Kaiko

MIGRATION GUIDE

Funding Rates

HTTP REST

Amberdata → Kaiko

2026

www.kaiko.com

1 At a glance

What this endpoint gives you. Perpetual futures funding-rate data for a single exchange and instrument — the periodic rate paid between long and short positions — over a date range you choose.

What changes for you

- **Unlike Trades, this Kaiko endpoint keeps `exchange`, `instrument_class` and `instrument` as query parameters**, not URL path segments. If you've already built URL-construction logic for Spot/Futures/Options Trades, you cannot reuse it here (see §2 and §3).
- **A regularly sampled time series.** Kaiko returns one row per `interval` (`1m`, `1h` or `1d`) carrying the latest known funding rate, rather than Amberdata's discrete funding-rate events (see §3 and §7).
- **Funding rates come from Kaiko's Derivatives Risk Metrics endpoint**, which also returns `24h_volume` and `open_interest` on each row (see §5).
- `is_actual_funding_rate` **changes meaning and type.** On Amberdata it told you whether a funding payment had actually happened; on Kaiko it tells you whether the rate on the row is the current one rather than a stale one. It is also a **string** (`"true"` / `"false"`), not a boolean. Existing logic will return the wrong series without any error (see §5).
- **No `rateType` filter.** Both rates come as columns on every row: for Amberdata `rateType=applied`, read `funding_rate` and keep one row per `funding_rate_ts` (see §3).

Coverage & distribution

Data type: Perpetual futures funding rate (plus open interest and 24h volume, Kaiko-only)

Source: One CEX derivatives exchange + instrument per request

Granularity: Amberdata: discrete funding-rate events. Kaiko: time series sampled at `1m`, `1h` or `1d` (see §7)

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`
Path	`/markets/futures/funding-rates/{instrument}`	`/v2/data/derivatives.v2/risk`
Auth	API key header `x-api-key`	API key header `X-Api-Key` (same mechanism, new key)

Kaiko API family: Classic (versioned)

Funding rates are served by Kaiko's global **Derivatives Risk Metrics** endpoint, part of 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
`exchange` (query, required; only 1 allowed)	`exchange` (query, required)	SAME POSITION	Stays a query parameter on both sides — unlike Trades, where `exchange` moves to the URL path. Values still change (see §6).
`{instrument}` (path, required)	`instrument` (query, required)	RENAMED	**Structural flip, opposite of Trades:** Amberdata keeps it in the URL path; Kaiko makes it a query parameter instead.
—	`instrument_class` (query, required)	NEW	**No Amberdata equivalent.** Same concept as Trades' `instrument_class`, but here it's a query parameter, not a path segment.
—	`interval` (query)	NEW	**No Amberdata equivalent.** Sampling step of the returned series: `1m`, `1h` or `1d` (see §7).
`startDate` (optional)	`start_time` (optional)	RENAMED	Inclusive lower bound on both sides. Amberdata: max 731-day range, defaults to the prior 24 hours if omitted. **Kaiko: ISO 8601 with milliseconds required**, e.g. `2026-10-01T00:00:00.000Z`; a value without `.000Z` is rejected.
`endDate` (optional)	`end_time` (optional)	RENAMED	Exclusive upper bound on both sides. Same millisecond format required on Kaiko (`.000Z`).
`timeFormat` (optional)	—	REMOVED	No alternate timestamp format on Kaiko.
`sortDirection` (optional)	`sort` (optional)	RENAMED	Same `asc` / `desc` values; Kaiko defaults to `desc` .
`rateType` (optional: applied / predicted / both)	—	REMOVED	No filter on Kaiko: every row carries both `funding_rate` and `predicted_funding_rate` . Migration path: `applied` → read `funding_rate` and keep **one row per** `funding_rate_ts` (rows are sampled per interval and repeat the same settled rate); `predicted` → read `predicted_funding_rate` ; `both` → read both columns.
—	`page_size` (optional)	NEW	Pagination control; no Amberdata equivalent param for this endpoint.

4 Response envelope & metadata

The Amberdata envelope is identical to the other endpoints in this kit. Kaiko's envelope here is **simpler than Trades'**: no top-level `result` field.

Amberdata field	Kaiko field	Change	Notes
`status` / `title` / `description`	—	REMOVED	No status fields in the Kaiko body.
`payload.data`	`data`	RENAMED	**Array** holding the records (see §5), now at the top level.
`payload.metadata.api-version`	`query.data_version`	RENAMED	Value `v4` observed here, vs `v2` on the Trades endpoints — different commodity, not necessarily meaningful beyond that.
—	`time`, `timestamp`	NEW	Server response time. No Amberdata equivalent.

5 Response - data fields

Amberdata field	Kaiko field	Type	Change	Notes
`instrument`	`query.instrument`	string	REMOVED	Not repeated per record, but recoverable from the echoed `query.instrument`.
`exchange`	`query.exchange`	string	REMOVED	Not repeated per record, but recoverable from the echoed `query.exchange`.
`exchangeTimestamp` + `exchangeTimestampNanoseconds`	`funding_rate_ts`	string +int → ISO 8601	RENAMED	**Careful — not the same field as `timestamp`.** Kaiko's per-row `timestamp` is the sampling time (one `interval` apart); `funding_rate_ts` stays fixed across rows until the next funding event, which is what Amberdata's `exchangeTimestamp` captures.
—	`timestamp`	integer (unix ms)	NEW	Sampling time of the row (see §7), not a funding-event time. No direct Amberdata equivalent.
`fundingInterval`	`funding_interval`	number → string	RENAMED	**Populated on Kaiko only.** Amberdata documents a number but returns `null` in practice (checked on Deribit and OKX). Kaiko returns the time between funding settlements as a string: `1h`, `4h`, `8h` ... or `continuous`.
`fundingRate`	`funding_rate`	number → string	RENAMED	**Adaptation required.** Same type-change pattern as other endpoints: number → string.

Amberdata field	Kaiko field	Type	Change	Notes
`nextFundingRate`	`predicted_funding_rate`	number (nullable)	RENAMED	**Venue-dependent, and usable where populated.** `null` on Binance `btc-usd` (sample in §8), but populated and independent of the settled rate on other venues: on Bybit `btc-usdt`, all 120 hourly rows checked differ from `funding_rate`.
`nextFundingTime`	`next_funding_time`	string → ISO 8601	RENAMED	Amberdata's human-readable `"2026-10-07 16:00:00 000"` format becomes ISO 8601 (`"2026-10-08T16:00:00Z"`) on Kaiko.
`isActualFundingRate`	`is_actual_funding_rate`	boolean → string	RENAMED	**Meaning changes:** Amberdata = a funding payment actually happened; Kaiko = the rate on this row is current (set at the last settlement, before the next one) rather than stale. Do not reuse Amberdata logic as is. **Type changes too — stringified boolean.** Kaiko returns the literal string `"true"` / `"false"`, not a JSON boolean — different from the real booleans you'll have seen on the Trades endpoints (e.g. `taker_side_sell`). A naive `if (value)` check will be truthy even for the string `"false"` in some languages — parse explicitly.
—	`24h_volume`	string	NEW	No Amberdata equivalent on this endpoint.
—	`open_interest`	string	NEW	No Amberdata equivalent on this endpoint.

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?

A different data model, not just a different encoding

Amberdata returns discrete funding-rate events: its `exchangeTimestamp` values are a few minutes apart. Kaiko returns a series sampled every `interval` (`1m` , `1h` or `1d`): each row repeats the latest known funding rate, `funding_rate_ts` stays constant until the next funding event, and `timestamp` advances by one interval per row.

Short answer, for the shared fields

Where the two sides describe the same funding event, the rate values themselves should match. The bigger risks here are structural: a type that silently looks truthy (`is_actual_funding_rate` as a string), and a per-row `timestamp` that is the sampling time, not the funding event time (§5).

- **URL construction is not reusable from Trades.** `exchange` / `instrument_class` / `instrument` are query parameters here, path segments there (§2, §3).
- `is_actual_funding_rate` **: new meaning and new type.** It flags a current rather than stale rate, not a payment event, and it is a string. Rewrite the logic and parse it explicitly; keeping Amberdata's logic yields the wrong series silently.
- **Don't confuse** `timestamp` **with the funding event time.** Use `funding_rate_ts` for "when was this rate set", not the per-row `timestamp` (§5).
- **Replacing** `rateType=applied` **.** Read `funding_rate` and keep one row per `funding_rate_ts` .
- **Predicted rate.** `predicted_funding_rate` is populated on some venues (e.g. Bybit) and `null` on others (e.g. Binance `btc-usd`).
- **Two bonus fields.** `24h_volume` and `open_interest` come with every row; Amberdata has no counterpart on this endpoint.

8 Before / after example

Request (real)

Amberdata

```
GET /markets/futures/funding-rates/BTCUSD_PERP
    ?exchange=binance
x-api-key: <your-key>
```

Kaiko

```
GET /v2/data/derivatives.v2/risk
    ?exchange=binc
    &instrument_class=perpetual-future
    &instrument=btc-usd
    &sort=desc
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": "BTCUSD_PERP",
        "exchange": "binance",
        "exchangeTimestamp": "2026-10-07 12:24:01
002",
        "exchangeTimestampNanoseconds": 0,
        "fundingInterval": null,
        "fundingRate": -0.00001147,
        "nextFundingRate": null,
        "nextFundingTime": "2026-10-07 16:00:00
000",
        "isActualFundingRate": false
      },
      {
        "instrument": "BTCUSD_PERP",
        "exchange": "binance",
        "exchangeTimestamp": "2026-10-07 12:25:01
000",
        "exchangeTimestampNanoseconds": 0,
        "fundingInterval": null,
        "fundingRate": -0.00001166,
        "nextFundingRate": null,
        "nextFundingTime": "2026-10-07 16:00:00
000",
        "isActualFundingRate": false
      }
    ]
  }
}
```


Kaiko


```
{
  "query": {
    "page_size": "100",
    "exchange": "binc",
    "instrument_class": "perpetual-future",
    "instrument": "btc-usd",
    "interval": "1m",
    "sort": "desc",
    "data_version": "v4",
    "commodity": "derivatives",
    "request_time": "2026-10-08T12:20:54.852Z"
  },
  "time": "2026-10-08T12:20:54.877Z",
  "timestamp": 1791462054877,
  "data": [
    {
      "timestamp": 1791462000000,
      "24h_volume": "7632820",
      "open_interest": "12966315",
      "funding_rate": "0.00007265",
      "predicted_funding_rate": null,
      "funding_rate_ts": "2026-10-08T08:00:00Z",
      "next_funding_time": "2026-10-08T16:00:00Z",
      "is_actual_funding_rate": "true",
      "funding_interval": "8h"
    },
    {
      "timestamp": 1791461940000,
      "24h_volume": "7632537",
      "open_interest": "12966300",
      "funding_rate": "0.00007265",
      "predicted_funding_rate": null,
      "funding_rate_ts": "2026-10-08T08:00:00Z",
      "next_funding_time": "2026-10-08T16:00:00Z",
      "is_actual_funding_rate": "true",
      "funding_interval": "8h"
    }
  ]
}
```


Two consecutive rows on each side: Kaiko's `timestamp` advances by `60000` ms (the `1m` interval) while `funding_rate_ts` stays the same, which illustrates the sampled series described in §7.

Questions? Kaiko support · support@kaiko.com

docs.kaiko.com