Batch Add

REQUEST wss://ws-auth.kraken.com/v2 batch_add *Authentication Required*

Sends a collection of orders (minimum of 2 and maximum 15):

- Validation is performed on the whole batch prior to submission to the engine. If an order fails validation, the whole batch will be rejected.
- On submission to the engine, if an order fails pre-match checks (i.e. funding), then the individual order will be rejected and remainder of the batch will be processed.
- All orders in batch are limited to a single pair.

Request

Request Schema Example

MESSAGE BODY

method string — REQUIRED

Value: batch_add

params object

deadline string

Format: RFC3339

Example: 2022-12-25T09:30:59.123Z

Range of valid offsets (from current time) is 500 milliseconds to 60 seconds, default is 5 seconds. The precision of this parameter is to the millisecond. The engine will prevent this order from matching after this time, it provides protection against latency on time sensitive orders.

symbol string — REQUIRED

Example: "BTC/USD"

The symbol of the currency pair.

validate boolean

Possible values: [true, false]

Default value: `false`

If set to `true` the order will be validated only, it will not trade in the matching engine.

`token` string — REQUIRED

This is a authenticated request, a session token is required.

`orders` array [

A list of orders in the batch.

`[many] order` object

`cash_order_qty` float — CONDITIONAL

Condition: market orders only.

Order volume expressed in quote currency.

`conditional` object

The conditional parameters are used as a template for generating the secondary close orders when the primary order fills. Each fill on the primary order will generate a new secondary order. The size of the secondary order will be the same size as the executed quantity and have the opposite side.

`order_type` string

Possible values: [`limit`, `stop-loss`, `stop-loss-limit`, `take-profit`, `take-profit-limit`, `trailing-stop`, `trailing-stop-limit`]

Defines the order type of the secondary close orders which will be created on each fill.

`limit_price` float

Defines the limit price on the secondary close orders. Only required on secondary order types that support limit price: `limit`, `stop-loss-limit`, `take-profit-limit`.

`limit_price_type` string — CONDITIONAL

Condition: Only available on trailing-stop-limit orders.
Possible values: [`static`, `pct`, `quote`]
Default value: `quote`

The units for the limit price on the secondary order.

- `static`: a static market price for the asset, i.e. 30000 for BTC/USD.
- `pct`: a percentage offset from the reference price, i.e. -10% from index price.
- `quote`: a notional offset from the reference price in the quote currency, i.e, 150 BTC/USD from last price

Note, for `trailing-stop-limit` order type, the value represents offset from the trigger price. 0 would set a limit price the same as the trigger price.

`trigger_price` float

Specifies the amount for the trigger price - it supports both static market prices and relative prices. This field is used in combination with the `price_type` field below to determine the effective trigger price.

Examples:

- To trigger at 29000.5 BTC/USD, use price=29000.5, price_type=static.
- To trigger when price rises by 5%, use price=5, price_type=pct.
- To trigger when price drops by 150 USD, use price=-150, price_type=quote.

Note, for `trailing-stop` and `trailing-stop-limit` order types, the price represents the reversion from the peak. It is always a positive offset value.

`trigger_price_type` string

Possible values: [`static`, `pct`, `quote`]

Default value: `static`

The units for the trigger price.

- `static`: a static market price for the asset, i.e. 30000 for BTC/USD.
- `pct`: a percentage offset from the reference price, i.e. -10% from index price.
- `quote`: a notional offset from the reference price in the quote currency, i.e, 150 BTC/USD from last price

~~`stop_price`~~ float — DEPRECATED

Deprecated Usage: Use trigger_price

Defines the trigger price on the secondary close orders. Only required on triggered secondary order types: `stop-loss`, `stop-loss-limit`, `take-profit`, `take-profit-limit`.

`display_qty` float — CONDITIONAL

Condition: iceberg orders only.

Defines the quantity to show in the book while the rest of order quantity remains hidden. Minimum value is 1 / 15 of `order_qty`.

`effective_time` string

Format: RFC3339
Example: 2022-12-25T09:30:59Z

Scheduled start time (precision to seconds).

`expire_time` string — CONDITIONAL

Condition: GTD orders only.
Format: RFC3339
Example: 2022-12-25T09:30:59Z

Expiration time of the order (precision to seconds). GTD orders can have an expiry time up to one month in future.

`fee_preference` string

Possible values: [`base` , `quote`]

Fee preference base or quote currency. `quote` is the default for buy orders, `base` is the default for sell orders.

`limit_price` float

Limit price for order types that support limit price restriction.

`limit_price_type` string — CONDITIONAL

Condition: Only available on trailing-stop orders.
Possible values: [`static` , `pct` , `quote`]
Default value: `quote`

The units for the limit price.

`margin` boolean

Possible values: [`false` , `true`]
Default value: `false`

Funds the order on margin using the maximum leverage for the pair (maximum is leverage of 5).

`no_mpp` boolean — CONDITIONAL

Condition: Market orders only.
Possible values: [`true` , `false`]
Default value: `false`

Disables Market Price Protection (MPP) if set to `true` . MPP is a feature that protects market orders from filling at a bad price due to price slippage in an illiquid or volatile market. See MPP support article.

cl_ord_id string

Adds a alphanumeric client order identifier which uniquely identifies an open order for each client. This field is mutually exclusive with `order_userref` parameter.

The `cl_ord_id` parameter can be one of the following formats:

- Long UUID: `6d1b345e-2821-40e2-ad83-4ecb18a06876` 32 hex characters separated with 4 dashes.
- Short UUID: `da8e4ad59b78481c93e589746b0cf91f` 32 hex characters with no dashes.
- Free text: `arb-20240509-00010` Free format ascii text up to 18 characters.

order_userref integer

This is an optional non-unique, numeric identifier which can associated with a number of orders by the client. This field is mutually exclusive with `cl_ord_id` parameter.

Many clients choose a unique integer value generated by their systems (i.e. a timestamp). However, because we don't enforce uniqueness on our side, it can also be used to easily tag a group of orders for querying or cancelling.

order_qty float — REQUIRED

Order quantity in terms of the base asset.

order_type string — REQUIRED

Possible values: [`limit`, `market`, `iceberg`, `stop-loss`, `stop-loss-limit`, `take-profit`, `take-profit-limit`, `trailing-stop`, `trailing-stop-limit`, `settle-position`]

The execution model of the order.

post_only boolean — CONDITIONAL

Condition: Orders with limit price only.
Possible values: [`true`, `false`]
Default value: `false`

Cancels the order if it will take liquidity on arrival. Post only orders will always be posted passively in the book.

reduce_only boolean

Possible values: [`true`, `false`]
Default value: `false`

Reduces an existing margin position without opening an opposite long or short position worth more than the current value of your leveraged assets.

side string — REQUIRED

Possible values: [`buy` , `sell`]

Side of the order.

stp_type string

Possible values: [`cancel_newest` , `cancel_oldest` , `cancel_both`]
Default value: `cancel_newest`

Self Trade Prevention (STP) is a protection feature to prevent users from inadvertently or deliberately trading against themselves. To prevent a self-match, one of the following STP modes can be used to define which order(s) will be expired:

- `cancel_newest` : arriving order will be canceled.
- `cancel_oldest` : resting order will be canceled.
- `cancel_both` : both arriving and resting orders will be canceled.

time_in_force string

Possible values: [`gtc` , `gtd` , `ioc`]
Default value: `gtc`

Time-in-force specifies how long an order remains in effect before being expired.

- `gtc` : Good Till Canceled - until user has cancelled.
- `gtd` : Good Till Date - until `expire_time` parameter.
- `ioc` : Immediate Or Cancel - immediately cancels back any quantity that cannot be filled on arrival.

triggers object — CONDITIONAL

Condition: Required for triggered order types only.

The parameters for setting the trigger price conditions.

reference string

Possible values: [`index` , `last`]
Default value: `last`

The reference price to track for triggering orders.

- `index` : the index price in the broader market (for this pair). Note, to keep triggers serviceable during connectivity issues with external index feeds, the last price will be used as the reference price.

- `last` : the last traded price in the Kraken order book (for this pair).

`price` float — REQUIRED

Specifies the amount for the trigger price - it supports both static market prices and relative prices. This field is used in combination with the `price_type` field below to determine the effective trigger price.

Examples:

- To trigger at 29000.5 BTC/USD, use price=29000.5, price_type=static.
- To trigger when price rises by 5%, use price=5, price_type=pct.
- To trigger when price drops by 150 USD, use price=-150, price_type=quote.

Note, for `trailing-stop` and `trailing-stop-limit` order types, the price represents the reversion from the peak. It is always a positive offset value.

`price_type` string

Possible values: [`static` , `pct` , `quote`]
Default value: `static`

The units for the trigger price.

- `static` : a static market price for the asset, i.e. 30000 for BTC/USD.
- `pct` : a percentage offset from the reference price, i.e. -10% from index price.
- `quote` : a notional offset from the reference price in the quote currency, i.e, 150 BTC/USD from last price

`sender_sub_id` string — CONDITIONAL

Condition: For institutional accounts with enhanced Self Trade Prevention (STP)

Adds a alphanumeric sub-account/trader identifier which enables STP to be performed at a more granular level.

The `sender_sub_id` parameter can be one of the following formats:

- Long UUID: `6d1b345e-2821-40e2-ad83-4ecb18a06876` 32 hex characters separated with 4 dashes.
- Short UUID: `da8e4ad59b78481c93e589746b0cf91f` 32 hex characters with no dashes.
- Free text: `arb-20240509-00010` Free format ascii text up to 18 characters.

`stop_price` float — DEPRECATED

Deprecated Usage: Use 'triggers' object.

The stop price for trigger order types.

~~trigger~~ string — DEPRECATED

Deprecated Usage: Use 'triggers' object.

Possible values: [`last` , `index`]

Default value: `last`

The reference price to trigger the order.

- `index` : the index price for the broader market for this symbol.
- `last` : the last traded price in the order book for this symbol.

]

token string — REQUIRED

This is a authenticated channel, a session token is required. See guides on how to generate a token via REST.

req_id integer

Optional client originated request identifier sent as acknowledgment in the response.

Response

Response Schema Example

The order of returned txid's in the response array is the same as the order of the order list sent in request.

MESSAGE BODY

method string

Value: `batch_add`

result array of objects — CONDITIONAL

Condition: On successful requests only

order_id string

Unique order identifier generated by Kraken.

cl_ord_id string

An optional, alphanumeric identifier specified by the client in the `batch_add` parameters.

order_userref integer

An optional order identifier specified by the client in the `batch_add` parameters.

warnings array of strings

An advisory message, highlighting deprecated fields or upcoming changes to the request.

error string — CONDITIONAL

Condition: On unsuccessful requests only

The error message for a rejected request.

success boolean

Possible values: [`true`, `false`]

Indicates if the request was successfully processed by the engine.

req_id integer

Optional client originated request identifier sent as acknowledgment in the response.

time_in string

Format: RFC3339
Example: 2022-12-25T09:30:59.123456Z

The timestamp when the request was received on the wire, just prior to parsing data.

time_out string

Format: RFC3339
Example: 2022-12-25T09:30:59.123456Z

The timestamp when the response was sent on the wire, just prior to transmitting data.