

Table 2 Code Guide

Overview

The script connects to the **EOD Historical Data** service and pulls only a tiny slice of their *Fundamentals* feed:

```
/fundamentals/{TICKER}?filter=Earnings::Trend&api_token=...
```

- The `filter=Earnings::Trend` parameter tells EOD to return *only* the **Trend** block (not the full 2–3 MB fundamentals file).
- Inside that block, each record has fields such as `epsTrendCurrent`, `epsTrend7daysAgo`, ... plus a period label (`0q`, `+1q`, `0y`, `+1y`...).

Main Code Components

1. `_get_json(url)`
Small helper that performs an HTTP GET, checks the status code, and returns decoded JSON.
2. `_pull_trend(ticker)`
Calls the endpoint above. Two possible shapes are handled:
 - *Filtered* call (our default): JSON *is* the **Trend** object.
 - *Unfiltered* call: JSON is the full fundamentals file, so the function digs down to **Earnings** → **Trend**.
3. `_latest_snapshots(trend)`
The API may contain multiple dated snapshots for each period code. This function keeps only the *newest* snapshot per period (e.g. the most recent `0q`, `+1q`, ...). **N.B.**: the edge case seen on the TICKER **AFRM**, was due to the wrong labeling by EOD, they labeled it `0q` instead of `+1q`.
4. `_get_period(rec_map, wanted)`
Returns the requested period snapshot. If a company is missing `+1q`, the function automatically falls back to the *nearest future quarter* that exists (`+2q`, `+3q`, ...). The same logic applies to years.
5. `_row_for(ticker)`
Orchestrates the previous helpers and constructs one tidy table row with the 20 columns:
$$(Curr\ Qtr\ Curr, \ Curr\ Qtr\ 7, \ 30, \ 60, \ 90), \dots (Next\ Yr\ 90).$$
6. `build_table(ticker_list)`
Loops through all tickers, pauses ≈ 1 second between calls to respect rate limits, and concatenates the rows into a **pandas** data-frame.
7. `save(df, "eps_trend.xlsx")`
Writes the final table:
 - `.xlsx` (uses `openpyxl`) or `csv` (UTF-8).Choosing CSV avoids Excel-library dependencies if desired.

Result

Each ticker ends up on *one* line, ready for side-by-side comparison of how analysts' EPS estimates have moved over the last 7, 30, 60 and 90 days for the:

- current quarter,
- next quarter,
- current fiscal year and
- next fiscal year.