CASE STUDY · NO-CODE AUTOMATION · MAKE (INTEGROMAT)

No-Code Lead Pipeline in Make:

Automated Email Dispatch + Smart CRM Sync via Webhook

B2B / B2C · Lead Automation · January–February 2026

Two Make scenarios built, tested, and deployed with zero errors

2	0	200	2s / 1s
Make (Integromat)	Google Sheets · Gmail	Webhook · Router	Postman · API

The business problem this solves

Small and mid-size businesses running email outreach face two operational bottlenecks that grow with scale: manually filtering which leads should receive an email based on their status, and keeping a CRM-style record current when new contact data arrives from external sources. Both tasks are repetitive, error-prone when done by hand, and consume admin time that should go toward revenue-generating work.

The Data Source

A Google Sheet holds leads: email, subject, message, send_status, send_ts. New rows arrive manually or via external form submissions.

The Gap

No system decides which rows should trigger an email. No deduplication when new data arrives. No automatic status logging after actions.

The Goal

Build Make scenarios that handle filtering, dispatch, search, and sync automatically — with full audit trail and zero manual intervention.

WHAT WAS BUILT: Two independent Make scenarios that together cover the full lead processing cycle — from monitoring a data source to syncing records and dispatching emails. Each scenario includes a validation layer so only clean, complete records trigger actions.

Three operational gaps that cost time and data quality

Unfiltered Sends

100%

of rows triggered email

Emails were sent to all spreadsheet rows regardless of lead status — no condition check, no record of what was sent or when it happened.

Duplicate Records

∞

risk of duplicates

When a contact submitted data again through an external source, no system checked if they already existed — a new row was always added.

No API Pre-validation

0

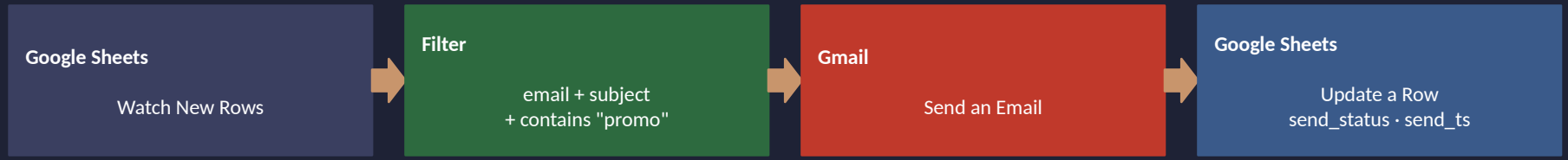
endpoints tested upfront

External data sources were connected to automations without prior testing, causing runtime errors when the response format didn't match.

OBJECTIVE: Replace all manual lead handling with two automated Make scenarios that filter, dispatch, validate, and sync data — reliably, without human intervention after initial setup.

Two scenarios — one complete pipeline

SCENARIO 1 — Conditional Email Dispatch with Status Logging

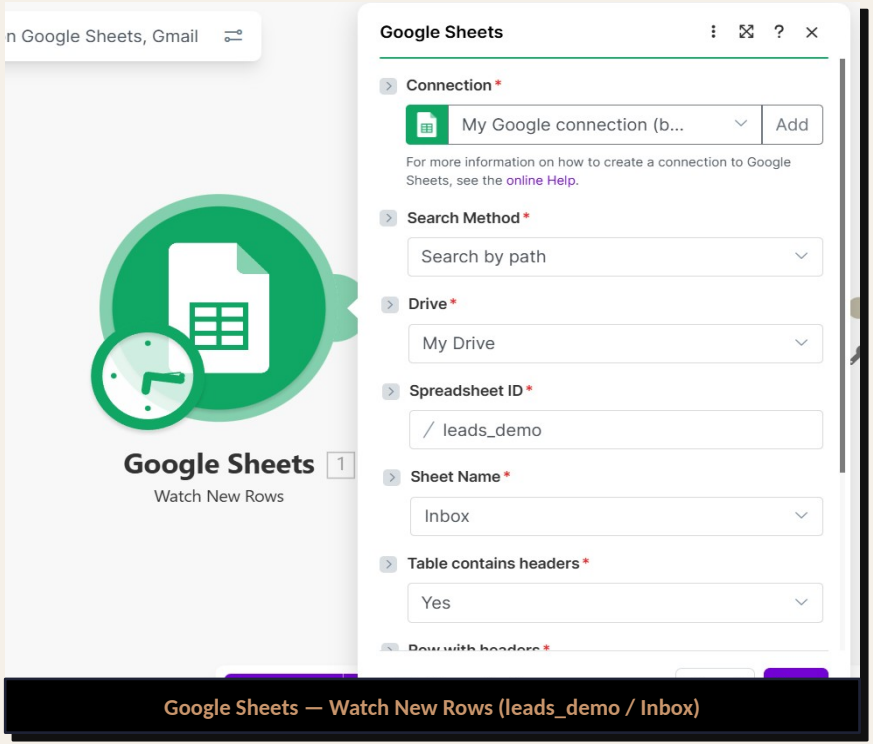


SCENARIO 2 — Webhook-Driven Upsert: Update Existing or Create New Record

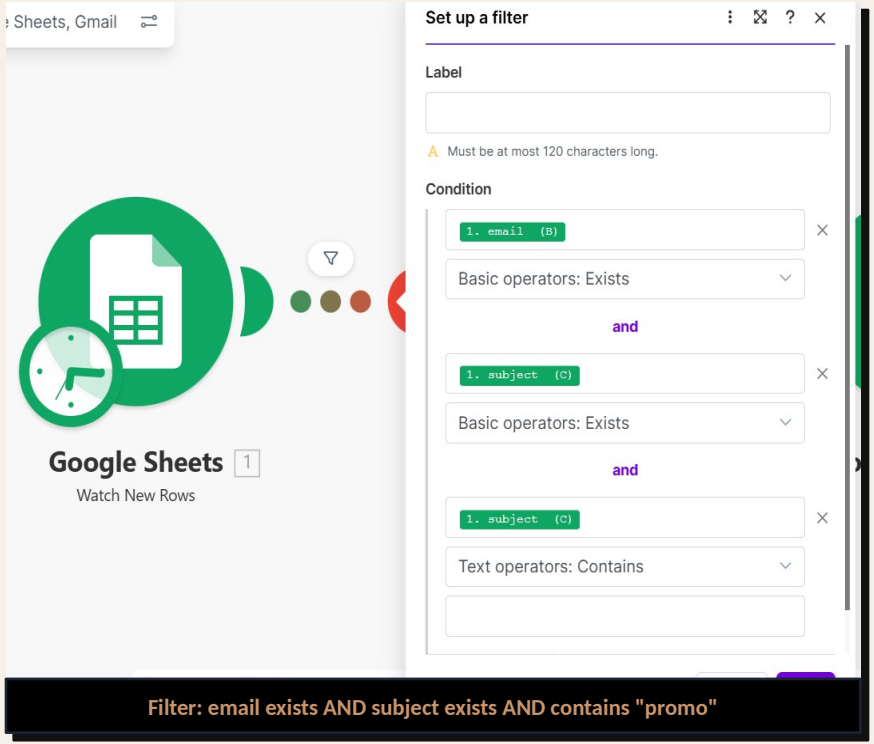
KEY PRINCIPLE: Every action is conditional. No email is sent and no row is updated unless all filter conditions pass first.

SHARED FOUNDATION: Google Sheets as lead database · Gmail for delivery · Postman for API validation before any Make connection

Step 1: Watch new rows · Step 2: Apply 3-condition filter



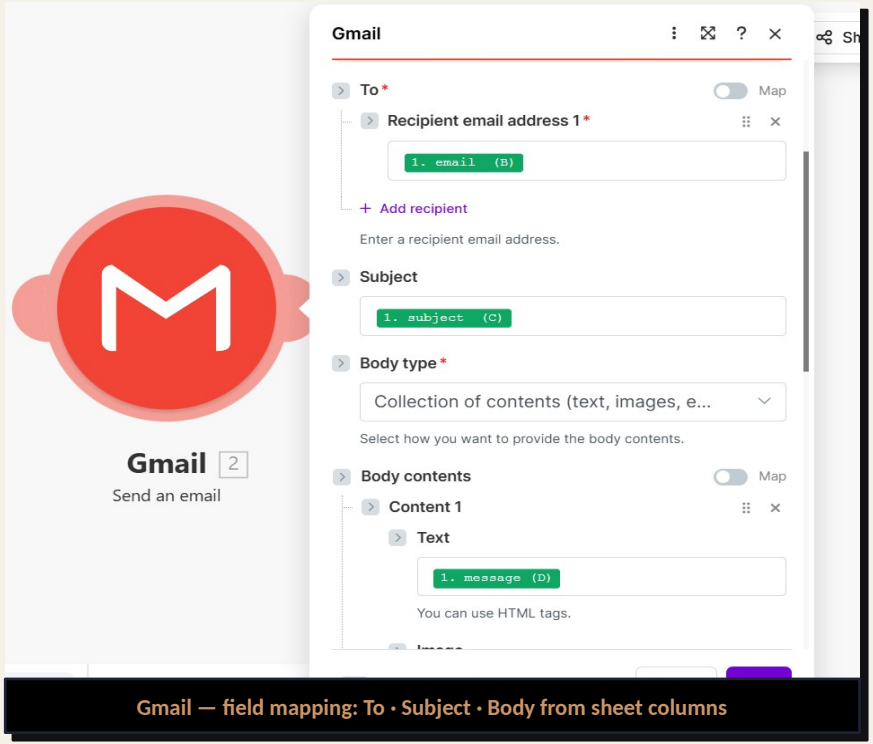
Google Sheets — Watch New Rows (leads_demo / Inbox)



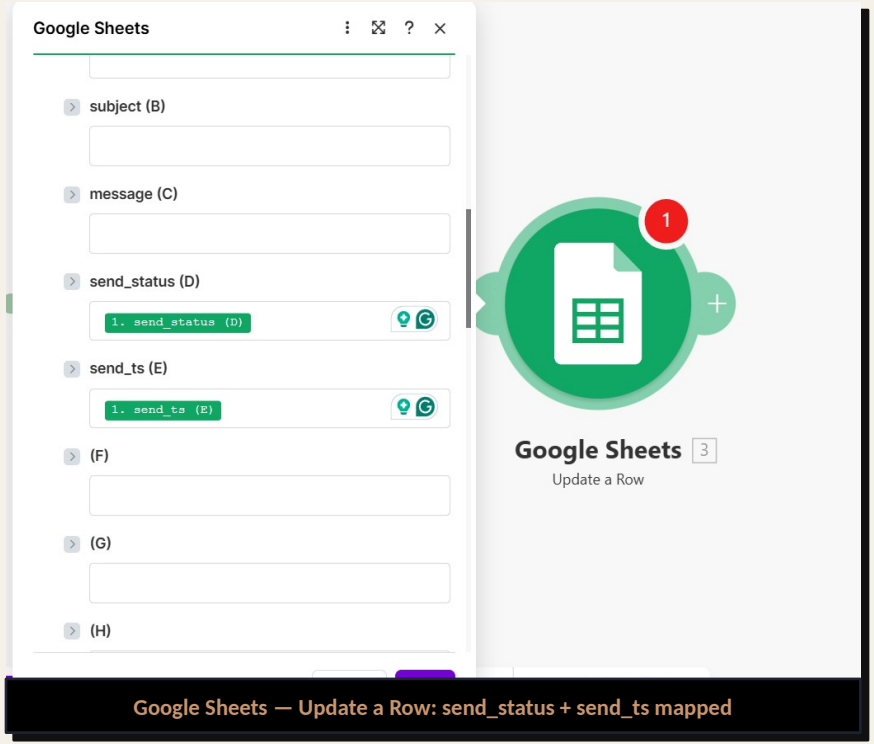
Filter: email exists AND subject exists AND contains "promo"

Trigger monitors Inbox sheet for new rows. Filter passes only rows where email is filled, subject exists, AND subject contains the word "promo" (case insensitive).

Step 3: Send email · Step 4: Write send_status + timestamp



Gmail — field mapping: To · Subject · Body from sheet columns



Google Sheets — Update a Row: send_status + send_ts mapped

Gmail maps To → email (B), Subject → subject (C), Body → message (D). After send, Update a Row writes send_status = "sent" and send_ts = now() to the same row automatically.

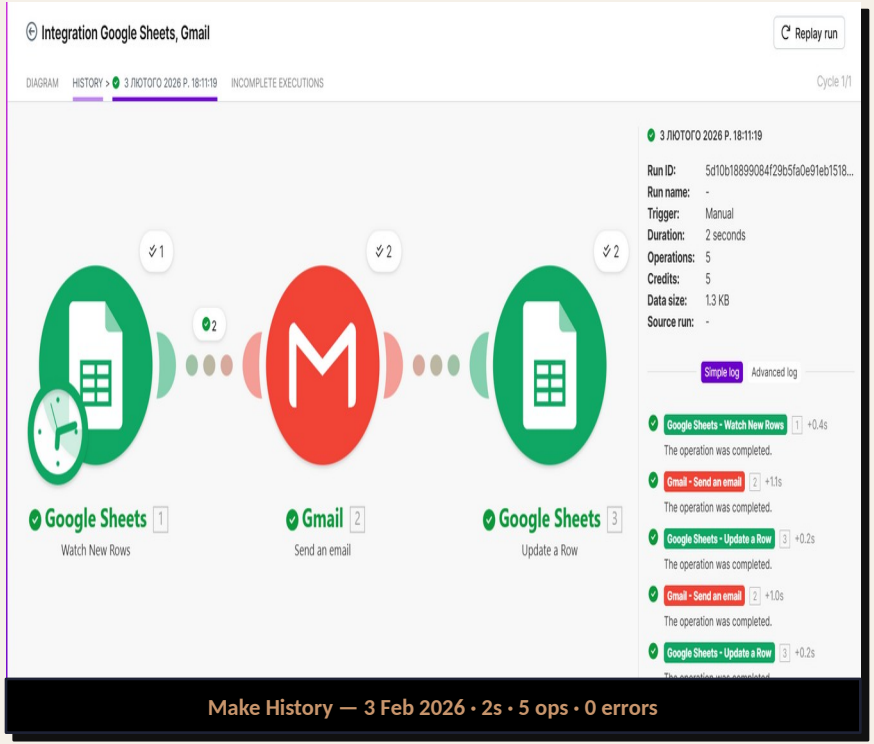
Google Sheets data · Make execution confirmed green

leads_demo

Файл Змінити Вигляд Вставити Формат Дані Інструменти Розширення Довідка

	A	B	C	D	E	F	G
1		email	subject	message	send_status	send_ts	
2	1	borat8907@gmail.com	spring promo -10%	Hello!			
3	2	borat8907+1@gmail.com	VIP promo	Limited offer			
4	3	borat8907+2@gmail.com	Any subject	No email case			
5	4	borat8907+3@gmail.com		No subject case			
6	5	borat8907+4@gmail.com	Regular update	Just info			
7			promo test				
8			hello test				
9							
10							
11							

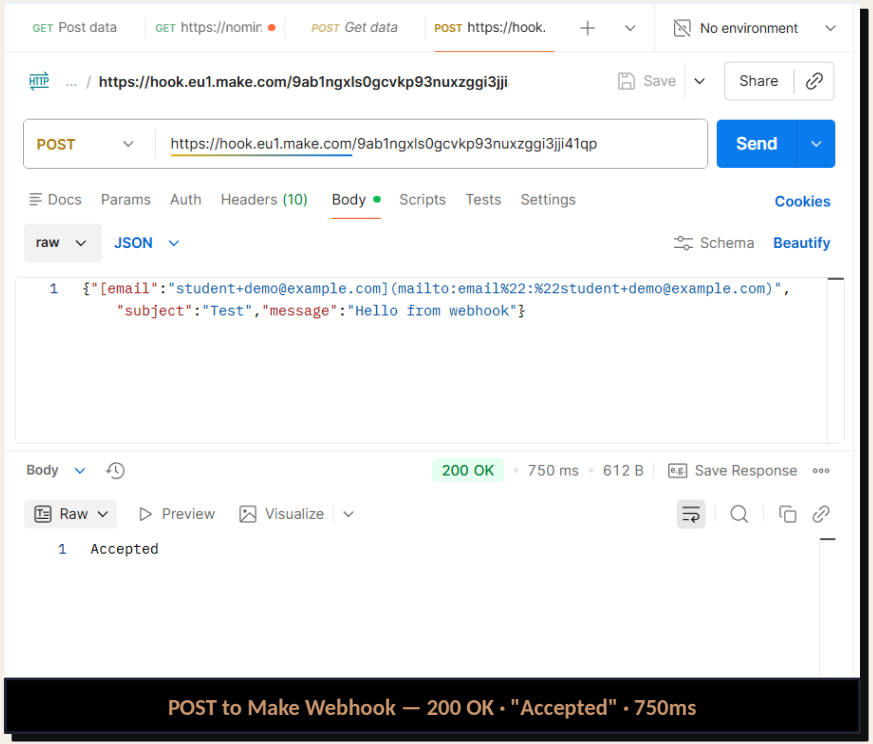
leads_demo spreadsheet — test rows with promo and non-promo subjects



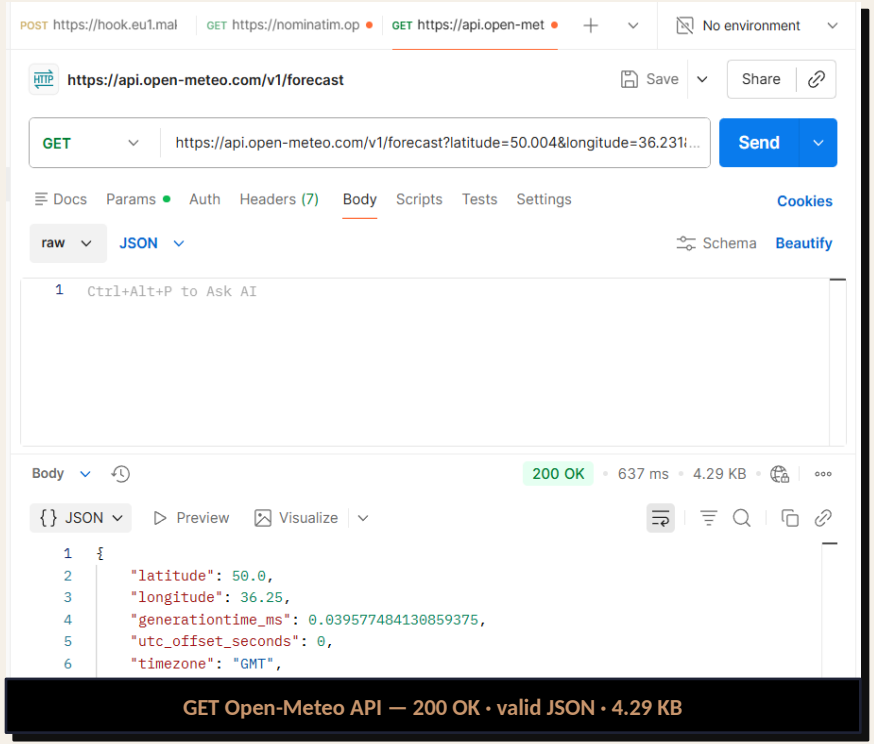
Make History — 3 Feb 2026 · 2s · 5 ops · 0 errors

Test confirmed: rows 1 and 2 (contain "promo") passed filter → Gmail sent → row updated. Rows 3–5 (no promo or no email) were blocked. History shows all operations green, 2 seconds total.

Testing external endpoints before connecting to Make



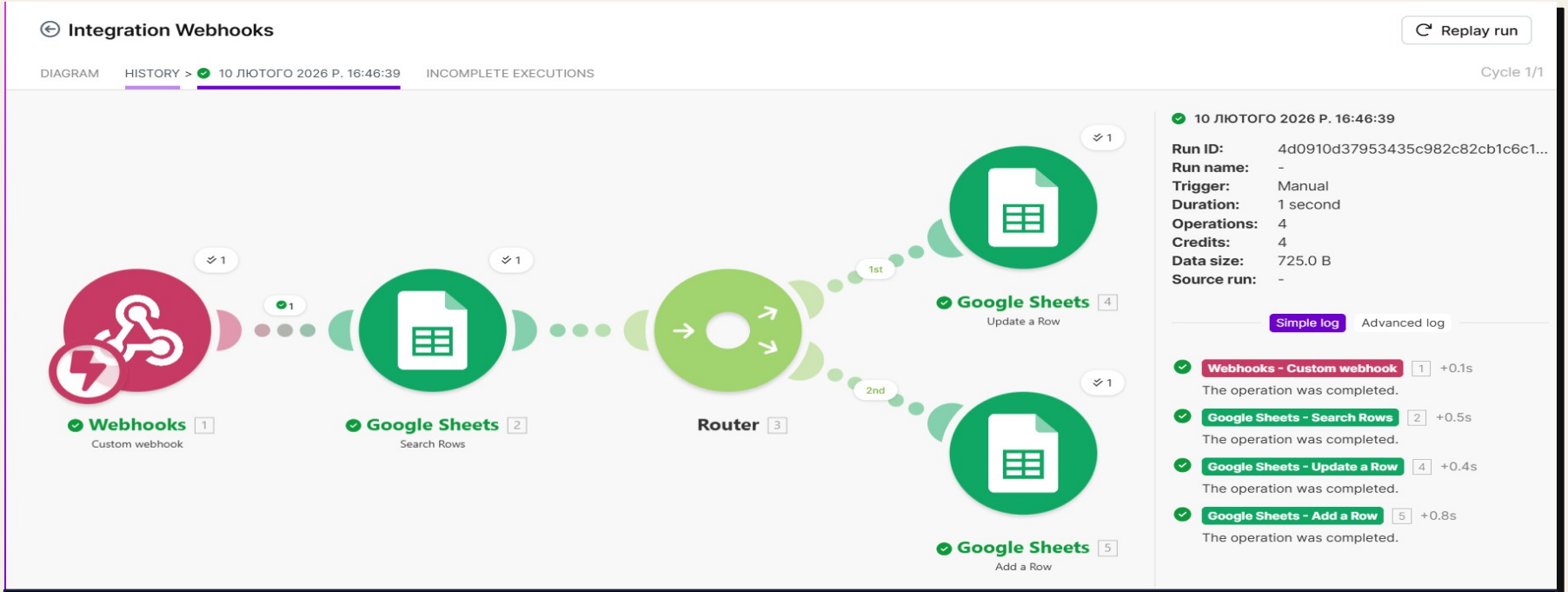
POST to Make Webhook — 200 OK · "Accepted" · 750ms



GET Open-Meteo API — 200 OK · valid JSON · 4.29 KB

API-first approach: all endpoints validated in Postman before any Make module is connected. Left: webhook POST returns 200 OK + "Accepted". Right: Open-Meteo GET returns structured JSON with hourly temperature data.

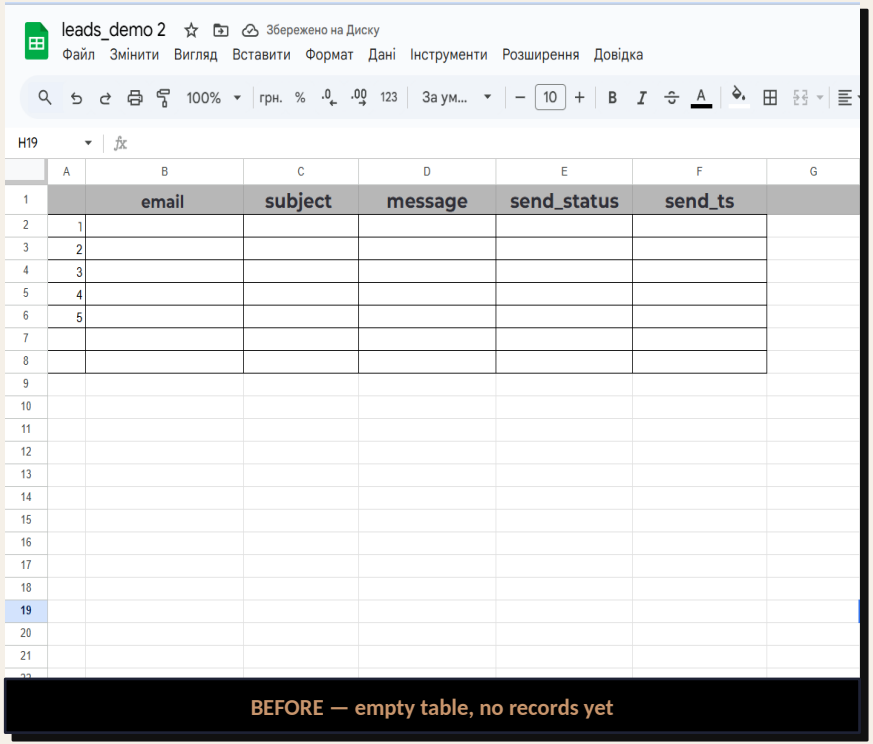
Webhook → Search → Router: Update existing or Add new record

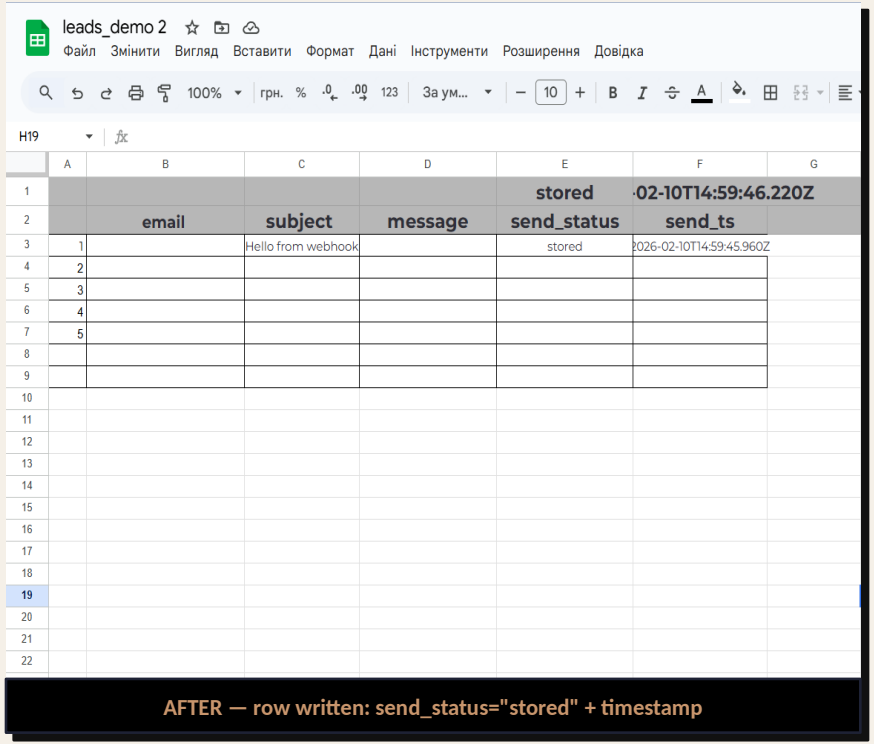


Make History — 10 Feb 2026 · 1s · 4 operations · both Router branches confirmed

History confirms: Webhook (1) received POST → Search Rows (2) found contact → Router (3) chose correct branch → Update a Row (4) updated, Add a Row (5) added new. Both branches active. Zero errors.

Google Sheets before and after webhook execution





Left: leads_demo 2 before any webhook call — table empty. Right: after POST via Postman — row 1 appears with subject "Hello from webhook", send_status = "stored", send_ts = 2026-02-10T14:59:45.960Z.

What was delivered and verified

2	2s	1s	200	0
Make scenarios built & confirmed	Scenario 1 execution time	Scenario 2 execution time	HTTP status all Postman tests	Errors in Make History

✓ **Zero unnecessary sends**	Filter blocks unqualified rows at the entry point — no emails go out to leads that don't match conditions
✓ **No duplicate records**	Router identifies returning contacts and routes to Update instead of creating a second entry
✓ **Full audit trail**	Every processed row gets send_status and send_ts automatically — no manual logging ever needed
✓ **Validated before connected**	All external endpoints tested in Postman prior to Make integration — eliminates the most common source of automation failures
✓ **Production-ready architecture**	Both scenarios are modular — Sheets → any CRM, Gmail → any email provider, webhook source → any form or platform

Everything used to build and validate the pipeline

Make (Integromat)

Main automation platform — both scenarios built and executed here

Google Sheets

Lead database — stores email, subject, message, send_status, send_ts

Gmail

Email delivery channel — mapped fields from spreadsheet columns

Postman

API testing — validated webhook POST and Open-Meteo GET before Make connection

Custom Webhook

Entry point for Scenario 2 — receives POST JSON from external source

Router (Make)

Branching logic — routes to Update or Add based on Search Rows result

Filter (Make)

Condition gate — ensures only complete, qualified records trigger actions

Open-Meteo API

External REST API used for GET validation practice (200 OK confirmed)

No paid tools, no custom code. The entire pipeline is built with free-tier services and no-code modules — demonstrating that production-grade automation is accessible without engineering resources.