Kaiko

MIGRATION GUIDE

Options Trades

HTTP REST

Amberdata → Kaiko

2026

www.kaiko.com

1 At a glance

What this endpoint gives you. The raw, tick-level trade history for a single exchange and options contract (underlying, expiry, strike and type), with price, size, side, trade ID and timestamp, over a date range you choose.

What changes for you

- **Block trades are not included** in Kaiko options trades. Trade counts and volumes will differ wherever block trades occurred (see §7).
- **Kaiko does not return the options-specific analytics fields Amberdata provides on this endpoint — by design.** `indexPrice`, `markPrice`, `underlyingPrice`, `iv`, `tickDirection` and `underlyingIndex` are intentionally kept out of the trades payload: they belong to Kaiko's **derivatives metrics** product. If your integration reads any of these from Amberdata's trades payload, plan to source them from that separate Kaiko product instead (see §5 and §7).
- `exchange` and `instrument` move to the **URL path**, plus a new required `instrument_class` segment (`option`), same pattern as Futures Trades.
- **The same real-world exchange can have a different Kaiko code per market.** Binance is `binc` for spot/futures but `bopt` for options (confirmed) — don't reuse the Spot Trades venue table as-is (see §6).
- Instrument code construction changes (dash-separated → concatenated), and the expiry-date substring format is exchange-native and not uniform (see §6).
- Amberdata caps this endpoint at a **1-hour** date range per request (documented), tighter than Spot/Futures Trades — more pagination needed for backfills.

Coverage & distribution

Data type: Tick-level (per-trade) options trade records

Source: One CEX options exchange + instrument per request

Granularity: Per-trade (no aggregation)

Distribution: REST API

2 Endpoint & authentication

	Amberdata (today)	Kaiko (new)
Method	GET	GET
Base URL	`https://api.amberdata.com`	`https://us.market-api.kaiko.io` (or `eu.market-api.kaiko.io`)
Path	`/markets/options/trades/{instrument}`	`/v4/data/exchanges/{exchange}/option/{instrument}/trades`
Auth	API key header `x-api-key`	API key header `X-Api-Key` (same mechanism, new key)

`{instrument_class}` is `option` (singular) — confirmed via a real call: `.../exchanges/bopt/option/btc26112791000c/trades`.

Kaiko API family: Classic (versioned)

This endpoint belongs to Kaiko's existing versioned API. Its URL structure and response envelope differ from Kaiko's New API grammar endpoints (`market-data`), such as OHLCV Batch and Tickers.

3 Request parameters

Amberdata param	Kaiko param	Change	Notes
—	`{instrument_class}` (path, required)	NEW	**No Amberdata equivalent.** Value is `option` . Same new requirement as Futures Trades' `future` / `perpetual-future` .
`{instrument}` (path, required)	`{instrument}` (path, required)	RENAMED	Same position (URL path), different construction — see §6.
`exchange` (query, required)	`{exchange}` (path, required)	RENAMED	Moves to a URL path segment. Per Amberdata's docs, this endpoint accepts a **comma-separated list** of exchanges (or all, if omitted); Kaiko always takes exactly one. Exchange codes also change, and can differ by market for the same venue (see §6).
`startDate` (optional)	`start_time` (optional)	RENAMED	Amberdata accepts epoch seconds, epoch milliseconds, or ISO 8601, and **caps the range at 1 hour** between `startDate` and `endDate` (documented). Inclusive lower bound on both Amberdata and Kaiko. Kaiko accepts **ISO 8601 only** and has **no 1-hour cap**: as on Spot and Futures Trades, any date range is served page by page (`page_size` , `continuation_token`).
`endDate` (optional)	`end_time` (optional)	RENAMED	Exclusive upper bound on both Amberdata and Kaiko.
`timeFormat` (optional)	—	REMOVED	Kaiko always returns `timestamp` as unix nanoseconds; no alternate format option.
`sortDirection` (optional)	`sort` (optional)	RENAMED	Same `asc` / `desc` values; Kaiko defaults to `desc` .
—	`page_size` , `continuation_token` (optional)	NEW	Pagination controls; no Amberdata equivalent param for this endpoint.

4 Response envelope & metadata

Identical envelope mechanics to Spot and Futures Trades — see those guides for the full breakdown. Summary below.

Amberdata field	Kaiko field	Change	Notes
`status` / `title` / `description`	`result`	RENAMED	**Adaptation required.** Single string (e.g. `"success"`) replaces the HTTP-style status object.

Amberdata field	Kaiko field	Change	Notes
`payload.data`	`data`	RENAMED	**Array** holding the trade records (see §5), now at the top level.
`payload.metadata.api-version`	`query.data_version`	RENAMED	Data version surfaced inside the echoed `query` object.
`payload.metadata.next`	`continuation_token` + `next_url`	RENAMED	Pagination mechanism differs — rewrite pagination logic, don't just rename.
—	`time`, `timestamp`	NEW	Server response time. No Amberdata equivalent.

5 Response - data fields

Amberdata field	Kaiko field	Type	Change	Notes
`instrument`	(path) / `query.instrument`	string	REMOVED	Not repeated per record, but recoverable from the request path or the echoed `query.instrument`.
`exchange`	(path) / `query.exchange`	string	REMOVED	Not repeated per record, but recoverable from the request path or the echoed `query.exchange`.
`exchangeTimestamp` + `exchangeTimestampNanoseconds`	`timestamp`	int+int → integer	RENAMED	Merged into one unix-nanosecond integer, same as Spot/Futures. On this endpoint Amberdata's `exchangeTimestamp` is an epoch-millisecond integer; on Spot and Futures it is a human-readable string.
`isBuySide`	`taker_side_sell`	boolean	RENAMED	**Adaptation required.** Same inverted convention confirmed for Spot Trades (§7 there): `isBuySide = NOT taker_side_sell`.
`price`	`price`	number → string	IDENTICAL NAME	**Adaptation required.** Same field name, JSON type changes from number to string.
`volume`	`amount`	number → string	RENAMED	For options this is the **underlying asset quantity** (like Spot Trades), **not** a contract count (unlike Futures Trades) — mind the difference if you've migrated both.
`tradeId`	`trade_id`	string	RENAMED	Casing only.
`sequence`	`sequence`	→ null	IDENTICAL NAME	Both `null` in the sample reviewed. Kaiko's is reserved for future use regardless (see Spot Trades §5).
`quoteVolume`	`quote_volume`	number (nullable) → string	RENAMED	Same pattern as Spot/Futures: Kaiko **computes** it and populates it (`"3068.7"` in the sample) even where Amberdata returns `null`.

Amberdata field	Kaiko field	Type	Change	Notes
`tickDirection`	—		REMOVED	**Intentionally out of scope for this endpoint** — belongs to Kaiko's derivatives metrics product (see §7).
`indexPrice`	—		REMOVED	**Intentionally out of scope for this endpoint** — belongs to Kaiko's derivatives metrics product. Populated on Amberdata (e.g. `83340.36782609` in the sample).
`markPrice`	—		REMOVED	**Intentionally out of scope for this endpoint** — belongs to Kaiko's derivatives metrics product.
`underlyingIndex`	—		REMOVED	**Intentionally out of scope for this endpoint** — belongs to Kaiko's derivatives metrics product.
`underlyingPrice`	—		REMOVED	**Intentionally out of scope for this endpoint** — belongs to Kaiko's derivatives metrics product.
`iv`	—		REMOVED	**Intentionally out of scope for this endpoint** — belongs to Kaiko's derivatives metrics product. Implied volatility at trade time — if you consume this, source it from that product instead.

6 Exchange & instrument names

Kaiko identifies venues and instruments with its own normalized codes, which differ from Amberdata's names. Codes can also differ by market for the same venue (e.g. Binance spot and options).

Reference

The mapping rules and the full code lists apply to every endpoint and are described in the **Exchange, Asset & Instrument Mapping** migration guide (`exchange-instrument-asset-codes.pdf`).

7 Will your values change?

Biggest difference for this product: a structural product split, not a gap

For Spot and Futures Trades, every Amberdata field had a Kaiko counterpart (even if renamed or retyped). That is different here: `indexPrice` , `markPrice` , `underlyingPrice` , `iv` , `tickDirection` and `underlyingIndex` are absent from Kaiko's trades response **by design**, confirmed: Amberdata bundles these six metrics fields into its options trades payload, while Kaiko keeps trades and derivatives metrics as **separate products**. If a client depends on implied volatility, mark price, or the underlying index/price at trade time, **the adaptation is to source those fields from Kaiko's derivatives metrics product instead** — consistent with Kaiko's overall separation of trades from derivatives metrics (see the Funding Rates guide for the sibling case: that product also lives outside the trades endpoints). This is a structural integration item to plan for, not a field mapping to fix in client code.

Short answer, for the fields both sides do share

Same as Spot and Futures Trades: the shared trade facts (price, amount, timestamp, side) are the exchange's raw executed trades on both sides, tick-for-tick, with the same encoding differences and the same highest-risk adaptations, **except block trades, which Kaiko excludes**.

- **Block trades are excluded.** Kaiko's options trades do not include block trades. Trade counts and volumes will differ wherever block trades occurred.
- **Trade side is inverted.** `isBuySide = NOT taker_side_sell`.
- `price` **and** `amount` **change JSON type** (number → string).
- **Status/error handling changes shape** (`status` / `title` / `description` → single `result` string).
- **Exchange code depends on market, not just venue.** `binance` is `bopt` here, not `binc` (see §6).
- `amount` **= underlying quantity, not contracts.** Different from Futures Trades' same-named field.
- **Tighter rate/range constraints on Amberdata.** 1-hour max date range per request (documented) — expect more pagination for historical backfills than Spot/Futures.
- `quote_volume`**.** Kaiko always computes it; expect it populated even where Amberdata's `quoteVolume` is `null`.

8 Before / after example

Request (real)

Amberdata

```
GET /markets/options/trades/BTC-261127-91000-C
    ?exchange=binance
x-api-key: <your-key>
```

Kaiko

```
GET /v4/data/exchanges/bopt/option/
btc26112791000c/trades
X-Api-Key: <your-key>
```

Response (real, captured)

Amberdata


```
{
  "status": 200,
  "title": "OK",
  "description": "Successful request",
  "payload": {
    "metadata": {
      "next": null,
      "api-version": "2023-09-30"
    },
    "data": [
      {
        "instrument": "BTC-261127-91000-C",
        "exchange": "binance",
        "exchangeTimestamp": 1791379779578,
        "exchangeTimestampNanoseconds": 0,
        "isBuySide": false,
        "price": 2030,
        "volume": 0.01,
        "tradeId": "45",
        "quoteVolume": null,
        "sequence": null,
        "tickDirection": null,
        "indexPrice": 83340.36782609,
        "markPrice": null,
        "underlyingIndex": null,
        "underlyingPrice": null,
        "iv": null
      }
    ]
  }
}
```


Kaiko


```
{
  "result": "success",
  "time": "2026-10-08T09:41:17Z",
  "timestamp": 1791452477031,
  "query": {
    "exchange": "bopt",
    "instrument": "btc26112791000c",
    "instrument_class": "option",
    "start_time": "2010-01-01T00:00:00Z",
    "end_time": "2026-10-08T09:41:17.000291147Z",
    "page_size": "100",
    "sort": "desc"
  },
  "data": [
    {
      "timestamp": 1791443070823000000,
      "trade_id": "59",
      "price": "1930",
      "amount": "1.59",
      "taker_side_sell": false,
      "quote_volume": "3068.7",
      "sequence": null
    }
  ]
}
```


The two calls use different trades (real, independently captured examples, not a matched pair) — shown for envelope and field-mapping purposes. Note Kaiko's response carries none of Amberdata's `tickDirection` / `indexPrice` / `markPrice` / `underlyingIndex` / `underlyingPrice` / `iv` fields (see §5, §7).