

# Lab 9: Writer, critic, reviser

AI Administrator · Day 5 · Session 9 · Multi-Step and Multi-Agent Workflows

Participant lab handout

---

## Objective

---

Extend what you built in Module 8 into a **two-agent chain** for supplier e-mails in n8n: a **Writer** drafts from a form, a **Critic** on a different model checks the draft against a checklist and answers APPROVED or REVISE with numbered points, the Writer revises **at most twice**, every pass is logged to a Google Sheet, an approved draft lands in Gmail Drafts for a person to send, and anything still not approved after two revisions goes to a person by e-mail.

**Time:** 35 minutes in breakout rooms. Finish at home before Session 10 if needed. **Deliverable:** at the end.

## What you need

---

- Your **n8n Cloud** workspace from Session 5 (check the days left at the top) and the **same Google account** you used for Gmail and Sheets.
- From the Zoom chat: Day5\_S9\_Prompts.pdf (the two prompts), Day5\_S9\_Review\_Log.xlsx, Day5\_S9\_Test\_Cases.xlsx. AI Caravan Ltd. and its suppliers are fictional.
- No API key: both models (Gemini 2.5 Flash for the writer, Gemini 2.5 Pro for the critic) run on your trial's **Gateway credits**. Our three test cases cost a few cents.

Two words for today: a **pass** is one trip through writer and critic; a **hand-off** is what one step passes to the next. In n8n both chains output only their text, so you will add two **Edit Fields** nodes that carry the task, the facts and the counter forward. That is the module's hand-off rule in node form.

## Before you build (5 minutes)

---

1. **The Review log.** Upload Day5\_S9\_Review\_Log.xlsx to Google Drive and open it with Google Sheets (**File, Save as Google Sheets**). Keep the name *Review log*; the first tab has the five column headers.
2. **The test cases.** Open Day5\_S9\_Test\_Cases.xlsx (no upload needed): three form submissions and what should happen.
3. **The workflow.** **Create workflow**, rename it **M9 Supplier email with review**. Workflow menu (...), **Settings**: Timezone = yours (we use Asia/Amman) so the log timestamps are local.

## Part A: the form and the writer

### 1. Start from a form

First step: **On form submission** (the n8n Form Trigger). Form Title *AI Caravan Ltd. - Supplier email with review*. Add six fields, in this order: **Supplier** (text), **Supplier email** (email), **Your email** (email: where an escalation goes), **Purpose of email** (textarea), **Key facts** (textarea), **Deadline** (text, not required). The form replaces the chat so every run is repeatable.

### 2. Add the counter (the loop cap starts here)

+ after the form: **Edit Fields (Set)**, rename it **Start**. Add nine fields, *Include Other Input Fields* off:

| Field          | Type          | Value                            |
|----------------|---------------|----------------------------------|
| supplier       | String        | {{ \$json['Supplier'] }}         |
| supplier_email | String        | {{ \$json['Supplier email'] }}   |
| your_email     | String        | {{ \$json['Your email'] }}       |
| purpose        | String        | {{ \$json['Purpose of email'] }} |
| facts          | String        | {{ \$json['Key facts'] }}        |
| deadline       | String        | {{ \$json['Deadline']    '' }}   |
| revision       | <b>Number</b> | 0                                |
| feedback       | String        | (empty)                          |
| draft          | String        | (empty)                          |

**revision must be a Number.** As text, "0" + 1 is "01" and the loop never reaches the cap (Trap 1).

### 3. Add the writer

+ after Start: **Basic LLM Chain**, rename it **Writer**. Prompt: *Define below*, paste the Writer prompt from `Day5_S9_Prompts.pdf`; the braces must turn green (Expression mode). + under *Model*: **Google Gemini Chat Model**, Gateway credits, **models/gemini-2.5-flash**, temperature 0.

### 4. Keep the draft (first hand-off)

+ after Writer: **Edit Fields (Set)**, rename it **Keep draft**. The chain output only text, so put the state back, *Include Other Input Fields* off:

| Field                                                          | Type   | Value                                                         |
|----------------------------------------------------------------|--------|---------------------------------------------------------------|
| supplier, supplier_email, your_email, purpose, facts, deadline | String | {{ \$('Start').item.json.supplier }} and so on, one per field |

| Field    | Type   | Value                                                                              |
|----------|--------|------------------------------------------------------------------------------------|
| revision | Number | {{ \$('Next revision').isExecuted ? \$('Next revision').item.json.revision : 0 }}  |
| feedback | String | {{ \$('Next revision').isExecuted ? \$('Next revision').item.json.feedback : '' }} |
| draft    | String | {{ \$json.text.trim() }}                                                           |

The two `isExecuted` lines read “on the first pass there is no Next revision node yet, so 0 and empty”. You add the Next revision node in step 8; the expression is fine to type now.

## Part B: the critic and the log

### 5. Add the critic

+ after Keep draft: **Basic LLM Chain**, rename it **Critic**. Paste the Critic prompt (six checks, fixed answer APPROVED or REVISE:). + under *Model*: **Google Gemini Chat Model**, Gateway credits, **models/gemini-2.5-pro**, temperature 0. A different and stronger model than the writer: the check must not share the writer’s assumptions, and a weak critic invents faults (we tried Flash-Lite: it sent a fixable draft to escalation).

### 6. Record the verdict (second hand-off)

+ after Critic: **Edit Fields (Set)**, rename it **Verdict**, *Include Other Input Fields* off:

| Field                                                                 | Type   | Value                                               |
|-----------------------------------------------------------------------|--------|-----------------------------------------------------|
| verdict                                                               | String | {{ \$json.text.trim() }}                            |
| timestamp                                                             | String | {{ \$now.toFormat('yyyy-MM-dd HH:mm:ss') }}         |
| supplier, supplier_email, your_email, purpose, facts, deadline, draft | String | {{ \$('Keep draft').item.json.supplier }} and so on |
| revision                                                              | Number | {{ \$('Keep draft').item.json.revision }}           |

### 7. Log every pass

+ after Verdict: **Google Sheets**, *Append Row in Sheet*. Document **From list: Review log**, Sheet **From list: Review log**. Map the columns: timestamp, supplier, revision, verdict, draft (drag from the input, or

type `{{ $json.timestamp }}` and so on). Options, **Cell Format: RAW**. This is the audit trail Module 10 will ask you for.

**The Sheets node outputs the row it wrote**, not your item: after it, `$json` has only the five columns. Every node from here on reads its fields from the Verdict node, `${'Verdict'}.item.json...` (Trap 2).

## Part C: decide, loop, cap

---

### 8. Approved?

+ after the Sheets node: **IF**, rename it **Approved?**. Condition: `{{ ${'Verdict'}.item.json.verdict }}` *String, starts with APPROVED*.

**True branch: Gmail, Create a draft**. To: `{{ ${'Verdict'}.item.json.supplier_email }}` (under Options, *To Email*). Subject: AI Caravan Ltd. Finance: `{{ ${'Verdict'}.item.json.purpose.slice(0, 60) }}`. Message: `{{ ${'Verdict'}.item.json.draft }}`. A draft, not Send: a person reads it in Gmail and clicks send.

**False branch: Edit Fields (Set)**, rename it **Next revision**, *Include Other Input Fields* off: the same seven carried fields from `${'Verdict'}.item.json...`, plus **revision** (Number) = `{{ ${'Verdict'}.item.json.revision + 1 }}` and **feedback** = `{{ ${'Verdict'}.item.json.verdict }}`.

### 9. Under the cap?

+ after Next revision: **IF**, rename it **Under the cap?**. Condition: `{{ $json.revision }}` *Number, is less than or equal to 2*.

**True branch:** drag the connector back to the **Writer** node's input. The canvas now shows a loop. The Writer reads feedback and draft from its input and revises.

**False branch: Gmail, Send a message**. To: `{{ $json.your_email }}`. Subject: Needs a human: supplier email to `{{ $json.supplier }}`. Message: the purpose, the facts, the last draft (`{{ $json.draft }}`) and the critic's last points (`{{ $json.feedback }}`). This is the human route the module requires when the cap is reached.

**Save**. Then click **Execute workflow**: n8n opens the test form in a new tab and waits.

## Part D: three cases

---

Submit the form three times with the rows of `Day5_S9_Test_Cases.xlsx`, and after each one open the Review log and the Executions tab.

| # | Case                                                                                 | Expected                                                                                                                                                             |
|---|--------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Simple: ask Northwind for the 2027 price list, facts complete, deadline given        | APPROVED on revision 0 (sometimes one revision). A draft to the supplier in Gmail Drafts.                                                                            |
| 2 | Tempting: ask Omar (Contoso) to call Sara back; Sara's mobile number is in the facts | REVISE on revision 0 (point 5: the mobile number), APPROVED on revision 1 without it.                                                                                |
| 3 | Impossible: promise Tailspin payment on 30 October, but no date has been approved    | REVISE three times (point 2 against point 6), then the escalation e-mail reaches <i>your</i> inbox with the last draft and the points. Nothing goes to the supplier. |

Our run of 21 October 2026: six rows in the Review log, one escalation, two drafts. Yours should look the same; the exact wording of the critic's points differs between runs.

## If something goes wrong

| What you see                                                                         | Fix                                                                                                                                                               |
|--------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| The Writer's prompt shows <code>{{ \$json.supplier }}</code> literally in the draft  | The Prompt field is in Fixed mode. Click into it and switch to Expression (the braces turn green).                                                                |
| "Referenced node is unexecuted" on Keep draft                                        | The <code>isExecuted</code> guard is missing from the revision or feedback line (step 4).                                                                         |
| The loop never escalates; revision reads 01, 011                                     | revision is a String somewhere. Make it a Number in Start, Keep draft, Verdict and Next revision (Trap 1).                                                        |
| "The email address " in the 'To' field isn't valid"                                  | A node after the Sheets node reads <code>\$json.supplier_email</code> or <code>\$json.your_email</code> : use <code>\$( 'Verdict' ).item.json...</code> (Trap 2). |
| The critic answers in prose, never exactly APPROVED                                  | Temperature 0 and the prompt's last sentence intact; the IF checks <i>starts with</i> APPROVED, so "APPROVED." also passes.                                       |
| Case 1 goes to a revision: "requests the 2027 price list, which is not in the facts" | The critic did not receive the purpose. The prompt sheet's Critic has the line <i>Purpose of the email</i> ; the module's original did not (Trap 3).              |
| Case 3 is approved first time                                                        | The critic lacks check 6 (does what the purpose asks); the writer just left the impossible part out. Use the prompt sheet's six checks.                           |
| Google Sheets: 403 or "Sheet with ID gid=0 not found"                                | The sheet must belong to the same Google account as the credential; pick Document and Sheet From list (Session 5).                                                |

| What you see                                                                                    | Fix                                                                                                                       |
|-------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| An OpenAI model under Gateway credits: “Gateway credits don’t currently support this operation” | Gateway credits serve Gemini chat models only. Gemini 2.5 Pro is the critic; with your own OpenAI key, gpt-4.1 works too. |

## Deliverable: M9-writer-critic

---

1. A screenshot of the loop on the canvas with the counter (Start, Next revision) and both IF nodes visible.
2. Your *Review log* sheet showing at least five rows across the three cases, including one escalation.
3. **Workflow menu (...), Export JSON**, saved as `M9-writer-critic.json`.

## Going further (optional)

---

- Give the critic a second checklist item about tone in Arabic and submit a case in Arabic. Does the fixed APPROVED / REVISE form survive?
- Replace the escalation e-mail with a Slack message (if your organisation uses Slack) and compare the audit trail.
- Make the supervisor shape: an AI Agent whose tool is this workflow (*Call n8n Workflow Tool*), so a chat request “write to Northwind about the price list” fills the form for you. What is the new failure point?