Kaiko

MIGRATION GUIDE

Spot 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 instrument: every executed trade with its price, size, side, trade ID and timestamp, over a date range you choose.

What changes for you

- You point your code at the **Kaiko endpoint** and switch to Kaiko authentication, still an API key header (see §2).
- `exchange` and `instrument` move from query parameters to the **URL path**, and exchange/pair identifiers change format (see §3 and §6).
- The response envelope is completely different (see §4), and a few data fields are renamed, merged, or change type — see §5 for the full mapping.
- Two fields help reach parity with Amberdata: `quote_volume` and `sequence` (see §5 and §7).
- **Highest-risk item:** the trade-side flag is an **inverted** convention, confirmed by testing — a naive field-rename migration silently flips every trade side. See the highlighted box in §7 before you migrate.

Coverage & distribution

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

Source: One CEX spot 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/spot/trades/{instrument}`	`/v4/data/exchanges/{exchange}/spot/{instrument}/trades`
Auth	API key header `x-api-key`	API key header `X-Api-Key` (same mechanism, new key)

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}` (path, required)	`{instrument}` (path, required)	RENAMED	Same position (URL path) on both sides, but separator changes: `eth_usd` → `eth-usd` (see §6).
`exchange` (query, required; only 1 allowed)	`{exchange}` (path, required)	RENAMED	Moves from a query parameter to a **URL path segment** — URL construction code must change, not just the parameter name. Exchange codes also change, e.g. `gdax` → `cbse` (see §6).
`startDate` (optional)	`start_time` (optional)	RENAMED	Amberdata accepts epoch seconds, epoch milliseconds, or ISO 8601; Kaiko accepts **ISO 8601 only** — epoch inputs are no longer accepted. Inclusive lower bound on both Amberdata and Kaiko.
`endDate` (optional)	`end_time` (optional)	RENAMED	Same format restriction as `start_time` (ISO 8601 only). Exclusive upper bound on both Amberdata and Kaiko.
`timeFormat` (optional)	—	REMOVED	Kaiko always returns `timestamp` as unix nanoseconds; no alternate (ISO/human-readable) 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

The fields around the data, the wrapper your parser reads before reaching the numbers. This envelope is **identical on every Amberdata endpoint**; Kaiko's is a different shape but plays the same role.

Amberdata field	Kaiko field	Change	Notes
`status` / `title` / `description`	`result`	RENAMED	**Adaptation required.** Amberdata's HTTP-style status object (integer code + "OK" title + free-text description) is replaced by a single string, e.g. `"success"`. Error-handling code that branches on `status` / `title` must be rewritten, not just renamed.
`payload.data`	`data`	RENAMED	**Array** holding the trade records (see §5), now at the top level (no `payload` wrapper).
`payload.metadata.api-version`	`query.data_version`	RENAMED	Data version is surfaced inside the echoed `query` object instead of a dedicated `metadata` object.
`payload.metadata.next`	`continuation_token` + `next_url`	RENAMED	Pagination mechanism differs: Amberdata gives a ready-to-call `next` URL; Kaiko gives a `continuation_token` (also embedded in `next_url`). Pagination logic must be rewritten, not just field-renamed.
—	`time`, `timestamp`	NEW	Server response time, as ISO 8601 string and unix milliseconds. 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`	string+int → integer	RENAMED	The two Amberdata fields merge into one unix-**nanosecond** integer on Kaiko. No precision lost, just restructured into a single field.
`isBuySide`	`taker_side_sell`	boolean	RENAMED	**Adaptation required — opposite convention, confirmed by testing.** The two fields are inverted: `isBuySide = NOT taker_side_sell`. A naive field-rename migration silently flips every trade side.
`price`	`price`	number → string	IDENTICAL NAME	**Adaptation required.** Same field name, but the JSON type changes from number to string — a breaking change for typed/strict clients.
`volume`	`amount`	number → string	RENAMED	**Adaptation required.** Trade quantity; also changes type from number to string — a breaking change for typed/strict clients.
`tradeId`	`trade_id`	string	RENAMED	Casing only (camelCase → snake_case).
`sequence`	`sequence`	string → null (reserved)	IDENTICAL NAME	Amberdata populates a value; Kaiko's `sequence` is reserved for future use and currently returns `null`.
`quoteVolume`	`quote_volume`	number (nullable) → string	RENAMED	Kaiko **computes** this (price × amount) and populates it even when Amberdata returns `null` for venues that don't report it natively.

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?

Short answer

The underlying trade facts (price, amount, timestamp) are the exchange's raw executed trades on both sides, tick-for-tick. But the two APIs disagree on how a few of these facts are encoded — confirmed by testing below — so a migration that only renames fields will compile, run, and silently produce wrong data for some of them. These are spec differences, not defects, and each needs an explicit adaptation in client code.

Highest-risk adaptations (confirmed by testing)

- **Trade side is inverted.** `isBuySide` (Amberdata) and `taker_side_sell` (Kaiko) encode **opposite** conventions: `isBuySide = NOT taker_side_sell`. A rename-only migration flips every trade side without erroring.
- `price` **and** `amount` **/** `volume` **change JSON type.** Amberdata returns numbers; Kaiko returns strings. Typed or strict-schema clients will fail or silently mis-parse if not updated.
- **Status/error handling changes shape.** Amberdata's `status` / `title` / `description` trio becomes a single `result` string on Kaiko — error branches keyed on the old fields need to be rewritten.

- **Timestamp precision is preserved.** Amberdata splits the timestamp into a millisecond string plus a nanosecond remainder (`exchangeTimestampNanoseconds`); Kaiko merges both into one nanosecond-precision integer. No information loss, just a different representation.
- `instrument` **/** `exchange`**.** Removed from the per-trade record, but recoverable from the request path or the echoed `query` object — low risk, just a different lookup.
- **Pagination mechanism differs.** Amberdata's `payload.metadata.next` URL becomes Kaiko's `continuation_token` / `next_url` — loud failure if not adapted (pagination simply won't work), not a silent one.
- `quote_volume`**.** Kaiko always computes it (price × amount), so expect it to be populated for venues where Amberdata's `quoteVolume` is `null`.
- `sequence`**.** Currently always `null` on Kaiko (reserved for future use) while Amberdata already populates it — don't use it for dedup/ordering yet; use `trade_id` and `timestamp` instead.

8 Before / after example

Request

Amberdata

```
GET /markets/spot/trades/eth_usd
    ?exchange=gdax
    &startDate=2023-01-20T00:00:00
    &endDate=2023-01-20T00:10:00
    &timeFormat=hr
x-api-key: <your-key>
```

Kaiko

```
GET /v4/data/exchanges/krkn/spot/btc-usd/trades
X-Api-Key: <your-key>
```

The two calls below use different exchanges/instruments (`gdax` / `eth_usd` vs `krkn` / `btc-usd`) because each is a real, captured example from its respective source — shown for envelope and field-mapping purposes, not a side-by-side value comparison.

Response (real, captured; fields ordered as returned)

Amberdata


```
{
  "description": "Successful request",
  "payload": {
    "data": [
      {
        "instrument": "eth_usd",
        "exchange": "gdax",
        "exchangeTimestamp": "2023-01-20 00:00:00
137",
        "exchangeTimestampNanoseconds": 927000,
        "isBuySide": true,
        "price": 1550.96,
        "volume": 0.19342803,
        "tradeId": "414516225",
        "sequence": "41195335428",
        "quoteVolume": null
      },
      {
        "instrument": "eth_usd",
        "exchange": "gdax",
        "exchangeTimestamp": "2023-01-20 00:00:00
137",
        "exchangeTimestampNanoseconds": 927000,
        "isBuySide": true,
        "price": 1550.96,
        "volume": 0.2347599,
        "tradeId": "414516226",
        "sequence": "41195335430",
        "quoteVolume": null
      }
    ]
  }
}
```


Kaiko


```
{
  "result": "success",
  "time": "2026-09-07T14:02:47Z",
  "timestamp": 1788789767543,
  "query": {
    "exchange": "krkn",
    "instrument": "btc-usd",
    "instrument_class": "spot",
    "start_time": "2010-01-01T00:00:00Z",
    "end_time": "2026-09-07T14:02:47.463924339Z",
    "page_size": "100",
    "sort": "desc"
  },
  "data": [
    {
      "timestamp": 1788789762617314000,
      "trade_id": "107066603",
      "price": "79383.5",
      "amount": "0.00102531",
      "taker_side_sell": true,
      "quote_volume": "81.392696385",
      "sequence": null
    },
    {
      "timestamp": 1788789762617314000,
      "trade_id": "107066602",
      "price": "79383.5",
      "amount": "0.00047239",
      "taker_side_sell": true,
      "quote_volume": "37.499971565",
      "sequence": null
    }
  ]
}
```