeem |
| amt | String | Purchase/Redeem amount |
| redeemingAmt | String | Redeeming amount |
| status | String | Status pending success failed |
| 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/eth/apy-history

Request Example

GET /api/v5/finance/staking-defi/eth/apy-history?days=2

#### 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": "3.74000000",

"ts": "1699228800000"

},

{

"rate": "3.61000000",

"ts": "1699142400000"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| rate | String | APY(Annual percentage yield), e.g. 0.01 represents 1% |
| ts | String | Data time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

## Simple earn flexible

Simple earn flexible (saving) is earned by lending to leveraged trading users in the lending market. [learn more](https://www.okx.com/earn/simple-earn)

### GET / Saving balance

#### Rate Limit: 6 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/finance/savings/balance

Request Example

GET /api/v5/finance/savings/balance?ccy=USDT

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 saving balance

result = fundingAPI.get\_saving\_balance(

ccy="USDC"

)

print(result)

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | No | Currency, e.g. BTC |

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 | Lending rate |
| loanAmt | String | Lending amount |
| pendingAmt | String | Pending amount |
| redemptAmt | String | (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: UserID

#### HTTP Request

POST /api/v5/finance/savings/purchase-redempt

Request Example

POST /api/v5/finance/savings/purchase-redempt

body

{

"ccy":"BTC",

"amt":"1",

"side":"purchase",

"rate":"0.01"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | Yes | Currency, e.g. BTC |
| amt | String | Yes | Purchase/redemption amount |
| side | String | Yes | Action type.  purchase: purchase saving shares  redempt: redeem saving shares |
| rate | String | Yes | Annual purchase rate 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 |

### POST / Set lending rate

#### Rate Limit: 6 requests per second

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/finance/savings/set-lending-rate

Request Example

POST /api/v5/finance/savings/set-lending-rate

body

{

"ccy":"BTC",

"rate":"0.02"

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | Yes | Currency, e.g. BTC |
| 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. BTC |
| rate | String | Annual lending rate |

### GET / Lending history

Return data in the past month.

#### Rate Limit: 6 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/finance/savings/lending-history

Request Example

GET /api/v5/finance/savings/lending-history

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | No | Currency, e.g. BTC |
| after | String | No | Pagination of data to return records earlier than the requested ts, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| before | String | No | Pagination of data to return records newer than the requested ts, 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",

"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. BTC |
| 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. 1597026383085 |

### 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

GET /api/v5/finance/savings/lending-rate-summary

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | No | Currency, e.g. BTC |

Response Example

{

"code": "0",

"msg": "",

"data": [{

"ccy": "BTC",

"avgAmt": "10000",

"avgAmtUsd": "10000000000",

"avgRate": "0.03",

"preRate": "0.02",

"estRate": "0.01"

}]

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ccy | String | Currency, e.g. BTC |
| avgAmt | String | 24H average borrowing amount |
| avgAmtUsd | String | 24H average borrowing amount in USD value |
| avgRate | String | 24H average lending rate |
| preRate | String | Last annual interest rate |
| estRate | String | Next estimate annual 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

GET /api/v5/finance/savings/lending-rate-history

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | No | Currency, e.g. BTC |
| after | String | No | Pagination of data to return records earlier than the requested ts, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| before | String | No | Pagination of data to return records newer than the requested ts, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| limit | String | No | Number of results per request. The maximum is 100. The default is 100. If ccy is not specified, all data under the same ts will be returned, not limited by limit |

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. BTC |
| amt | String | Lending amount |
| rate | String | Lending annual interest rate |
| ts | String | Time, Unix timestamp format in milliseconds, e.g. 1597026383085 |

## Simple earn fixed

### GET / Lending offers (public)

Get lending-supported currencies and estimated APY.

#### Rate Limit: 3 requests per second

#### Rate limit rule: IP

#### HTTP Request

GET /api/v5/finance/fixed-loan/lending-offers

Request Example

GET /api/v5/finance/fixed-loan/lending-offers

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | No | Lending currency, e.g. BTC |
| term | String | No | Fixed term for lending order 30D: 30 days |

Response Example

{

"code": "0",

"data": [

{

"ccy": "BTC",

"lendQuota": "0.5",

"minLend": "0.02",

"rate": "0.0058",

"term": "30D"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ccy | String | Lending currency, e.g. BTC |
| term | String | Fixed term for lending order 30D: 30 days |
| rate | String | Latest lending APY, in decimal. e.g. 0.01 represent 1% |
| minLend | String | Minimum lending amount |
| lendQuota | String | Lending quota |

### GET / Lending APY history (public)

#### Rate Limit: 3 requests per second

#### Rate limit rule: IP

#### HTTP Request

GET /api/v5/finance/fixed-loan/lending-apy-history

Request Example

GET /api/v5/finance/fixed-loan/lending-apy-history?ccy=USDT&term=30D

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | Yes | Lending currency, e.g. BTC |
| term | String | Yes | Fixed term for lending order 30D: 30 days |

Response Example

{

"code": "0",

"data": [

{

"ccy": "USDT",

"rate": "0.0100",

"ts": "1712559600000"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ccy | String | Lending currency, e.g. BTC |
| rate | String | Lending APY, in decimal. e.g. 0.01 represent 1% |
| ts | String | Timestamp for lending, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### GET / Lending volume (public)

#### Rate Limit: 3 requests per second

#### Rate limit rule: IP

#### HTTP Request

GET /api/v5/finance/fixed-loan/pending-lending-volume

Request Example

GET /api/v5/finance/fixed-loan/pending-lending-volume?ccy=BTC&term=30D

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | Yes | Lending currency, e.g. BTC |
| term | String | Yes | Fixed term for lending order 30D: 30 days |

Response Example

{

"code": "0",

"data": [

{

"ccy": "USDT",

"pendingVol": "1000",

"rateRangeFrom": "0.001",

"rateRangeTo": "0.031",

"term": "30D"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ccy | String | Lending currency, e.g. BTC |
| term | String | Fixed term for lending order 30D: 30 days |
| rateRangeFrom | String | Lending APR of the lower range, e.g. 0.0100 represent 1% |
| rateRangeTo | String | Lending APR of the higher range, e.g. 0.0100 represent 1% |
| pendingVol | String | Lending volume pending to match |

### POST / Place lending order

#### Rate Limit: 2 requests per second

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/finance/fixed-loan/lending-order

Request Example

POST /api/v5/finance/fixed-loan/lending-order

body

{

"ccy": "USDT",

"amt": "1",

"rate": "0.01",

"term": "30D",

"autoRenewal": true

}

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | Yes | Lending currency, e.g. BTC |
| amt | String | Yes | Lending amount |
| rate | String | Yes | Lending APR, in decimal. e.g. 0.01 represents 1%. |
| term | String | Yes | Fixed term for Lending order |
| autoRenewal | Boolean | No | Whether or not auto-renewal when the term is due true: auto-renewal false: not auto-renewal Default is false. |

Response Example

{

"code": "0",

"data": [

{

"ordId":"2405162053378222"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ordId | String | Lending order ID |

### POST / Amend lending order

#### Rate Limit: 2 requests per second

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/finance/fixed-loan/amend-lending-order

Request Example

POST /api/v5/finance/fixed-loan/amend-lending-order

body

{

"ordId":"2405162053378222",

"changeAmt":"-100"

}

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ordId | String | Yes | Order ID |
| changeAmt | String | No | Redemption Amount, e.g. -0.1 represents redemption amount is 0.1. |
| rate | String | No | Lending APR, in decimal. e.g. 0.01 represents 1%. |
| autoRenewal | Boolean | No | Whether or not auto-renewal when the term is due true: auto-renewal false: not auto-renewal |

Response Example

{

"code": "0",

"data": [

{

"ordId":"2405162053378222"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ordId | String | Lending order ID |

### GET / Lending order list

#### Rate Limit: 3 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/finance/fixed-loan/lending-orders-list

Request Example

GET /api/v5/finance/fixed-loan/lending-orders-list

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ordId | String | No | Lending order ID |
| ccy | String | No | Lending currency, e.g. BTC |
| state | String | No | State pending earning expired settled |
| after | String | No | Pagination of data to return records earlier than the requested ordId |
| before | String | No | Pagination of data to return records newer than the requested ordId |
| limit | String | No | Number of results per request. The maximum is 100. The default is 100. |

Response Example

{

"code": "0",

"data": [

{

"amt": "10",

"autoRenewal": true,

"cTime": "1718955882000",

"ccy": "USDT",

"earningAmt": "0",

"ordId": "2406211544415051",

"pendingAmt": "10",

"rate": "0.01",

"settledTime": "",

"startTime": "",

"state": "pending",

"term": "30D",

"totalInterest": "0",

"uTime": "1718955881000"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ordId | String | Lending order ID |
| state | String | State |
| ccy | String | Lending currency, e.g. BTC |
| amt | String | Lending amount |
| rate | String | lending APR, in decimal. e.g. 0.01 represent 1% |
| term | String | Fixed term for lending order, e.g. 30D |
| autoRenewal | Boolean | Whether or not auto-renewal when the term is due true: auto-renewal false: not auto-renewal |
| totalInterest | String | Total interest |
| pendingAmt | String | Pending amount |
| earningAmt | String | Earning amount |
| startTime | String | Start earning time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| settledTime | String | Settled time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| cTime | String | Creation time for lending order, unix timestamp format in milliseconds, e.g. 1597026383085 |
| uTime | String | Update time for lending order, unix timestamp format in milliseconds, e.g. 1597026383085 |

### GET / Lending sub order list

#### Rate Limit: 2 requests per second

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/finance/fixed-loan/lending-sub-orders

Request Example

GET /api/v5/finance/fixed-loan/lending-sub-orders?ordId=2405231639344615

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ordId | String | Yes | Lending order ID |
| state | String | No | State earning expired settled |
| after | String | No | Pagination of data to return records earlier than the requested subOrdId |
| before | String | No | Pagination of data to return records newer than the requested subOrdId |
| limit | String | No | Number of results per request. The maximum is 100. The default is 100. |

Response Example

{

"code": "0",

"data": [

{

"accruedInterest": "",

"amt": "100",

"cTime": "1716453989000",

"ccy": "USDT",

"earlyTerminatedPenalty": "0.0209",

"expiryTime": "1719045989000",

"finalSettlementTime": "1721637989000",

"ordId": "2405231639344615",

"overdueInterest": "0",

"rate": "0.01",

"settledTime": "1716454032000",

"state": "settled",

"subOrdId": "2405231646292913",

"term": "30D",

"totalInterest": "0",

"uTime": "1716454032000"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| ordId | String | Lending order ID |
| subOrdId | String | Sub-order ID |
| state | String | Sub-order state |
| ccy | String | Sub-order currency, e.g. BTC |
| amt | String | Sub-order lending amount |
| rate | String | Sub-order lending APR, in decimal, e.g. 0.01 represent 1% |
| term | String | Fixed term for sub-order, e.g. 30D |
| expiryTime | String | Sub-order expiration time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| totalInterest | String | Sub-order total interest |
| accruedInterest | String | Sub-order accrued interest |
| earlyTerminatedPenalty | String | Sub-order early terminated penalty |
| overdueInterest | String | Sub-order overdue interest |
| finalSettlementTime | String | Sub-order final settlement time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| settledTime | String | Sub-order actual settlement time, Unix timestamp format in milliseconds, e.g. 1597026383085 |
| cTime | String | Creation time for sub-order, unix timestamp format in milliseconds, e.g. 1597026383085 |
| uTime | String | Update time for sub-order, unix timestamp format in milliseconds, e.g. 1597026383085 |

## 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](https://www.okx.com/loan)

### GET / Borrowable currencies

Get borrowable currencies

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/finance/flexible-loan/borrow-currencies

Request Example

GET /api/v5/finance/flexible-loan/borrow-currencies

Response Example

{

"code": "0",

"data": [

{

"borrowCcy": "USDT"

},

{

"borrowCcy": "USDC"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| borrowCcy | String | Borrowable currency, e.g. BTC |

### GET / Collateral assets

Get collateral assets in funding account.

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/finance/flexible-loan/collateral-assets

Request Example

GET /api/v5/finance/flexible-loan/collateral-assets

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | No | Collateral currency, e.g. BTC |

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 object | Collateral assets data |
| > ccy | String | Currency, e.g. BTC |
| > amt | String | Available amount |
| > notionalUsd | String | Notional value in USD |

### POST / Maximum loan amount

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/finance/flexible-loan/max-loan

Request Example

POST /api/v5/finance/flexible-loan/max-loan

{

"borrowCcy": "USDT"

}

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| borrowCcy | String | Yes | Currency to borrow, e.g. USDT |
| supCollateral | Array of object | No | Supplementary collateral assets |
| > ccy | String | Yes | Currency, e.g. BTC |
| > 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. USDT |
| maxLoan | String | Maximum available loan |
| notionalUsd | String | Maximum available loan notional value, unit in USD |
| remainingQuota | String | Remaining quota, unit in borrowCcy |

### POST / Adjust collateral

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

POST /api/v5/finance/flexible-loan/adjust-collateral

Request Example

POST /api/v5/finance/flexible-loan/max-loan

{

"type":"add",

"collateralCcy": "BTC",

"collateralAmt": "0.1"

}

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| type | String | Yes | Operation type add: Add collateral reduce: Reduce collateral |
| collateralCcy | String | Yes | Collateral currency, e.g. BTC |
| 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: UserID

#### HTTP Request

GET /api/v5/finance/flexible-loan/loan-info

Request Example

GET /api/v5/finance/flexible-loan/loan-info

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 USD |
| loanData | Array of object | Loan data |
| > ccy | String | Loan currency, e.g. USDT |
| > amt | String | Loan amount |
| collateralNotionalUsd | String | Collateral value in USD |
| collateralData | Array of object | Collateral data |
| > ccy | String | Collateral currency, e.g. BTC |
| > amt | String | Collateral amount |
| riskWarningData | Object | Risk warning data |
| > instId | String | Liquidation instrument ID, e.g. BTC-USDT This field is only valid when there is only one type of collateral and one type of borrowed currency. In other cases, it returns "". |
| > liqPx | String | Liquidation price The unit of the liquidation price is the quote currency of the instrument, e.g. USDT in BTC-USDT. This field is only valid when there is only one type of collateral and one type of borrowed currency. In other cases, it returns "". |
| curLTV | String | Current LTV, e.g. 0.1 represents 10% Note: LTV = Loan to Value |
| marginCallLTV | String | Margin call LTV, e.g. 0.1 represents 10% 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. 0.1 represents 10% 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: UserID

#### HTTP Request

GET /api/v5/finance/flexible-loan/loan-history

Request Example

GET /api/v5/finance/flexible-loan/loan-history

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| type | String | No | Action type borrowed repaid collateral\_locked collateral\_released forced\_repayment\_buy forced\_repayment\_sell forced\_liquidation partial\_liquidation |
| after | String | No | Pagination of data to return records earlier than the requested refId(not include) |
| before | String | No | Pagination of data to return records newer than the requested refId(not include) |
| limit | String | No | Number of results per request. The maximum is 100. The default is 100. |

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. BTC |
| amt | String | Amount |
| ts | String | Timestamp for the action, Unix timestamp format in milliseconds, e.g. 1597026383085 |

### GET / Accrued interest

#### Rate Limit: 5 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP Request

GET /api/v5/finance/flexible-loan/interest-accrued

Request Example

GET /api/v5/finance/flexible-loan/interest-accrued

#### Request Parameters

| **Parameters** | **Types** | **Required** | **Description** |
| --- | --- | --- | --- |
| ccy | String | No | Loan currency, e.g. BTC |
| after | String | No | Pagination of data to return records earlier than the requested refId(not include) |
| before | String | No | Pagination of data to return records newer than the requested refId(not include) |
| limit | String | No | Number of results per request. The maximum is 100. The default is 100. |

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. BTC |
| loan | String | Loan when calculated interest |
| interest | String | Interest |
| interestRate | String | Hourly APY, e.g. 0.01 represents 1% |
| ts | String | Timestamp to calculated interest, Unix timestamp format in milliseconds, e.g. 1597026383085 |

# 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: UserID

#### HTTP request

GET /api/v5/affiliate/invitee/detail

Request sample

GET /api/v5/affiliate/invitee/detail?uid=11111111

#### 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** |
| --- | --- | --- |
| inviteeLv | 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 |

### Get the user's affiliate rebate information

This endpoint will be offline soon, please use [Get the invitee's detail](https://www.okx.com/docs-v5/en/#affiliate-rest-api-get-the-invitee-39-s-detail)

It is used to get the user's affiliate rebate information for affiliate.

#### Rate limit：20 requests per 2 seconds

#### Rate limit rule: UserID

#### HTTP request

GET /api/v5/users/partner/if-rebate

Request sample

GET /api/v5/users/partner/if-rebate?apiKey=86b02e93-67ab-497d-9970-8cce00a028c3

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| apiKey | String | Yes | The user's API key. Only applicable to the API key of invitee master account |

Returned results

{

"code": "0",

"msg": "",

"data": {

"result": true,

"type": "0"

}

}

#### Response parameters

| **Parameter name** | **Type** | **Description** |
| --- | --- | --- |
| result | Boolean | Whether the user is invited by the current affiliate. true, false |
| type | String | Whether there is affiliate rebate. 0 There is affiliate rebate 1 There is no affiliate rebate. Because the account which is requesting this endpoint is not affiliate  2 There is no affiliate rebate. Because there is no relationship of invitation or recall, e.g. api key does not exist  4 There is no affiliate rebate. Because the user level is equal to or more than VIP6 |

# 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.

#### Rate Limit: 1 request per 5 seconds

#### HTTP Requests

GET /api/v5/system/status

Request Example

GET /api/v5/system/status

GET /api/v5/system/status?state=canceled

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 | System maintenance status scheduled: waiting ongoing: processing pre\_open: pre\_open completed: completed canceled: canceled Generally, pre\_open last about 10 minutes. There will be pre\_open when the time of upgrade is too long.  If this parameter is not filled, the data with status scheduled, ongoing and pre\_open will be returned by default |

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. 1617788463867 |
| end | String | Time of resuming trading totally. Unix timestamp format in milliseconds, e.g. 1617788463867. It is expected end time before completed, changed to actual end time after completed. |
| preOpenBegin | String | The time of pre\_open. Canceling orders, placing Post Only orders, and transferring funds to trading accounts are back after preOpenBegin. |
| 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 0: WebSocket 5: Trading service 6: Block trading 7: Trading bot 8: Trading service (in batches of accounts) 9: Trading service (in batches of products) 10: Spread trading 11: Copy trading 99: Others (e.g. Suspend partial instruments) |
| system | String | System unified: Trading account |
| scheDesc | String | Rescheduled description, e.g. Rescheduled from 2021-01-26T16:30:00.000Z to 2021-01-28T16:30:00.000Z |
| maintType | String | Maintenance type 1: Scheduled maintenance 2: Unscheduled maintenance 3: System disruption |
| env | String | Environment 1: Production Trading 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

{

"op": "subscribe",

"args": [

{

"channel": "status"

}

]

}

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| op | String | Yes | subscribe unsubscribe |
| args | Array | Yes | List of subscribed channels |
| > channel | String | Yes | Channel name status |

Successful Response Example

{

"event": "subscribe",

"arg": {

"channel": "status"

},

"connId": "a4d3ae55"

}

Failure Response Example

{

"event": "error",

"code": "60012",

"msg": "Invalid request: {\"op\": \"subscribe\", \"argss\":[{ \"channel\" : \"statuss\"}]}",

"connId": "a4d3ae55"

}

#### Response parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| event | String | Yes | subscribe unsubscribe error |
| arg | Object | No | Subscribed channel |
| > channel | String | Yes | Channel name status |
| 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 | Subscribed data |
| > title | String | The title of system maintenance instructions |
| > state | String | System maintenance status,scheduled: waiting; ongoing: processing; pre\_open: pre\_open; completed: completed ;canceled: canceled.  Generally, pre\_open last about 10 minutes. There will be pre\_open when the time of upgrade is too long. |
| > begin | String | Start time of system maintenance, Unix timestamp format in milliseconds, e.g. 1617788463867 |
| > end | String | Time of resuming trading totally. Unix timestamp format in milliseconds, e.g. 1617788463867. It is expected end time before completed, changed to actual end time after completed. |
| > preOpenBegin | String | The time of pre\_open. Canceling orders, placing Post Only orders, and transferring funds to trading accounts are back after preOpenBegin. |
| > 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, 0: WebSocket ; 5: Trading service; 6: Block trading; 7: Trading bot; 8: Trading service (in batches of accounts); 9: Trading service (in batches of products); 10: Spread trading; 11: Copy trading; 99: Others (e.g. Suspend partial instruments) |
| > system | String | System, unified: Trading account |
| > scheDesc | String | Rescheduled description, e.g. Rescheduled from 2021-01-26T16:30:00.000Z to 2021-01-28T16:30:00.000Z |
| > maintType | String | Maintenance type 1: Scheduled maintenance; 2: Unscheduled maintenance; 3: System disruption |
| > env | String | Environment. 1: Production Trading, 2: Demo Trading |
| > ts | String | Push time due to change event, Unix timestamp format in milliseconds, e.g. 1617788463867 |

# Announcement

## GET / Announcements

Get announcements, the response is sorted by pTime 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: UserID(Private) or IP(Public)

#### HTTP Requests

GET /api/v5/support/announcements

Request Example

GET /api/v5/support/announcements

#### Request Parameters

| **Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| annType | String | No | Announcement type. Delivering the annType 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-latest-announcements",

"pTime": "1726128000000",

"title": "OKX to delist KISHU margin trading pairs",

"url": "https://www.okx.com/help/okx-to-delist-kishu-margin-trading-pairs"

},

{

"annType": "announcements-latest-announcements",

"pTime": "1725967800000",

"title": "OKX completed MATIC token migration",

"url": "https://www.okx.com/help/okx-completed-matic-token-migration"

}

],

"totalPage": "90"

}

],

"msg": ""

}

#### Response Parameters

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| totalPage | String | Total number of pages |
| details | String | List of announcements |
| > title | String | Announcement title |
| > annType | String | Announcement type |
| > pTime | String | Publish time. Unix timestamp format in milliseconds, e.g. 1597026383085 |
| > url | String | Announcement url |

## GET / Announcement types

Authentication is not required for this public endpoint.

Get announcements types

#### Rate Limit: 1 request per 2 seconds

#### Rate limit rule: IP

#### HTTP Requests

GET /api/v5/support/announcement-types

Request Example

GET /api/v5/support/announcement-types

#### Request Parameters

None

Response Example

{

"code": "0",

"data": [

{

"annType": "announcements-latest-announcements",

"annTypeDesc": "Latest announcements"

},

{

"annType": "announcements-latest-events",

"annTypeDesc": "Latest events"

}

],

"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. 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 is offline or unavailable. |
| 50006 | 400 | Invalid Content-Type. Please use "application/JSON". |
| 50007 | 200 | Account blocked. |
| 50008 | 200 | User does not exist. |
| 50009 | 200 | Account is suspended due to ongoing liquidation. |
| 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} cannot be empty. |
| 50015 | 400 | Either parameter {param0} or {param1} is required. |
| 50016 | 400 | Parameter {param0} and {param1} is an invalid pair. |
| 50017 | 200 | Position frozen and related operations restricted due to auto-deleveraging (ADL). Try again later |
| 50018 | 200 | {param0} frozen and related operations restricted due to auto-deleveraging (ADL). Try again later |
| 50019 | 200 | Account frozen and related operations restricted due to auto-deleveraging (ADL). Try again later |
| 50020 | 200 | Position frozen and related operations are restricted due to liquidation. Try again later |
| 50021 | 200 | {param0} frozen and related operations are restricted due to liquidation. Try again later |
| 50022 | 200 | Account frozen and related operations are restricted due to liquidation. Try again later |
| 50023 | 200 | Funding fees frozen and related operations are restricted. Try again later |
| 50024 | 200 | Either parameter {param0} or {param1} should be submitted. |
| 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 take the order, please reach out to support center 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. |
| 50044 | 200 | Must select one broker type |
| 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. |
| 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. |
| 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. |

#### 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. |
| 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 does not exist |
| 51002 | 200 | Instrument ID does not match underlying index |
| 51003 | 200 | Either client order ID or order ID is required |
| 51004 | 200 | Order failed. For isolated long/short mode of {instId}, the sum of current order size, position quantity in the same direction, and pending orders in the same direction cannot be more than {tierLimitQuantity}(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: {leverage}×, current order size: {size} contracts, position quantity in the same direction: {posNumber} contracts, pending orders in the same direction: {pendingNumber} contracts). |
| 51004 | 200 | Order failed. For cross long/short mode of {instId}, the sum of current order size, position quantity in the long and short directions, and pending orders in the long and short directions cannot be more than {tierLimitQuantity}(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: {leverage}×, current order size: {size} contracts, position quantity in the long and short directions: {posLongShortNumber} contracts, pending orders in the long and short directions: {pendingLongShortNumber} contracts). |
| 51004 | 200 | Order failed. For cross buy/sell mode of {businessType} and instFamily {instFamily}, 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 cannot be more than {tierLimitQuantity}(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: {leverage}×, current order size: {size} contracts, current instId position quantity in the long and short directions: {posLongShortNumber} contracts, current instId pending orders in the long and short directions: {pendingLongShortNumber} contracts, other contracts of the same instFamily: {otherQuote} contracts). |
| 51004 | 200 | Order failed. For buy/sell mode of {instId}, the sum of current buy order size, position quantity, and pending buy orders cannot be more than {tierLimitQuantity}(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: {leverage}×, current buy order size: {size} contracts, position quantity: {posNumber} contracts, pending buy orders: {pendingNumber} contracts). |
| 51004 | 200 | Order failed. For buy/sell mode of {instId}, the sum of current sell order size, position quantity, and pending sell orders cannot be more than {tierLimitQuantity}(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: {leverage}×, current sell order size: {size} contracts, position quantity: {posNumber} contracts, pending sell orders: {pendingNumber} contracts). |
| 51004 | 200 | Order failed. For cross buy/sell mode of {businessType} and instFamily {instFamily}, the sum of current buy order size, current instId position quantity, current instId pending buy orders, and other contracts of the same instFamily cannot be more than {tierLimitQuantity}(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: {leverage}×, current buy order size: {size} contracts, current instId position quantity: {posNumber} contracts, current instId pending buy orders: {pendingNumber} contracts, other contracts of the same instFamily: {otherQuote} contracts). |
| 51004 | 200 | Order failed. For cross buy/sell mode of {businessType} and instFamily {instFamily}, the sum of current sell order size, current instId position quantity, current instId pending sell orders, and other contracts of the same instFamily cannot be more than {tierLimitQuantity}(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: {leverage}×, current sell order size: {size} contracts, current instId position quantity: {posNumber} contracts, current instId pending sell orders: {pendingNumber} contracts, other contracts of the same instFamily: {otherQuote} contracts). |
| 51004 | 200 | Order amendment failed. For isolated long/short mode of {instId}, the sum of increment order size by amendment, position quantity in the same direction, and pending orders in the same direction cannot be more than {tierLimitQuantity}(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: {leverage}×, increment order size by amendment: {size} contracts, position quantity in the same direction: {posNumber} contracts, pending orders in the same direction: {pendingNumber} contracts). |
| 51004 | 200 | Order amendment failed. For cross long/short mode of {instId}, 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 cannot be more than {tierLimitQuantity}(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: {leverage}×, increment order size by amendment: {size} contracts, position quantity in the long and short directions: {posLongShortNumber} contracts, pending orders in the same direction: {pendingLongShortNumber} contracts). |
| 51004 | 200 | Order amendment failed. For cross buy/sell mode of {businessType} and instFamily {instFamily}, 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 cannot be more than {tierLimitQuantity}(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: {leverage}×, increment order size by amendment: {size} contracts, current instId position quantity in the long and short directions: {posLongShortNumber} contracts, current instId pending orders in the long and short directions: {pendingLongShortNumber} contracts, other contracts of the same instFamily: {otherQuote} contracts). |
| 51004 | 200 | Order amendment failed. For buy/sell mode of {instId}, the sum of increment order size by amending current buy order, position quantity, and pending buy orders cannot be more than {tierLimitQuantity}(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: {leverage}×, increment order size by amending current buy order: {size} contracts, position quantity: {posNumber} contracts, pending buy orders: {pendingNumber} contracts). |
| 51004 | 200 | Order amendment failed. For buy/sell mode of {instId}, the sum of increment order size by amending current sell order, position quantity, and pending sell orders cannot be more than {tierLimitQuantity}(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: {leverage}×, increment order size by amending current sell order: {size} contracts, position quantity: {posNumber} contracts, pending sell orders: {pendingNumber} contracts). |
| 51004 | 200 | Order amendment failed. For cross buy/sell mode of {businessType} and instFamily {instFamily}, 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 cannot be more than {tierLimitQuantity}(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: {leverage}×, increment order size by amending current buy order: {size} contracts, current instId position quantity: {posNumber} contracts, current instId pending buy orders: {pendingNumber} contracts, other contracts of the same instFamily: {otherQuote} contracts). |
| 51004 | 200 | Order amendment failed. For cross buy/sell mode of {businessType} and instFamily {instFamily}, 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 cannot be more than {tierLimitQuantity}(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: {leverage}×, increment order size by amending current sell order: {size} contracts, current instId position quantity: {posNumber} contracts, current instId pending sell orders: {pendingNumber} contracts, other contracts of the same instFamily: {otherQuote} 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 | Order failed. The requested borrow amount is larger than the available {param0} borrow amount of your position tier (Existing pending orders and the new order are required to borrow {param1}, Remaining limit {param2}, Limit {param3}, Limit used {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} crypto limitation causes insufficient 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 USD 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) |
| 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 does not exist. |
| 51014 | 200 | Index does not exist. |
| 51015 | 200 | Instrument ID does not match instrument type. |
| 51016 | 200 | Client order ID already exists. |
| 51017 | 200 | Loan amount exceeds borrowing limit. |
| 51018 | 200 | User with option account cannot hold net short positions. |
| 51019 | 200 | No net long positions can be held under cross margin mode in options. |
| 51020 | 200 | Order amount should be greater than the min available amount. |
| 51021 | 200 | The pair or contract is not yet listed |
| 51022 | 200 | Contract suspended. |
| 51023 | 200 | Position does not 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. |
| 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 does not 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 positions at 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 | Account does not have permission to submit MM quote order |
| 51036 | 200 | Only Options instrument of the PM account supports MMP orders. |
| 51411 | 200 | Account does not have permission for mass cancellation |
| 51042 | 200 | Under the Portfolio margin account, users can only place MMP orders in cross margin mode in Options. |
| 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 net 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. |
| 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 configured buying lead trade pairs |
| 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' |
| 51076 | 200 | TP/SL orders in Split TPs only support one-way TP/SL. You can not 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 attachAlgoClOrdId 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 attachAlgoClOrdId. |
| 51100 | 200 | Unable to place order. Take profit/Stop loss conditions cannot be added to reduce-only orders. |
| 51101 | 200 | Order failed. The size of the current order cannot be more than {maxSzPerOrder} (contracts). |
| 51102 | 200 | Order failed. The number of pending orders for this instId cannot be more than {maxNumberPerInstrument} (orders). |
| 51103 | 200 | Order failed. The number of pending orders across all instIds under the current {businessType} instFamily cannot be more than {maxNumberPerInstFamily} (orders). |
| 51104 | 200 | Order failed. The aggregated contract quantity for all pending orders across all instIds under the current {businessType} instFamily cannot be more than {maxSzPerInstFamily} (contracts). |
| 51105 | 200 | Order failed. The maximum sum of position quantity and pending orders in the same direction for current instId cannot be more than {maxPositionSzPerInstrument} (contracts). |
| 51106 | 200 | Order failed. The maximum sum of position quantity and pending orders in the same direction across all instIds under the current {businessType} instFamily cannot be more than {maxPostionSzPerInstFamily51106} (contracts). |
| 51107 | 200 | Order failed. The maximum sum of position quantity and pending orders in both directions across all instIds under the current {businessType} instFamily cannot be more than {maxPostionSzPerInstFamily51107} (contracts). |
| 51108 | 200 | Positions exceed the limit for closing out with the market price. |
| 51109 | 200 | No available offer. |
| 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 your available size. |
| 51113 | 429 | Market-price liquidation requests too frequent. |
| 51116 | 200 | Order price or trigger price exceeds {param0}. |
| 51117 | 200 | Pending close-orders count exceeds limit. |
| 51120 | 200 | Order quantity is less than {param0}. Please try again. |
| 51121 | 200 | Order quantity must be a multiple of the lot size. |
| 51122 | 200 | Order price must be higher than the minimum price {param0}. |
| 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. |
| 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. The maximum buy price is {param0}. |
| 51136 | 200 | Your closing price has triggered the limit price. The minimum 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. |
| 51143 | 200 | Insufficient conversion amount |
| 51147 | 200 | To trade options, make sure you have more than 20,000 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. |
| 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. |
| 51155 | 200 | Due to local compliance requirements, trading of this pair or contract is restricted. |
| 51169 | 200 | Failed to place order. You don’t have any positions of this contract. Turn off the Reduce-only to continue. |
| 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. |
| 51174 | 200 | Order failed. The number of {param0} pending orders reached the upper limit of {param1} (orders). |
| 51175 | 200 | Parameters {param0} {param1} and {param2} cannot be empty at the same time |
| 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 spot mode |
| 51180 | 200 | The range of {param0} should be ({param1}, {param2}) |
| 51181 | 200 | ordType must be limit when 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. |
| 51201 | 200 | Value of per market order cannot exceed 1,000,000 USDT. |
| 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. |
| 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%. |
| 51272 | 200 | Price variance must be 0 < x <= 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 |
| 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% |
| 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 | 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 does not support cancellation. |
| 51404 | 200 | Order cancellation unavailable during the second phase of call auction. |
| 51405 | 200 | Cancellation failed as you do not 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 does not match the order info. |
| 51409 | 200 | Either pair ID or pair name ID is required. |
| 51410 | 200 | Cancellation failed as the order is already under cancelling status. |
| 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. |
| 51500 | 200 | You must enter a price, quantity, or TP/SL |
| 51501 | 400 | Maximum of {param0} orders can be modified. |
| 51502 | 200 | Order failed. Insufficient {param0} balance in account |
| 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 (Portfolio margin mode can try IOC orders to lower the risks) |
| 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) |
| 51503 | 200 | Order modification failed as the order has been filled, canceled or does not exist. |
| 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. |
| 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 | Operation 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. |
| 51514 | 200 | Order modification failed. The price length must be 32 characters or shorter. |
| 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. |
| 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. |
| 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 |
| 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} contracts. |
| 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. |

#### 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 |
| 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} |

### 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 |
| 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 |

### 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. |
| 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. |
| 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 | Withdrawal fee exceeds the upper limit. |
| 58211 | 200 | Withdrawal fee is lower than the lower limit (withdrawal endpoint: incorrect fee). |
| 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. |
| 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 (rcvrInfo). For the exchange address, please also provide exchange information and recipient identity information ({consientParameters}). |
| 58238 | 200 | Incomplete info. The info of the exchange and the recipient are required if you're withdrawing to an exchange platform. |
| 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. |
| 58249 | 200 | API withdrawal for this currency is currently unavailable. Try withdrawing via our app or website. |
| 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. |
| 58400 | 200 | Request Failed |
| 58401 | 200 | Payment method is not supported |
| 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 {0}. 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 type corresponding to this {0} 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 |
| 59125 | 200 | {0} does not 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. |
| 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 |
| 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. |
| 59320 | 200 | Borrow quota exceeded |
| 59321 | 200 | Borrowing isn't available in your region. |
| 59322 | 200 | This action is unavailable for this order. |
| 59323 | 200 | Borrowing amount is less than minimum |
| 59324 | 200 | No available lending offer |
| 59325 | 200 | Repay the full loan amount to proceed. |
| 59326 | 200 | Invalid lending amount. Lending amount has to be between {minLend} to {lendQuota}. |
| 59327 | 200 | You can’t renew your loan order automatically because the amount you’re renewing isn’t enough to cover your current liability. Repay manually to avoid high overdue interest. |
| 59328 | 200 | Lending APR has to be between {minRate} to {maxRate}. |
| 59329 | 200 | Liability reduction failed. Repay this order instead. |
| 51152 | 200 | Holdings already reached the limit. |
| 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}. |
| 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 does not 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 {0} is suspended. |
| 59509 | 200 | Account doesn't have permission to reset market maker protection (MMP) status. |
| 59510 | 200 | Sub-account does not exist |
| 59512 | 200 | Unable to set up permissions for ND broker subaccounts. By default, all ND subaccounts can transfer funds out. |
| 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. |
| 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 margin-level status is safe. |
| 59650 | 200 | Switching your offset unit may increase the risk of liquidation. Adjust the size of your positions and ensure your margin-level status 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. |

### 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. |
| 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. |
| 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 clQuoteId. |
| 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. |
| 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 {quoteId} |
| 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 |
| 70511 | 200 | Execution is being processed |
| 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. |
| 56008 | 200 | Duplicate instruments in legs array. |

### Copy trading

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}× |
| 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 spot and 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. |
| 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. |
| 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 spot and 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 spot and 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 |

### 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. |

## WebSocket

### Public

Error Code from 60000 to 64002

#### 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} |
| 60020 | APIKey subscription amount exceeds the limit {0}. |
| 60021 | This operation does not support multiple accounts login. |
| 60022 | Bulk login partially succeeded |
| 60023 | Bulk login requests too frequent |
| 60024 | Wrong passphrase |
| 60025 | token subscription amount exceeds the limit {0} |
| 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. |
| 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: https://www.okx.com/help-center/changes-to-v5-api-websocket-subscription-parameter-and-url. |
| 64001 | This channel has been migrated to the '/business' URL. Please subscribe using the new URL. More details can refer to: https://www.okx.com/help-center/changes-to-v5-api-websocket-subscription-parameter-and-url. |
| 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: https://www.okx.com/help-center/changes-to-v5-api-websocket-subscription-parameter-and-url. |
| 64003 | Your trading fee tier doesn't meet the requirement to access this channel |

#### 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.

[HTTP](https://www.okx.com/docs-v5/en/) [Python](https://www.okx.com/docs-v5/en/)

