Executions

CHANNEL wss://ws-auth.kraken.com/v2 `executions` *Authentication Required*

The `executions` channel streams order status and execution events for this account.

It corresponds to a combination of the following Websockets v1 channels: `openOrders` and `ownTrades`.

This channel contains account specific data, an authentication token is required in the request.

Subscribe Request

Subscribe Schema | Subscribe Ack Schema | Example: Subscribe | Example: Subscribe Ack

MESSAGE BODY

- **method** string — REQUIRED
  - Value: `subscribe`
- **params** object
  - **channel** string — REQUIRED
    - Value: `executions`
  - **snap_trades** boolean
    - Possible values: [`true`, `false`]
    - Default value: `false`
    - If `true`, the last 50 order fills will be included in snapshot.
  - **snap_orders** boolean
    - Possible values: [`true`, `false`]
    - Default value: `true`
    - If `true`, open orders will be included in snapshot.
  - **order_status** boolean

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

If `true`, all possible status transitions will be sent. Otherwise, only open / close transitions will be streamed: `new`, `filled`, `canceled`, `expired`.

`ratecounter` boolean

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

If `true`, the rate-limit counter is included in the stream.

~~`snapshot_trades`~~ boolean — DEPRECATED

Deprecated Usage: Use 'snap_trades' field.
Possible values: [`true`, `false`]

If `true`, snapshot provides only trade events. Otherwise, open orders and trades will be included in snapshot.

~~`snapshot`~~ boolean — DEPRECATED

Deprecated Usage: Use 'snap_orders' or 'snap_trades' field.
Possible values: [`true`, `false`]

Request a snapshot after subscribing.

`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.

Snapshot / Update Responses

The snapshot and update stream share the same data schema, the fields included in the message is dependant on the `exec_type`.

By default, the snapshot response contains all open orders and latest 50 trades.

The snapshot message content can be adjusted with the subscription parameters.

Snapshot / Update Schema Example: Pending Example: Live Order Example: Execution

MESSAGE BODY

- **channel** string

  Value: `executions`

- **type** string

  Possible values: [`snapshot`, `update`]

- **data** array [

  A list of execution reports: order status and fills.

  - [many] execution_report object

    - **amended** boolean

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

      Indicates if the order has been amended, the modification history can be extracted from the REST `OrderAmends` endpoint. This field is present in the snapshot and the `amended`, `restated` event types.

    - **avg_price** float

      Order's average fill price.

    - **cash_order_qty** float

      Order volume expressed in quote currency (if specified on the original order).

    - **cl_ord_id** string

      Optional client identifier associated with the order.

    - **contingent** object

      The contingent object describes the template for generating the secondary close orders when the primary order fills.

      - **order_type** string

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

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

      - **trigger_price** float — CONDITIONAL

        Condition: Only on triggered secondary order types.

Describes the trigger price amount on the secondary orders. This field is used in combination with the `contingent.trigger_price_type` field to determine the effective trigger price.

`trigger_price_type` string — CONDITIONAL

Condition: Only on triggered secondary order types.
Possible values: [`static` , `pct` , `quote`]

Describes trigger price units on the secondary orders.

- `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

`limit_price` float — CONDITIONAL

Condition: Only on secondary order types that support limit price.

Describes limit price amount on the secondary orders. This field is used in combination with the `contingent.limit_price_type` field to determine the effective limit price.

`limit_price_type` string — CONDITIONAL

Condition: Only on secondary order types that support limit price.
Possible values: [`static` , `pct` , `quote`]

Describes limit price units on the secondary orders.

- `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

`cost` float — CONDITIONAL

Condition: trade events only.

Value of an individual execution.

`cum_cost` float

The order cumulative value executed.

`cum_qty` float

The order cumulative executed quantity.

display_qty float

Display quantity for iceberg order types.

display_qty_remain float — CONDITIONAL

Condition: Iceberg Order

Indicates next display_qty in Iceberg order.

effective_time string

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

Scheduled start time of the order.

exec_id string — CONDITIONAL

Condition: trade events only.

Execution identifier.

exec_type string

Possible values: [`pending_new`, `new`, `trade`, `filled`, `iceberg_refill`, `canceled`, `expired`, `restated`, `status`]

Describes the type of order event and determines the set of fields in the message.

- `pending_new`: Order request has been received and validated but the order is not live yet.
- `new`: Order has been created and is live in the engine.
- `trade`: The order has received a fill.
- `filled`: The order has been fully filled.
- `canceled`: The order has been cancelled.
- `iceberg_refill`: Indicates an Iceberg order refill.
- `expired`: The order has expired.
- `amended`: There is a user initiated amend on the order, i.e. limit price change.
- `restated`: There is a engine initiated amend on the order for maintenance of position or book, see `reason` field, i.e. reduce non-tradable liquidity.
- `status`: The order has a status update, i.e. trigger price has been updated.

expire_time string

Format: RFC3339

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

Scheduled expiration time of the order.

ext_ord_id string

Format: UUID

An optional, external partner order identifier shown on order events.

ext_exec_id string

Format: UUID

An optional, external partner execution identifier shown on trade events.

fees array [CONDITIONAL

Condition: trade events only.

The fees paid on this trade event. Currently, the fees are expressed in the quote currency only.

[0] fee object

asset string
The fee currency.

qty float
The fee amount.

]

fee_ccy_pref string

The preferred currency for paying fees.

- `fcib`: prefer fee in base currency.
- `fciq`: prefer fee in quote currency.

fee_usd_equiv float

The total fee paid in USD.

limit_price float

Limit price for order types that support limit price restriction.

liquidated boolean

Indicates if the order has been liquidated by the engine.

liquidity_ind string

Possible values: [`m`, `t`]

The liquidity indicator: `t` taker, `m` maker.

`last_price` float — CONDITIONAL

Condition: trade events only.

The average price in this trade event.

`last_qty` float — CONDITIONAL

Condition: trade events only.

The quantity filled in this trade event.

`margin` boolean

Indicates if the order can be funded on margin.

`margin_borrow` boolean

Indicates if an execution is on margin, i.e. if the trade increased or reduced size of margin borrowing. On trade events only.

`no_mpp` boolean

Indicates if the order has market price protection.

`ord_ref_id` string

Referral order transaction id that created this order.

`order_id` string

Unique order identifier generated by Kraken.

`order_qty` float

The client order quantity.

`order_type` string

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.

`order_status` string

Describes current state of the order.

- `pending_new`: Order has been received but not yet created by the engine.
- `new`: Order is live but has no fills.
- `partially_filled`: Order is live and some fills.
- `filled`: The order has been fully filled.
- `canceled`: The order has been cancelled.

- `expired` : The order has expired.

`order_userref` integer

Optional numeric, client identifier associated with one or more orders.

`post_only` boolean

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

Indicates a post only order.

`position_status` string

Possible values: [`opened` , `closing` , `closed`]

Indicates status of the position on a margin order.

`reason` string

The reason associated with an event, if applicable.

`reduce_only` boolean

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

Indicates a reduce only order.

`sender_sub_id` string

For institutional accounts, identifies underlying sub-account/trader for Self Trade Prevention (STP).

`side` string

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

Side of the order.

`symbol` string

Example: "BTC/USD"

The symbol of the currency pair.

`time_in_force` string

Possible values: [`GTC` , `GTD` , `IOC`]

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

- `GTC` : Good Till Canceled
- `GTD` : Good Till Date
- `IOC` : Immediate Or Cancel

timestamp string

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

Time of the event.

trade_id integer

The trade identifier.

triggers object

Describes the parameters and status of the price trigger for triggered order types.

reference string

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

The reference price tracked for triggering orders.

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.

price_type string

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

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

actual_price float

The current value of the effective trigger price, this is useful if the trigger was entered using a relative price or the trigger price changes over time.

peak_price float

The peak / trough price on `trailing-stop` and `trailing-stop-limit` orders.

last_price float

On trigger activation, the value of the reference last price that triggered the order.

status string

Possible values: [`triggered`, `untriggered`]

The status is set to `triggered` when the trigger conditions are met and the order becomes active.

timestamp string

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

On trigger activation, the timestamp of the trigger event.

~~**cancel_reason**~~ string — DEPRECATED

Deprecated Usage: Use 'reason' field.

Cancellation reason.

~~**stop_price**~~ float — DEPRECATED

Deprecated Usage: Use 'triggers' object.

The stop price for triggered order types.

~~**trigger**~~ string — DEPRECATED

Deprecated Usage: Use 'triggers' object.
Possible values: [`index`, `last`]

Reference price for triggered order types.

~~**triggered_price**~~ float — DEPRECATED

Deprecated Usage: Use 'triggers' object.

Price which triggered the order.

]

Unsubscribe Request

Unsubscribe Schema | Unsubscribe Ack Schema | Example: Unsubscribe | Example: Unsubscribe Ack

MESSAGE BODY

method string — REQUIRED

Value: `unsubscribe`

params object

channel string — REQUIRED

Value: `executions`

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.