

Google Sheets

From the SourceLace manual: How to connect Google Sheets, so people can read a sheet by pasting its link and, when an admin allows it, add rows and change cells after confirming a preview.

SourceLace version 0.1.0 · Made October 4, 2026

The latest version is at <https://sourcelace.onrender.com/docs>

Google Sheets

How to connect Google Sheets, so people can read a sheet by pasting its link and, when an admin allows it, add rows and change cells after confirming a preview.

Status: in testing. Built against Google's documented Sheets API and covered by automated tests against a simulated Google; not yet used with a customer's live sheets.

Source	Kind	Reads	Changes	People sign in with
Google Sheets	google_sheets	The tabs of one sheet, named by its link or id	Add a row, update cells of a row, after the person confirms (off unless an admin turns it on). No deletes.	Their own Google account

How it works

- **Each person signs in with their own Google account.** Google applies that person's own sharing: they read only sheets shared with them, and change only sheets they can edit.
- **A separate source from Google Drive.** A Google Drive source asks only for read-only Drive access and stays read-only. A Google Sheets source asks only for `https://www.googleapis.com/auth/spreadsheets`, plus `openid` and `email` so SourceLace knows who signed in for the audit trail. Connecting one never grants the other.
- **Sheets are named by link.** The Sheets permission cannot list or search files, so people paste the sheet's link (or its id). An admin can also list sheets the team uses often; their tabs then show in `search_schema`.
- **Nothing is stored.** Rows are read when asked and held in memory like any other result (up to 30 minutes), never written to a database.

Set up

SourceLace's Google app (whoever runs this SourceLace server)

Google Sheets signs in through SourceLace's own Google app, the same one as Gmail, Google Calendar and Google Drive, so there is no new app or redirect URL. In the Google Cloud project that holds that app, once:

1. **APIs & Services → Library:** search for **Google Sheets API** and click **Enable**.
2. **Google Auth Platform → Data Access** (or **OAuth consent screen → Scopes**): **Add or remove scopes**, then under **Manually add scopes** paste `https://www.googleapis.com/auth/spreadsheets`, click **Add to table**, tick it, then **Update** and **Save**. `openid` and `.../auth/userinfo.email` are usually there already; tick them if not.

`spreadsheets` is one of Google's "sensitive" scopes: while the app's consent screen is in testing, only its test users can connect. Opening it to everyone needs Google's verification.

Your Google Workspace admin may need to allow SourceLace's app under **Google Admin console → Security → Access and data control → API controls → App access control**, if your organization restricts third-party apps.

Add the source (SourceLace admin)

Kind: **Google Sheets** (google_sheets).

Option	Type	Default	Example	What it is
spreadsheets	List, separated by commas	(none)	https://docs.google.com/spreadsheets/d/1AbC.../edit	Optional. Links or ids of sheets your team uses often, at most 20. Their tabs show in search_schema. People can still use any other sheet shared with them by pasting its link.

To allow changes, turn on **Allow changes** and put * under **Objects that can be changed** (every sheet people can edit), or name tabs as <sheet id>/<tab name>. The sign-in already covers changes, so nobody needs to connect again; Google still refuses changes to sheets the person cannot edit.

What people can do

An object is one tab, written <sheet id>/<tab name>. Anywhere a sheet is named, its full link (<https://docs.google.com/spreadsheets/d/<id>/edit#gid=0>) or its bare id works.

- `describe_object` on a sheet's link lists its tabs. On <id>/<tab name> it lists the tab's columns (from row 1) and the size of its grid. A link with #gid=... picks that tab.
- `query` (language sheets) reads a tab. It takes one JSON object, such as {"sheet": "https://docs.google.com/spreadsheets/d/1AbC.../edit", "tab": "Pipeline", "range": "A1:H500"}. Only sheet is required. Without tab, SourceLace reads the tab the link points at, or the first tab; without range, the whole tab (up to the row limit). With header (the default, true), row 1 of the range names the columns, and an empty header cell is named by its column letter. `limit` returns at most that many rows. Every row also has `_row`, its row number in the sheet. Empty rows are skipped. Values come back as the sheet shows them (dates as dates, currency with its symbol, formula results).
- `get_record` with object <id>/<tab name> and a row number returns that row.

Changes

Through `propose_change` and `apply_change`, like every other source: the person sees a preview, and nothing is written until they confirm.

- **Add a row** (create): adds one row after the table on that tab. Fields are named by the column names in row 1 (any capitalization); columns not named stay empty. Names that are not in row 1 are listed as problems in the preview.
- **Update cells** (update): changes some cells of one row. The record id is the row number (the `_row` value, 2 or more). The preview shows each cell's current value, and a null value empties the cell. Just before writing, SourceLace reads the row and row 1 again; if either changed since the preview (for example someone inserted a row above, so row 12 is now a different row),

the change is refused and must be proposed again.

- **Deleting rows is not offered:** deleting a row changes the number of every row below it. To empty a row, update its cells to `null`.

Values are always written **RAW**: Google stores exactly what was given and never treats it as a formula. Text such as `=IMPORTXML(...)` or `+1 555 0100` stays plain text.

SourceLace only ever reads a sheet's title and tabs, reads cells, appends a row and sets cells. It has no call that deletes rows or tabs, clears ranges, or changes a sheet's structure or sharing.

When something goes wrong

What you see	What to do
"Google did not grant access to Google Sheets. Connect again and tick the box on Google's consent screen."	Connect again and tick the Sheets box.
"Google Sheets: that sheet was not found, or it is not shared with you."	Check the link; ask the sheet's owner to share it with you.
"Google Sheets: you do not have access to this sheet. Ask its owner to share it with you."	Ask the owner to share it.
"Google Sheets: you do not have edit access to this sheet. Ask its owner to share it with you as an editor."	Changes need edit access in Google.
"This sheet has no tab named '...'. Its tabs: ..."	Use one of the tab names listed.
"Google Sheets: that tab or range does not exist. describe_object on the sheet lists its tabs."	Check the tab name and range.
"Row 1 has no column named '...'. Use describe_object to see the column names."	Name fields exactly as row 1 does.
"Google Sheets: row 1 of ... is empty. SourceLace needs column names in row 1 to change rows."	Put column names in row 1 of the tab.
"Row ... of ... changed after the preview (or rows moved). Propose the change again."	Someone changed the sheet; ask for a new preview.
"Google Sheets rows cannot be deleted through SourceLace ..."	Update the row's cells to <code>null</code> instead.
"Google Sheets is limiting how fast SourceLace can call it. Wait a minute and try again."	Wait, then try again.
"The spreadsheets option of ... lists at most 20 sheets."	Shorten the spreadsheets option.
"The google_sheets connector is switched off on this server..."	SourceLace's Google app is not set up on this server yet. Contact support@sourcelace.com .