Kaiko

MIGRATION GUIDE · REST

Order Book Snapshots

Amberdata → Kaiko

2026

www.kaiko.com

1 At a glance

What this endpoint gives you. Historical, point-in-time snapshots of an exchange's Level 2 order book: the resting **bids** and **asks** at each price level, as polled at regular intervals. On Amberdata these are the `order-book-snapshots` endpoints; on Kaiko it is the **Raw Order Book Snapshot** endpoint.

What changes for you

- **History is a rolling 30-day window.** Kaiko REST serves about the last 30 days of snapshots. A request whose `start_time` falls before that window returns **no data at all** (with `result: "success"`), not a partial result. Older history is delivered through Kaiko Cloud Delivery (see §7).
- **The number of price levels varies.** Kaiko returns the book within **±10% of the best bid and ask**, so the number of levels changes from one snapshot to the next and from one venue to another. Amberdata can return the exchange's full book (see §7).
- **Set the depth you want.** Without `maxLevel` Amberdata returns every level; without `limit_orders` Kaiko returns **10 levels per side** (see §3).
- **One Kaiko endpoint replaces three Amberdata endpoints.** Spot, futures and options use the same path; the market is a path segment (`instrument_class`, see §3).
- **Futures and options contract data moves to the Ticker endpoint.** Amberdata adds mark price, implied volatility, Greeks and open interest to each derivatives snapshot. On Kaiko the snapshot holds the book only; those fields come from the Kaiko **Ticker** endpoint, and perpetual funding rates from the Derivatives risk endpoint (see §5).
- **Snapshots are more frequent, and not on the minute.** Kaiko polls about every 30 seconds, against about 60 seconds on Amberdata. Kaiko timestamps are the actual poll times, so do not expect a snapshot at exactly `hh:mm:00` (see §7).
- **Default sort order is reversed.** Amberdata returns oldest first; Kaiko returns **newest first** unless you set `sort=asc`.
- **Not carried over:** `numOrders` (orders per level), `sequence` and `exchangeTimestampNanoseconds` have no Kaiko equivalent (see §5).
- **Prices and amounts are returned as strings**, not JSON numbers. Parse them as decimals (see §5).

Coverage & distribution

Data type: Level 2 order book snapshots (price and amount per level, bids and asks)

Markets: CEX spot, dated futures, perpetual futures and options. Amberdata uses one endpoint per market; one Kaiko endpoint covers all

Instruments per call: 1, on one exchange

Cadence: about one snapshot every 30 s (Amberdata: about every 60 s, on the minute)

Depth: up to ±10% from the best bid and ask; 10 levels per side by default (Amberdata: full book by default)

History via REST: rolling ~30 days (Amberdata: multi-year)

Distribution: REST API; older history via Kaiko Cloud Delivery

2 Endpoint & authentication

	Amberdata (today)	Kaiko (new)
Method	GET	GET
Base URL	`https://api.amberdata.com/markets`	`https://us.market-api.kaiko.io` `https://eu.market-api.kaiko.io`

Path	`/spot/order-book-snapshots/{instrument}`
	`/futures/order-book-snapshots/{instrument}`
	`/options/order-book-snapshots/{instrument}`
		`/v2/data/order_book_snapshots.v1/exchanges/{exchange}/{instrument_class}/{instrument}/snapshots/raw`

Contract fields (futures, options)	Same response as the book	`https://api.kaiko.io/market-data/tickers`
		Funding rate:
		`https://{us|eu}.market-api.kaiko.io/v2/data/derivatives.v2/risk`

Auth	API key header `x-api-key`	API key header `x-api-key` (same mechanism, new key), on all three Kaiko endpoints

The three Amberdata endpoints share the same core parameters; options and futures add contract fields to each snapshot, which move to the Kaiko Ticker endpoint (see §5). On Kaiko, the exchange and the market move into the path, and you choose a `us` or `eu` host.

3 Request parameters

Kaiko path segments (required): `exchange`, `instrument_class`, `instrument`. Query parameters (optional): `start_time`, `end_time`, `limit_orders`, `sort`, `page_size`, `continuation_token`.

Amberdata param	Kaiko param	Change	Notes
`spot` | `futures` | `options` (path)	`instrument_class` (path)	RENAMED	Values `spot`, `future`, `perpetual-future`, `option`. Amberdata `futures` maps to both `future` and `perpetual-future`.
`instrument` (path, required)	`instrument` (path)	RENAMED	Same position, Kaiko instrument code (e.g. `btc_usd` → `btc-usd` for spot); see §6.
`exchange` (query, required)	`exchange` (path)	RENAMED	Moves from query to path. One exchange per call (Amberdata options also accepts a comma-separated list). Kaiko exchange code (e.g. `gdax` → `cbse`); see §6.
`startDate` (optional)	`start_time`	RENAMED	Inclusive on both sides. ISO 8601 only on Kaiko (e.g. `2026-11-10T14:00:00Z`). **Must fall inside the rolling ~30-day window**, or the whole request returns no data (see §7).
`endDate` (optional)	`end_time`	RENAMED	**Exclusive on both sides.** ISO 8601 only on Kaiko. Amberdata's per-request range caps (18 months spot, 1 hour options) do not apply; long ranges are paginated.

Amberdata param	Kaiko param	Change	Notes
`maxLevel` (optional)	`limit_orders`	RENAMED	Levels returned **per side**. **Defaults differ:** Amberdata returns all available levels; Kaiko returns `10`. Set it explicitly. Set `1` for best bid and ask only. A large value does not extend depth past ±10% of the best price (see §7).
`timestamp` (optional)	—	REMOVED	Amberdata returns the snapshot closest to, at or before, a time. On Kaiko, request `end_time` = that time + 1 ms, `sort=desc`, `page_size=1`, with a `start_time` a few minutes earlier.
`sortDirection` (optional)	`sort`	RENAMED	**Defaults differ:** Amberdata `asc`, Kaiko `desc` (newest first). Amberdata only allows sorting within the most recent 24 hours; Kaiko sorts any range. Values `asc` / `desc`.
`timeFormat` (optional)	—	REMOVED	Kaiko returns each snapshot time both as Unix milliseconds (`poll_timestamp`) and ISO 8601 (`poll_date`).
`api-version` (header, optional)	—	REMOVED	Kaiko versions this endpoint in the path (`/v2/`, `order_book_snapshots.v1`).
`Accept-Encoding` (header, required)	`Accept-Encoding`	IDENTICAL	Amberdata rejects requests without this header; Kaiko supports compression (e.g. `curl --compressed`), recommended for deep books.
—	`page_size`	NEW	Snapshots per page. Default `10`, maximum `100`.
—	`continuation_token`	NEW	Pagination cursor. Already embedded in `next_url`, which also carries your time range, sort and page size; you rarely set it yourself.

4 Response envelope & metadata

The fields around the data, the wrapper your parser reads before reaching the book. This envelope is **identical on every Amberdata endpoint**.

Amberdata field	Kaiko field	Change	Notes
`status`	`result`	RENAMED	Integer `200` on Amberdata; string `"success"` on Kaiko. 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** of snapshots (see §5), now at the top level.

Amberdata field	Kaiko field	Change	Notes
`payload.metadata.next`	`next_url`	RENAMED	Full URL of the next page. Call it as-is until it is null or absent.
`payload.metadata.requestedStartDate` / `requestedEndDate`	—	REMOVED	Keep your own request range if you need it.
`payload.metadata.returnedStartDate` / `returnedEndDate`	—	REMOVED	Read `poll_timestamp` on the first and last rows instead.
`payload.metadata.api-version`	—	REMOVED	Version lives in the URL path on Kaiko; `query.data_version` echoes the data version (`v1`).
—	`continuation_token`	NEW	Cursor for the next page, also embedded in `next_url` .
—	`query`	NEW	Echo of the request: `exchange` , `instrument_class` , `instrument` , `limit_orders` , `sort` , `page_size` , `request_time` , etc.
—	`time` / `timestamp`	NEW	When the **response** was produced (ISO 8601 / Unix ms). Not a snapshot time: do not confuse with `data[].poll_timestamp` or `data[].timestamp` .
—	`access`	NEW	`access_range` and `data_range` , as Unix ms start/end. On this endpoint they **do not reflect the 30-day REST window**; do not use them to decide how far back to query.

5 Response - data fields

One row per snapshot. Each row holds two arrays of price levels, best price first.

Amberdata field	Kaiko field	Change	Notes
`exchange`	`query.exchange`	RENAMED	Not repeated on each row; echoed once in `query` . Kaiko exchange code.

Amberdata field	Kaiko field	Change	Notes
`instrument`	`query.instrument`	RENAMED	Not repeated on each row; echoed once in `query`. Kaiko code format (see §6).
`timestamp`	`poll_timestamp`	RENAMED	When the book was sampled. Amberdata values fall on the minute; Kaiko values are the actual poll time (e.g. `14:00:10.237`). `poll_date` gives the same time in ISO 8601.
`exchangeTimestamp`	`timestamp`	RENAMED	Time supplied by the exchange. **Same name, different meaning** from Amberdata's `timestamp`. Can be null; on some venues it equals `poll_timestamp`.
`exchangeTimestampNanoseconds`	—	REMOVED	Futures and options only on Amberdata. Kaiko times are in milliseconds.
`sequence`	—	REMOVED	No exchange sequence number on Kaiko snapshots. Order and de-duplicate by `poll_timestamp`.
`ask[]`	`asks[]`	RENAMED	Sell side, best (lowest) price first. Length varies (see §7).
`bid[]`	`bids[]`	RENAMED	Buy side, best (highest) price first. Length varies (see §7).
`[].price`	`[].price`	RENAMED	Same name; in the quote currency, returned as a decimal string.
`[].volume`	`[].amount`	RENAMED	Resting quantity at the level, in the base currency, returned as a decimal string.
`[].numOrders`	—	REMOVED	Number of orders at a level. Populated on Amberdata only for some venues (e.g. Bitfinex, Coinbase); not available on Kaiko and cannot be derived from `amount`.

Futures and options: contract fields

Amberdata's futures and options snapshots also carry contract data on each row. Kaiko keeps the snapshot to the book; request the same instrument from the **Ticker** endpoint, which also serves historical values, and join on time. Ticker takes `exchange`, `class` and `instrument` as query parameters (note `class`, not `instrument_class`), e.g. `/market-data/tickers?exchange=drbt&class=perpetual-future&instrument=btc-usd`. Ticker field names below are taken from sample Kaiko responses for Deribit futures and options.

Amberdata field	Kaiko Ticker field	Change	Notes
`exchangeTimestamp`	`exchange_timestamp`	RENAMED	Unix ms on Amberdata; ISO 8601 string on the Ticker.
`bestBidPrice` / `bestAskPrice`	`bid` / `ask`	RENAMED	Also the first element of the snapshot's `bids[]` / `asks[]`. Can be null.

Amberdata field	Kaiko Ticker field	Change	Notes
`bestBidVolume` / `bestAskVolume`	`bid_volume` / `ask_volume`	RENAMED	Can be null.
`markPrice`	`mark_price`	RENAMED	Futures and options.
`indexPrice`	`index_price`	RENAMED	Futures and options.
`lastPrice`	`last`	RENAMED	Last traded price.
`underlyingPrice`	`underlying_price`	RENAMED	Options.
`openInterest`	`open_interest`	RENAMED	Options.
`markIv` / `bidIv` / `askIv`	`mark_iv` / `bid_iv` / `ask_iv`	RENAMED	Options. **Check the scale:** Amberdata's sample shows a percentage (`40.59`), Kaiko's a fraction (`0.483` = 48.3%).
`delta` , `gamma` , `vega` , `theta` , `rho`	same names	IDENTICAL	Options. Returned as decimal strings.
`minPrice` / `maxPrice`	`min_price` / `max_price`	RENAMED	Options. Allowed price bounds.
`low` / `high`	`low_24h` / `high_24h`	RENAMED	Options.
`volume`	`base_volume`	RENAMED	Traded volume.
`volumeUSD`	—	REMOVED	No USD volume. The Ticker's `quote_volume` is in the quote currency, and can be null on options.
`state`	`state`	IDENTICAL	Options, e.g. `open` .
`currentFunding`	`funding_rate`	RENAMED	Perpetuals. **Not on the Ticker:** from the Derivatives risk endpoint (`/v2/data/derivatives.v2/risk`), one value per minute by default, which also returns `predicted_funding_rate` . Its `start_time` / `end_time` must include milliseconds.
`exchangeTimestampNanoseconds` , `underlyingIndex` , `priceChange` , `interestRate` , `estimatedDeliveryPrice`	—	REMOVED	Not in the Ticker response.
—	`mid` , `settlement_price` , `quote_volume` , `last_volume`	NEW	Mid price, settlement price, volume in quote currency, last trade size.

6 Exchange & instrument names

Kaiko identifies venues and instruments with its own normalized codes, which differ from Amberdata's names (e.g. `gdax` / `btc_usd` becomes `cbse` / `btc-usd`). Codes can also differ by market for the same venue.

> **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 **book values are the same**: compared at the same moment, Kaiko and Amberdata report the same prices and sizes. What differs is **when** each provider samples the book, **how deep** the returned book goes, and **how far back** you can query over REST.

The number of price levels varies

- **Depth is a price band, not a level count.** Kaiko returns the levels within ±10% of the best bid and ask. How many levels fit in that band depends on the venue, the instrument and the moment, so `bids[]` and `asks[]` lengths change from snapshot to snapshot and the two sides rarely have the same length.
- **Full-book requests return less.** In a Coinbase BTC/USD test, Amberdata without `maxLevel` returned 27,564 ask and 18,262 bid levels, reaching far from the mid price. Kaiko with `limit_orders=50000` returned 4,858 ask and 8,164 bid levels, stopping 9.99% from the best price on each side. Where the exchange itself publishes a shallow book, both return the same levels.
- **What to do.** Do not assume a fixed array length. If you read the top N levels, set `limit_orders=N`; Kaiko returns fewer only if fewer exist within the band. If you need levels beyond ±10% of the best price, contact support@kaiko.com before migrating.

History: a rolling 30-day window

- **About 30 days are available via REST**, and the window moves forward every hour. Amberdata serves several years through the same kind of request.
- **Out-of-window requests return nothing.** If `start_time` is even slightly older than the window, the response is `result: "success"` with an empty `data` array for the **whole** range, even if most of it is within the window. It is not an error and not a partial result. A request for "the last 31 days" therefore returns nothing.
- **What to do.** Keep `start_time` within the last 29 days, and treat an empty `data` array on an old range as "out of window", not as "no activity". Back-fill jobs that walk further back should stop at this limit.
- **Older history.** Kaiko holds order book snapshots well beyond 30 days. They are delivered as daily files through Kaiko Cloud Delivery; contact support@kaiko.com to enable it.

Other differences

- **Cadence and alignment.** Kaiko returns about 120 snapshots per hour (slightly more on some venues), Amberdata about 60, on the minute. The two schedules are not aligned: the nearest snapshots were 8 to 12 seconds apart in testing. Match snapshots by nearest time, not exact time; if your logic expects one snapshot per minute, resample.

- **Measured parity.** Across 21 spot venues, sizes at matching price levels had a median ratio of 1.00. For Coinbase BTC/USD snapshots taken within 2 seconds of each other, 225 of 226 had the same mid price, and the remaining one differed by one tick. Differences grow with the time gap between snapshots, as the book moves.
- **Exchange timestamp.** Kaiko `timestamp` was populated on all venues tested. On some venues it equals `poll_timestamp`; use `poll_timestamp` as your reference time.
- **Order count and sequence.** Logic that relies on `numOrders` or on `sequence` (e.g. gap detection) needs to be removed or reworked.
- **Number representation.** Kaiko returns decimal strings, which keep the exact digits; Amberdata returns JSON numbers. Compare as decimals, not as text.
- **Response time.** In 1,000 identical one-hour requests, Kaiko answered all of them, with a median response time of 187 ms.

8 Before / after example

Request

One hour of Coinbase BTC/USD snapshots, top 5 levels per side, oldest first.

Amberdata

```
GET https://api.amberdata.com/markets
    /spot/order-book-snapshots/btc_usd
    ?exchange=gdax
    &startDate=2026-11-10T14:00:00
    &endDate=2026-11-10T15:00:00
    &maxLevel=5
    &timeFormat=ms
x-api-key: <your key>
Accept-Encoding: gzip, deflate, br
```

Kaiko

```
GET https://us.market-api.kaiko.io
    /v2/data/order_book_snapshots.v1
    /exchanges/cbse/spot/btc-usd
    /snapshots/raw
    ?start_time=2026-11-10T14:00:00Z
    &end_time=2026-11-10T15:00:00Z
    &limit_orders=5
    &sort=asc
    &page_size=100
x-api-key: <your key>
```

Add `sort=asc` to keep Amberdata's oldest-first order, and keep `start_time` within the last 30 days. The hour returns about 120 Kaiko snapshots, on two pages at `page_size=100`.

Response

Documentation samples. The two calls cover different markets and dates, so values are not comparable.

Amberdata


```
{
  "status": 200,
  "title": "OK",
  "description": "Successful request",
  "payload": {
    "metadata": {
      "requestedStartDate": 1717513062160,
      "requestedEndDate": 1717599462160,
      "returnedStartDate": 1717513080000,
      "returnedEndDate": 1717515360000,
      "next": "https://api.amberdata.com
        /markets/spot/order-book-snapshots
        /eth_usd?cursor=...",
      "api-version": "2023-09-30"
    },
    "data": [
      {
        "exchange": "bitfinex",
        "instrument": "eth_usd",
        "timestamp": 1717513080000,
        "sequence": null,
        "ask": [
          { "price": 3780.7,
            "volume": 7.33,
            "numOrders": 1 },
          { "price": 3780.8,
            "volume": 11.60410568,
            "numOrders": 1 }
        ],
        "bid": [ ... ]
      },
      ...
    ]
  }
}
```


Kaiko


```
{
  "query": {
    "page_size": 10,
    "exchange": "krkn",
    "instrument_class": "spot",
    "instrument": "btc-usd",
    "limit_orders": 2,
    "sort": "desc",
    "metric": "raw",
    "data_version": "v1",
    "commodity": "order_book_snapshots"
  },
  "time": "2020-05-26T14:13:08.899Z",
  "timestamp": 1590502388899,
  "data": [
    {
      "poll_timestamp": 1590502335760,
      "poll_date": "2020-05-26T14:12:15.760Z",
      "timestamp": null,
      "asks": [
        { "amount": "12", "price": "8830" },
        { "amount": "3.67", "price": "8832.9" }
      ],
      "bids": [
        { "amount": "13.316", "price": "8829.9" },
        { "amount": "0.097", "price": "8829.4" }
      ]
    },
    ...
  ],
  "result": "success",
  "continuation_token": "...",
  "next_url": "https://us.market-api.kaiko.io
    /v2/.../snapshots/raw
    ?continuation_token=...",
  "access": {
    "access_range": { ... },
    "data_range": { ... }
  }
}
```