01 / SET UP

One brief. One intake row.

Validate a content brief, label its queue and save it in Google Sheets. Start with fictional data and an unpublished workflow.

1. Import and run the safe demo

In a new n8n workflow, choose **Import from File** and open **WOL-Client-Intake.n8n.json**. Select **Run demo**, execute it and open **Demo results**. Expect 6 cases: 4 accepted, 2 rejected; every item shows zero writes. This branch needs no credentials.

Cloud verified: **n8n 2.41.3**, with JavaScript Code nodes enabled. Read **VERIFICATION.md** for the checks actually completed. The demo does not verify Google storage or HTTP authentication.

2. Prepare a dedicated test sheet

Create a blank spreadsheet and name its tab **Intake**. Copy the tab-separated line in **sheet-header.txt** into A1: it must populate 11 columns, A through K. Keep a continuous table with no merged cells or spacer rows.

3. Connect the two Sheets nodes

In **Find existing brief** and **Save new brief**, select a Google Sheets OAuth2 credential with edit access and the same Document. Keep Sheet = Intake. Check lookup column = brief_id and all 11 append mappings. Keep cell format **RAW / Let n8n format**; automatic retries stay off.

Cloud: use Sign in with Google when offered. Self-hosted: configure your OAuth app, Sheets + Drive APIs and the redirect URI shown in n8n. Follow the official OAuth guide in REFERENCES.md.

4. Protect the webhook

In **Brief webhook**, select a **Header Auth** credential: header name **X-WOL-Key**, with a strong random value from your password manager. Keep POST and Respond to Webhook mode. Never embed this key in a public browser form.

02 / VERIFY

Run the acceptance test.

Use the webhook Test URL. Select Listen for test event before each request. Leave the workflow unpublished during setup.

Send one request

Open a terminal inside the extracted package and run **python test-request.py valid**. Use py on Windows or python3 on macOS/Linux if needed. Paste the Test URL and enter the key at the hidden prompt. The helper sends one request and never retries automatically.

Alternative: an HTTP client sending POST JSON from samples/valid.json, with Content-Type: application/json and your X-WOL-Key header.

Test	Expected result
valid (fresh ID)	201 created; one VIDEO row
valid again	200 duplicate; still one row, unchanged
invalid	422 with field errors; no new row
article, then social	201 each; EDITORIAL and SOCIAL rows
Wrong header key	403 on tested Cloud; no storage action
Inaccessible test document	Storage error; never a success response. Restore targets AND all column mappings.

What the workflow accepts

Required: brief_id, client_name, contact_email, content_type, title, objective and due_date. **Optional:** asset_url. Send one object; all values are strings. Use video, article or social and a real YYYY-MM-DD date. The sample files are the reference payloads.

The workflow adds queue, intake_status = NEW and received_at in UTC. It labels one shared intake sheet; it does not notify a person, publish content or send client email. Unknown fields are discarded.

Record the evidence

Complete **ACCEPTANCE.md** with your exact n8n version, hosting type, execution IDs and row counts. Also test a modified payload with the same ID: it must leave the original row unchanged. Test a formula-like title as plain text.

03 / OPERATE

Know the starter limits.

A small, understandable intake workflow. Keep the test-release label until the Google integration checks pass in your intended environment.

Duplicates: first accepted brief wins

Use a stable, globally unique brief_id; retain it for retries. Existing IDs return duplicate without overwriting. A revision needs a new ID, such as acme-001-r2, or a deliberate manual edit.

Single sequential sender only. Lookup then append is not atomic. Concurrent requests may create duplicates or interfere with writes. The template does not enforce a global lock. Multi-sender production needs a durable queue/database with a unique key.

Recover from errors deliberately

After a timeout or storage error, inspect the sheet for the brief_id before retrying: a write may already have completed. Fix permissions or column mappings first. For quota errors, stop the sender, then use bounded backoff. Setup errors may stop n8n before its custom response node.

Automatic write retries are off. Formula-like text is prefixed with an apostrophe and stored using RAW; preserve this if exporting. Asset links are stored only, never downloaded.

Credentials, data and running costs

The shared key authenticates a trusted sender, not each client. Use fictional data first. Review sheet access and n8n execution retention; logs may include brief data and request headers. Decide retention and the intake notice before using real client data. Brief submission is not email-marketing consent.

No AI API is required. The template is free; n8n Cloud or self-hosting and your Google plan may cost money. Sheets API use has no additional API charge, but quotas apply. Current limits and setup links are in REFERENCES.md.

Release gate and measurement

Pass the full acceptance record in the target environment before public distribution. Cloud and self-hosted compatibility are separate checks. Measure active handling time, including review and error recovery, against manual processing of the same briefs. **No hours-saved claim has been measured.**