LIKEFOLIO DATA · REFERENCE

Data dictionary

V1 · SCHEMA 2026-09

Conventions

- **IDs.** Key your data to `company_id`, it's the one id that never changes, even when a company is renamed, re-lists, or is acquired, so your backtests don't break. Don't key on the `ticker`; tickers change over time (look one up with `/resolve`). `entity_id` identifies a single division or brand inside a company, larger companies have several, and a company's number is simply the sum of its divisions. Brand-level detail (subscription) rolls up the same way.
- **Date.** `date` is an **America/New_York calendar day** (ISO `YYYY-MM-DD`).
- **Kdate.** The knowledge date. `kdate=YYYY-MM-DD` returns the data as it would have been seen on that day, including mergers, acquisitions and brand-ownership changes, for company data (X and Google) and for the Google-search trends. X is recorded from launch and Google search from 2026-11-04; earlier history on both is reconstructed. X trend series are served as published and take no `kdate`.
- **Shape.** Tidy-long: one row per key × date × metric.
- **Values.** Counts are whole numbers of posts.
- **Encoding.** UTF-8 CSV with header, or JSON from the API. `format=parquet` and `format=arrow` return a time-limited download URL for bulk pulls, available on trial too, through the rolling six-month lag.
- **Provenance.** Only derived counts are shipped: no post text, ids, user objects, or third-party raw data.

Coverage

One row per entity. An entity is one company's consumer business in one category (Apple's devices, Match Group's dating apps). 651 entities across 497 tickers, built from 4,579 hand-curated brand and product aliases.

COLUMN	TYPE	MEANING
`entity_id`	string	Per-division / entity key. Example: `AAPL#consumer-electronics-devices`.
`company_id`	string	LikeFolio company id (tied to Refinitiv/LSEG permID), the stable primary key, unchanged across ticker renames, re-listings and M&A. Key everything by this. Resolve a ticker or permID to it via `/resolve`.
`perm_id`	string	This company's Refinitiv/LSEG PermID (a permanent cross-vendor id). Blank if permid.org hasn't issued one for this entity.

`figi`	string	The company's FIGI, Bloomberg's open Financial Instrument Global Identifier (e.g. `BBG000B9XRY4`), checked against OpenFIGI. Blank if none is known. `share_class_figi` is also available and resolvable via `/resolve`.
`ticker`	string	Current listed symbol, a mutable, point-in-time alias (join to your own price and fundamentals as-of a date), not a key. The universe carries dated ticker history; delisted symbols keep their last symbol.
`company`	string	Company name.
`division`	string	Consumer category the entity rolls up to.
`brand_label`	string	Set when the entity is a single acquired brand carved into its own division.
`valid_from` / `valid_to`	date	Ownership window. Blank `valid_from` means it was owned since the start of the data; blank `valid_to` means it is still currently owned. Set on 128 entities (acquired, divested, renamed).
`is_primary`	bool	True for the main entity of a category; false for a carved-out sub-brand.
`coverage_status`	enum	The registry's present status for the entity: `covered`, `delisted` or `acquired`. Count unique `company_id` by this to separate live coverage from history.
`coverage_start` / `coverage_end`	date	First and last covered day; `coverage_end` equals `valid_to` (blank while still covered).
`ticker_recorded_from`	date	First day of the company's dated ticker history.
`source_ticker`	string	The ticker as recorded in the source registry (compare `ticker`, the current listed symbol).
`core`	bool	False for non-core entities (excluded from ticker roll-ups).

Company metric series

Daily counts of English public posts on X naming the entity's brands, that also demonstrate that metric with natural language. Retweets, promotional, and spam posts are excluded. History from 2021-01-01. Refreshed daily. Served by `GET /company/{companyId}/metrics`, every signal is a metric you pull with `metric=`; there is no separate "demand" endpoint. Key rows by `company_id`, not by the mutable `ticker` (resolve a ticker via `/resolve`).

COLUMN	TYPE	MEANING

`entity_id`	string	The division this row belongs to. Example: `AAPL#consumer-electronics-devices`.
`company_id`	string	LikeFolio company id (tied to Refinitiv/LSEG permID), the stable key, unchanged across ticker renames, re-listings and M&A. Key everything by this. Resolve a ticker or permID to it via `/resolve`.
`ticker` / `ticker_at_date`	string	`ticker` is the company's current listed symbol (a mutable alias, not a key); `ticker_at_date` is the symbol on the row's own date from the dated ticker history (null where that history does not reach).
`division`	string	Human division label; present on `rollup=entity` rows.
`coverage_status`	enum	As in the coverage table: `covered`, `delisted` or `acquired`.
`source_ticker`	string	The ticker as recorded in the source registry.
`date`	date	Eastern calendar day (the event date; bound with `start` / `end`).
`kdate`	parameter	Knowledge date. `kdate=YYYY-MM-DD` returns the series as it would have been seen on that day.
`metric`	enum	One of the 16 metrics below.
`raw`	int	Posts counted that day, as pulled.
`corrected`	int	Raw with the chosen `correct=` categories replaced by a cleaned count, as whole numbers. Equal to raw where nothing was corrected, a blank flag, a `demand_event` day, or a category you excluded. **Use this column by default.**
`flag`	enum	Blank, or the editorial verdict on a reviewed spike: one of `meme`, `incident`, `marketing`, `investor`, `demand_event`. By default every category except `demand_event` is corrected (replaced in the `corrected` column); `demand_event` is kept as genuine demand. You choose the set with the `correct=` parameter; the flag tells you which category each day belongs to. Bots and spam are removed upstream and never appear as a flag.
`review_pending`	bool	True while a spike that affects this day is still under review (its category not yet settled).
`flagged_entities`	int	`rollup=ticker` rows only (in place of `entity_id` / `division`): how many of the company's entities carry a flag that day; `flag` is then the union of the entities' flags.
`kdate_from` / `kdate_to`	date	Bitemporal window (present when `kdate=all`): the knowledge dates over which this value was current. `kdate_to` null = still current.
`change_reason`	enum	Why this row supersedes the prior value (present when `kdate=all`): `as-pulled`, `spike-reclass`, `repull`, `negation`, `methodology`, `mna`.

`ownership_kdate` / `release_id`	date / string	Present when `kdate=all`: the ownership-map vintage applied to the row, and the publication release it came from.

The `date` axis is the event day. The separate **knowledge-date** axis is set with the `kdate` query param: it returns values as LikeFolio knew them on that Eastern day (no look-ahead through restatements or M&A), and `kdate=all` exposes the full bitemporal history via the three columns above.

Metrics

METRIC	WHAT IT COUNTS	TICKERS WITH MEANINGFUL DATA*
`mentions`	Every post naming the brand or its products. The headline demand series.	482
`pi`	Purchase intent: bought, buying, ordered, want, need, signed up.	335
`ps` / `ns`	Positive / negative sentiment. Share positive = ps / (ps + ns).	401 / 294
`npi`	Negative purchase intent: cancelling, returning, switching away, never again.	137
`switch_in` / `switch_out`	Switching to / away from the brand.	13 / 17
`promo`	Deals, coupons, sales, discount language. The "is demand bought or organic" flag.	246
`political`	Boycott and political framing. Read as a rate over mentions.	266
`price_value` / `price_pain`	"Worth it" vs "too expensive".	161 / 141
`defect`	Product problems, broke, doesn't work.	155
`outage`	Service down, can't log in.	156

`safety`	Recalls, injuries, contamination.	116
`labor`	Strikes, layoffs, unions.	72
`availability`	Out of stock, sold out, back in stock.	61

*Covered = averages **≥25 posts/week** over the trailing year, **or** spikes to **≥20% of that day's mentions** on at least one day (≥10 posts that day), so a metric that's quiet most days but carries real signal in bursts still counts. Switching and availability are genuinely thin, meaningful on only a few dozen names.

Mentions and purchase intent are the primary metrics. The secondary metrics are category-scoped and sparse by design: outage is a software and telecom signal, availability a retail one, safety an autos and food one, etc. Some are most useful on spikes only.

Consumer trends

141 consumer behaviors ("cutting back on spending", "taking a road trip", "cancelling a streaming service"), each a hand-built phrase rule counting people describing *doing* the behavior. Daily since 2016-09-01, refreshed daily, spike-checked and corrected like the company series.

TABLE	COLUMNS	NOTES
`trend_catalog`	`trend_id`, `name`, `group`, `mapped_tickers`, `mapped_datasets`	Names and the trend→company and trend→dataset maps are available at every tier. `group` is one of 15 themes (Health & Fitness, Eating & Drinking, Investing/Crypto/Gambling, ...); `mapped_tickers` / `mapped_datasets` count the companies and public-data series each trend is mapped to.
`trend_series`	`trend_id`, `name`, `date`, `raw`, `corrected`, `flag`	The published daily series, with the same `raw` / `corrected` / `flag` semantics as the company metrics. Trend series are served as published, the `kdate` knowledge-date axis applies to company metrics only.
`trend_ticker_map`	`trend_id`, `name`, `ticker`, `weight`, `added_at`, `removed_at`	Which companies a trend should move, and how strongly: `weight` is signed, positive = long, negative = short, in bands of 1.0, 0.6, 0.3 or 0.1. Dated membership so backtests use the map as it stood.
`trend_dataset_map`	`trend_id`, `name`, `series`, `desc`, `freq`, `sign`, `primary`	The public series each trend is checked against (Census retail sales, PCE, Fed credit data, BLS, TSA...): `series` is the series id, `desc` its plain name, `freq` its cadence, `sign` the expected direction, `primary` the main benchmark for that trend.

`search_trends`	catalog: `trend_id`, `name`, `members`, `first_date`, `last_date`, `resolution` · series: `trend_id`, `name`, `date`, `interest`	The Google-search version of a trend (141 served): `members` is how many search terms it sums, `first_date` / `last_date` its span, and every series response carries `complete_through`, the trend's last complete day. Takes `kdate`: recorded from 2026-11-04, earlier reconstructed.

Search interest

Brand-level Google search interest, daily, US, from 2020-09-01. Google scores each request 0-100 in isolation; tie all scores together so all brands sit on one global scale where the smallest covered brand averages roughly 50 and large brands read in the thousands. Any two brands on any two dates are always comparable. Refreshed daily. We publish **3,486** brands across **473** companies, each verified against Google's own series before it ships; brands too small or defunct to register are left out rather than published as flat-zero noise. Search-interest series are also served for **139** of the consumer trends. A brand catalog (`/v1/search/brands`) lists every published brand per company with its `brand_id`, whether it is `verified`, and its `first_date` / `last_date`. A `kdate` read adds `history_mode` (`recorded` or `reconstructed`), `ownership_kdate`, `kdate_from` / `kdate_to` and `recorded_from` to the response envelope.

COLUMN	TYPE	MEANING
`brand`	string	The brand's display name on subscription; on trial a stable code (`TICKER#b1`..., ordered by volume) with the name withheld. The name is a label, not the underlying Google query or Topic.
`ticker`	string	Owner on that date (point-in-time).
`date`	date	Calendar day.
`kdate`	parameter	Knowledge date. `kdate=YYYY-MM-DD` returns the data from the brands the company owned on that day, with the series through that day. Recorded snapshots from 2026-11-04; earlier dates are reconstructed from the first snapshot.
`interest`	number	Unbounded global index, comparable across brands and dates.
`resolution`	enum	`daily` or `weekly`: whether the row is built from Google's daily or weekly values (Google reports some brands only week by week).
`valid_from` / `valid_to`	date	The window in which this brand belonged to the ticker; no row falls outside it.
`brands`	int	`rollup=ticker` rows only: how many brands had a value that day (the company line is their sum).

What differs between trial and subscription

FIELD	TRIAL	SUBSCRIPTION
`brand_label`	withheld	included
`brand` (search)	`TICKER#bN`	brand name
Latest date	6-month lag (rolling), at every `kdate`	latest refresh
Values	identical	

Trial ids are stable across refreshes, so a model built on the trial transfers to the subscription by a one-time id join we supply at signing.