A new CSV column should not silently change your data

2026-11-27 · 8plus

On November 23, 2026, GitHub announced that selected usage reports containing username will gain an adjacent user_id column on December 1, shifting subsequent columns. Summarized reports are unchanged. GitHub recommends reading by header name rather than position. This article precedes the change; check the announcement for applicability.

A successful import can read the wrong thing

This is engineering guidance from 8plus, not an incident report. Suppose an importer treats the fourth column as usage. Inserting a column earlier in the file may leave the program running normally. If the new value also parses as a number, even conversion will not reveal the mistake. A successful job does not establish correct meaning.

Find every consumer of the export: production importers, spreadsheet formulas, one-off analyses, and scripts generated by AI. Updating the main service is insufficient if a weekly report still relies on position. Record which fields each consumer actually needs.

Header mapping still needs boundaries

Use a CSV parser that handles quoting and embedded newlines, then map headers to values. Splitting text on commas or changing one numeric index to another does not solve the underlying problem. Missing required fields should fail explicitly. Establish a consistent policy for extra fields.

Consider duplicate names, empty headers, and letter case. If you normalize names, define the rule and reject collisions after normalization. Excessively permissive matching can collapse distinct fields. Guessing is not a durable compatibility strategy.

Accept the change with reordered fixtures

Create a small synthetic fixture without real personal data. Keep the original headers, then produce variants with an inserted field, reordered fields, and a missing required field. The first two should preserve existing field values; the last should stop and explain what is missing. These counterexamples test the contract more directly than rerunning one familiar file.

Do not automatically turn an empty identifier into zero or a shared placeholder. Establish what absence means at the source before preserving, quarantining, or skipping it. Adding identity-based analysis also requires a separate review of access and retention needs; a new field need not be forwarded to every downstream system.

Preserve inputs separately from derived results

After updating the importer, compare results with controlled fixtures and record the code version and input format. Where needed, retain restricted original inputs so results can be recomputed. Log the information required for diagnosis rather than complete data rows.

The acceptance question is whether a changed column order still yields the same kind of data. Put that question into tests and delivery records so a small schema addition does not become a long-lived, silent error.

Sources

Official announcement

Related reading