Kaiko

MIGRATION GUIDE

# Spot Tickers

## HTTP REST

Amberdata → Kaiko

2026

www.kaiko.com

## 1 At a glance

**What this endpoint gives you.** Historical top-of-book snapshots for one spot pair on one exchange: best bid and ask with their sizes, mid and last price, timestamped and sequenced.

What changes for you

- **One Kaiko path for every market.** The instrument moves from the path to a query parameter and the market is set with `class` (see §2–3).
- **Field names move to snake_case**, e.g. `bidVolume` → `bid_volume` (see §5).
- **Timestamps are ISO 8601 strings.** The nanosecond and per-side timestamp fields are dropped; a `timestamp` field is added.
- **Numeric values are returned as strings**, not JSON numbers. Parse them as decimals.
- Each row now carries its `class`. The response envelope is simpler, with pagination in `meta.next_url` (see §4).

### Coverage & distribution

**Data type:** Top-of-book ticker snapshots (bid, ask, sizes, mid, last)

**Markets:** CEX spot

**Request scope:** One exchange and one instrument per call on both sides

**Distribution:** REST API

## 2 Endpoint & authentication

| | Amberdata (today) | Kaiko (new) |
|---|---|---|
| **Method** | GET | GET |
| **Base URL** | `https://api.amberdata.com/markets` | `https://api.kaiko.io` |
| **Path** | `/spot/tickers/{instrument}` | `/market-data/tickers` |
| **Auth** | API key header `x-api-key` | API key header `X-Api-Key` (same mechanism, new key) |

On Kaiko the instrument moves from the path to a query parameter, and the market is selected with `class`. The same Kaiko path serves spot, futures and options tickers.

### Kaiko API family: New API grammar

This endpoint uses Kaiko's new `market-data` grammar: every parameter is a query parameter and responses share the `data` + `meta` envelope. Kaiko's Classic (versioned) endpoints, such as Trades, use a different URL structure and envelope.

## 3 Request parameters

All Kaiko parameters are query parameters.

| Amberdata param | Kaiko param | Change | Notes |
|---|---|---|---|
| `instrument` (path, required) | `instrument` | RENAMED | One instrument per call on both sides. Kaiko instrument code, e.g. `btc-usdt` (see §6). |
| — | `class` | NEW | Market: `spot`. |
| `exchange` (query, required) | `exchange` | RENAMED | One exchange per call. Kaiko exchange code, e.g. `binc`, `drbt` (see §6). |
| `startDate` (optional) | `start_time` | RENAMED | **Inclusive on both sides.** Amberdata accepts seconds, milliseconds or ISO 8601 and defaults to the beginning of the current hour. Kaiko takes ISO 8601 (e.g. `2026-10-26T12:34:30.000Z`). **Optional: without dates, Kaiko returns the last hour.** With dates, there is no range cap: the range is served page by page (`page_size`). |
| `endDate` (optional) | `end_time` | RENAMED | **Exclusive on both sides.** ISO 8601 on Kaiko. Optional; defaults to now. |
| `sortDirection` (optional) | `sort` | RENAMED | Amberdata: `asc` (default) or `desc`, only for ranges within the last 24 hours. Kaiko: `asc` or `desc` (default, newest first). |
| `timeFormat` (optional) | — | REMOVED | Amberdata can return milliseconds, ISO 8601 or human-readable timestamps (default `hr` on spot and futures). Kaiko always returns ISO 8601 strings. |
| — | `page_size` | NEW | Optional. Number of rows per page. |
| `api-version` (header, optional) | `api_version` | RENAMED | A header on Amberdata; a query parameter on Kaiko, e.g. `2026-10-01`. |
| `Accept-Encoding` (header, required) | `Accept-Encoding` | IDENTICAL | Required by Amberdata (`gzip, deflate, br`); supported by Kaiko. |

### Error codes

Common to Kaiko's New API grammar endpoints:

- `400` : invalid parameter (value, format or precision not accepted).
- `403 not_entitled` : the requested asset or instrument is outside your licence. Lists are authorised all or nothing.
- `403 history_out_of_entitlement` : the time window goes beyond your key's history entitlement, including the default window when no dates are sent.
- `404 unknown_instrument` : a code that is not an instrument of the requested `class` / exchange; the response names it.

## 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**.

| Amberdata field | Kaiko field | Change | Notes |
|---|---|---|---|
| `status` | — | REMOVED | HTTP-style status integer (200). Not in the Kaiko body; rely on the HTTP status code. |
| `title` | — | REMOVED | "OK" label, no direct Kaiko equivalent. |
| `description` | — | REMOVED | Free-text status description. |
| `payload.data` | `data` | RENAMED | **Array** holding the ticker rows (see §5), now at the top level. |
| `payload.metadata` | `meta` | RENAMED | Metadata object, after `data`. |
| `payload.metadata.api-version` | `meta.api_version` | RENAMED | Date version of the API used, e.g. `2023-09-30` on Amberdata, `2026-10-01` on Kaiko. |
| — | `meta.count` | NEW | Number of items returned in `data` on this page. |
| `payload.metadata.next` | `meta.next_url` | RENAMED | Full URL of the next page when results are paginated (a cursor URL on Amberdata). Call it as-is until it is no longer returned. |
| — | `meta.url` | NEW | The query as requested, with every parameter it carries. |

## 5 Response - data fields

One row per ticker update. All Kaiko numeric values are returned as decimal strings; parse them as decimals.

| Amberdata field | Kaiko field | Change | Notes |
|---|---|---|---|
| `instrument` | `instrument` | RENAMED | Same name, Kaiko instrument code (see §6). |
| `exchange` | `exchange` | RENAMED | Same name, Kaiko exchange code, e.g. `drbt` (see §6). |
| — | `class` | NEW | Market of the row (e.g. `spot`, `option`). |
| `exchangeTimestamp` | `exchange_timestamp` | RENAMED | Timestamp provided by the exchange. Amberdata falls back to its own collection time when the exchange provides none; Kaiko keeps the two apart (`exchange_timestamp` and `timestamp`). Format: milliseconds or a human-readable string on Amberdata depending on `timeFormat`; ISO 8601 string on Kaiko (e.g. `2026-10-28T14:26:26.451Z`). |
| `exchangeTimestampNanoseconds` | — | REMOVED | Sub-millisecond part of the Amberdata timestamp. Kaiko timestamps carry millisecond precision. |

| Amberdata field | Kaiko field | Change | Notes |
|---|---|---|---|
| — | `timestamp` | NEW | **Kaiko collection timestamp** (when Kaiko received the update), ISO 8601. |
| `bid`, `ask`, `mid` | `bid`, `ask`, `mid` | RENAMED | Same names; number → decimal string. |
| `last` | `last` | RENAMED | Same name; decimal string. Can be `null` on both sides when the venue does not publish it. |
| `bidVolume` | `bid_volume` | RENAMED | Size at the best bid; decimal string. |
| `askVolume` | `ask_volume` | RENAMED | Size at the best ask; decimal string. |
| `sequence` | `sequence` | IDENTICAL | Venue sequence number as a string; `null` when the venue provides none. |
| `askTimestamp`, `bidTimestamp` | — | REMOVED | Per-side timestamps, `null` on Amberdata. |
| `askTimestampNanoseconds`, `bidTimestampNanoseconds` | — | REMOVED | Sub-millisecond parts of the per-side timestamps. |

## 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

Prices and sizes come from the same exchange order book, so values should match once rows are aligned on time.

- **Update capture.** A ticker row is a snapshot of the top of the book. Row frequency and capture timing can differ between providers, so rows will not match one-to-one; align them on time before comparing. Kaiko records each ticker update in real time, as received from the exchange: row frequency follows the venue (e.g. about 1 second on Deribit spot, 100 ms on Bybit perpetuals).
- **Timestamp precision.** Amberdata adds nanoseconds in a separate field; Kaiko timestamps carry milliseconds.
- **Last price.** `last` depends on what the venue publishes in its ticker feed; it was `null` in the Amberdata Binance sample and populated in the Kaiko Deribit sample.
- **Number representation.** Kaiko returns decimal strings, which keep the exact digits; Amberdata returns JSON numbers. Compare as decimals, not as text.

# 8 Before / after example

## Request

**Amberdata**

```
GET https://api.amberdata.com/markets
    /spot/tickers/eth_usdt
    ?exchange=binance
x-api-key: <your key>
```

**Kaiko**

```
GET https://api.kaiko.io
    /market-data/tickers
    ?exchange=binc
    &class=spot
    &instrument=eth-usdt
X-Api-Key: <your key>
```

## Response

Real samples. The two calls cover different venues, pairs and dates, so values are not comparable.

**Amberdata**


```
{
  "status": 200,
  "title": "OK",
  "description": "Successful request",
  "payload": {
    "metadata": {
      "next": "https://api.amberdata.com/...?
cursor=...",
      "api-version": "2023-09-30"
    },
    "data": [
      {
        "instrument": "eth_usdt",
        "exchange": "binance",
        "exchangeTimestamp": "2026-10-27 14:19:20
316",
        "exchangeTimestampNanoseconds": 72606,
        "bid": 2322.58,
        "ask": 2322.59,
        "mid": 2322.585,
        "last": null,
        "bidVolume": 29.5467,
        "askVolume": 7.0588,
        "sequence": "82549136133",
        "askTimestamp": null,
        "bidTimestamp": null,
        "askTimestampNanoseconds": null,
        "bidTimestampNanoseconds": null
      },
      ...
    ]
  }
}
```


**Kaiko**


```
{
  "data": [
    {
      "ask": "80612",
      "ask_volume": "0.0372",
      "bid": "80328",
      "bid_volume": "0.0657",
      "class": "spot",
      "exchange": "drbt",
      "exchange_timestamp":
"2026-10-28T14:26:26.451Z",
      "instrument": "btc-usdt",
      "last": "80778",
      "mid": "80470",
      "sequence": null,
      "timestamp": "2026-10-28T14:26:26.451Z"
    },
    ...
  ],
  "meta": {
    "api_version": "2026-10-01",
    "count": 2,
    "next_url": "https://api.kaiko.io/market-data/
tickers?...",
    "url": "https://api.kaiko.io/market-data/
tickers?..."
  }
}
```