

Lab 10: An approval gate for the triage flow

AI Administrator · Day 5 · Session 10 · Controls I: Human-in-the-Loop Design

Participant lab handout

Objective

Take the e-mail triage workflow you built in Module 6 and make sure **no reply leaves without a person seeing it**. The workflow drafts a reply, sends it to you for approval with the full text and two buttons, and then either sends it (approved) or labels the e-mail *needs-human* (rejected, or no answer in time). Every decision, including rejections and timeouts, is appended to an **Approval log**.

Time: 35 minutes in breakout rooms. Finish at home before Session 11 if needed. **Deliverable:** at the end.

What you need

- Your **n8n Cloud** workspace from Session 5 (check the days left at the top) and your **M6 Inbox triage** workflow from Session 6.
- The **same Google account** for Gmail and Sheets as in Sessions 5 and 6.
- From the Zoom chat: Day5_S10_Approval_Log.xlsx, Day5_S10_Tier_Worksheet.xlsx, Day5_S10_Send_Test_Emails.json. The three customers in the test e-mails are fictional.
- No API key: the model (Gemini 2.5 Flash) runs on your trial's **Gateway credits**.

Two words for today. A **gate** is a step where the run stops until a person answers. A **tier** is how bad a wrong call would be, from 1 (read-only) to 5 (money, people, legal). Only Tier 4 and 5 actions need a gate before they run.

Before you build (5 minutes)

1. **The Approval log.** Upload Day5_S10_Approval_Log.xlsx to Google Drive and open it with Google Sheets (**File, Save as Google Sheets**). The first tab, *Approval log*, has the nine column headers. **Keep the header row:** on an empty sheet the Append node ignores your mapping and writes the wrong columns (Trap 1).
2. **The label.** In Gmail, left menu, **Labels, +:** create the label **needs-human**.
3. **The workflow.** Open **M6 Inbox triage**, workflow menu (...), **Duplicate**. Name the copy **M10 Triage with approval gate**. If your M6 workflow does not work, ask the co-host for Day5_S10_Approval_Gate_Solution.json and start at step 5.
4. **The test sender.** **Create workflow**, menu (...), **Import from File**, Day5_S10_Send_Test_Emails.json. In **Send to myself** paste your own Gmail address and pick your Gmail credential.

Part A: tier, filter, draft

1. Tier every node

Open Day5_S10_Tier_Worksheet.xlsx, first tab. For each node choose the tier and the control from the dropdowns. Ask: *if this ran wrongly a hundred times before anyone noticed, what would it cost?* Check yourself on the *Answer key* tab afterwards. Only one node needs a gate.

2. The trigger: only the test e-mails, never the workflow's own mail

Open **New email** (Gmail Trigger). **Simplify** off. In **Filters**, **Search** type:

```
subject:"[S10 test]" -subject:"Re:" -subject:"Approve this reply"
```

Trap 2. Without the two exclusions the trigger also matches the approval request ("Approve this reply? Re: [S10 test] ...") and the reply the workflow sends ("Re: [S10 test] ..."). Published, the flow would draft replies to its own replies.

3. Draft reply

Keep your **Classify email** and **Clean label** nodes. From **Clean label** add a **Basic LLM Chain**, rename it **Draft reply**. Source for Prompt: *Define below*, switch the prompt to **Expression** and paste:

```
You draft replies to customer emails for the support team of a small software company.
Write a short, polite reply (under 120 words) to the email below. Category: {{
$json.label }}.
Do not promise refunds, discounts, dates or fixes; say what happens next and that a
colleague
will follow up if needed. Sign off as "Customer Support". Reply with the email body
only:
no subject line, no commentary.
Email subject: {{ $('New email').item.json.subject }}
Email body: {{ $('New email').item.json.text }}
```

Connect the same Gemini model (Gemini 2.5 Flash, Gateway credits, temperature 0) to its Model input. If your Clean label also goes to *Route by label*, leave that connection: the labels still run alone (Tier 2).

4. Keep reply (the hand-off)

+ after Draft reply: **Edit Fields (Set)**, rename it **Keep reply**, *Include Other Input Fields* off:

Field	Value
message_id	{{ \$('New email').item.json.id }}
to	{{ \$('New email').item.json.from.text }}
subject	Re: {{ \$('New email').item.json.subject }}

Field	Value
label	{{ \$('Clean label').item.json.label }}
reply	{{ \$json.text.trim() }}
approver	your own Gmail address (plain text)

The chain outputs only its text, and the approval node will replace the item again: carry everything the send and the log need now.

Part B: the gate

5. Ask for approval

+ after Keep reply: **Gmail**, operation **Send and Wait for Response** (in older menus: *Send message and wait for response*). Rename it **Ask for approval**.

- **To:** {{ \$json.approver }}
- **Subject:** Approve this reply? {{ \$json.subject }}
- **Message** (Expression):

The triage workflow wants to send this reply. Nothing is sent until you approve.

To: {{ \$json.to }}
 Subject: {{ \$json.subject }}
 Category: {{ \$json.label }}

--- reply ---
 {{ \$json.reply }}
 --- end ---

No answer within 3 minutes: nothing is sent and the e-mail is labelled needs-human.

- **Response Type:** Approval. **Approval Options, Type of Approval:** *Approve and Disapprove*; button labels *Approve and send* and *Reject*.
- **Options, Limit Wait Time:** after a time interval, **3 minutes** for the lab (24 hours in real use). **Append n8n Attribution** off.

This is rule 1: **show the actual action**. The approver sees the recipient, the subject and the whole reply.

6. Decision and Approved?

+ after Ask for approval: **Edit Fields**, rename it **Decision**, *Include Other Input Fields* off:

Field	Value
decision	{{ \$json.data?.approved === true ? 'approved' : (\$json.data?.approved === false ? 'rejected' : 'timed out') }}

Field	Value
message_id	{{ \$('Keep reply').item.json.message_id }}
to	{{ \$('Keep reply').item.json.to }}
subject	{{ \$('Keep reply').item.json.subject }}
reply	{{ \$('Keep reply').item.json.reply }}
approver	{{ \$json.data?.approved === undefined ? 'nobody (timed out)' : \$('Keep reply').item.json.approver }}

Then an **IF** node, rename it **Approved?: {{ \$json.decision }} is equal to approved**.

Trap 3. A timeout is not a “no”. Measured on 23 October 2026: approved gives data.approved = true, rejected gives false, and when the time limit runs out the node passes its **input item through unchanged**, with no data at all, on the same output. An IF that tests “is not false” would send the reply when nobody answered. Always test for approved.

7. Act, and log both branches

- **True branch: Gmail**, Resource Message, Operation **Reply**, Message ID {{ \$json.message_id }}, Email Type Text, Message {{ \$json.reply }}, Append Attribution off. Rename it **Send reply**. Then **Google Sheets, Append Row**, rename **Log sent**.
- **False branch: Gmail**, Operation **Add Label**, Message ID {{ \$json.message_id }}, Label *needs-human* (From list). Rename **Label needs human**. Then **Google Sheets, Append Row**, rename **Log not sent**.

In both Sheets nodes: Document and Sheet **From list** (Approval log), Mapping *Map Each Column Manually*, Options **Cell Format: Let n8n format** (RAW, so text stays text):

Column	Log sent	Log not sent
timestamp	{{ \$now.toFormat('yyyy-MM-dd HH:mm:ss') }}	same
workflow	{{ \$workflow.name }}	same
run_id	{{ \$execution.id }}	same
action_proposed	Reply to {{ \$('Decision').item.json.to }}: {{ \$('Decision').item.json.subject }}	same
tier	4	same

Column	Log sent	Log not sent
approver	<code>{{ \$('Decision').item.json.approver }}</code>	same
decision	<code>{{ \$('Decision').item.json.decision }}</code>	same
final_content	<code>{{ \$('Decision').item.json.reply }}</code>	(empty)
notes	sent as approved	not sent; labelled needs-human (<code>{{ \$('Decision').item.json.decision }}</code>)

Fields come from **Decision**, not from `$json`: the Gmail node before each log outputs the message it sent or labelled, not your item (Session 9's Trap 2). Rejections and timeouts are rows too: that is what makes the log **complete**.

Part C: three test e-mails, one at a time

Execute workflow on a Gmail trigger takes only the newest matching e-mail, so send one, run, decide, then the next.

Test	In S10 Send three test e-mails	In M10	What you should see
1 Khaled, cannot log in	<code>const i = 0;</code> , Execute workflow	Execute workflow; in Gmail open <i>Approve this reply? ...</i> , Approve and send	Green path to Send reply and Log sent; the reply in the thread
2 Lina, refund “today”	<code>i = 1</code>	Execute workflow; Reject	Label needs human, Log not sent, decision <i>rejected</i>
3 Nadia, a demo next week	<code>i = 2</code>	Execute workflow; do nothing for 3 minutes	Label needs human, Log not sent, decision <i>timed out</i> , approver <i>nobody (timed out)</i>

Open the Approval log: three rows. Compare with the *Example (our run)* tab.

Troubleshooting

Symptom	Cause and fix
The log has columns id, threadId, labelIds	The sheet was empty (Trap 1). Clear it, use the template's header row, map again.
The run picks up "Re: [S10 test] ..."	The trigger search lacks the two exclusions (Trap 2).
A reply was sent though nobody answered	The IF tests the wrong thing (Trap 3). Use <code>Decision and decision is equal to approved</code> .
The approval e-mail shows <code>{{ \$json.reply }}</code> literally	The Message field is in Fixed mode: switch it to Expression.
Approve was clicked but nothing happens	Use the newest approval e-mail; check the Executions tab shows the run as <i>Waiting</i> .
The link opens but the run keeps waiting	n8n ignores clicks that look like a bot (mail link scanners must not approve). Open the link in your normal browser.
Send reply: no message id	Keep reply and Decision must both carry <code>message_id</code> (step 4 and 6).
Label not found	Create <i>needs-human</i> in Gmail first, then pick it From list.

Deliverable: M10-approval-gate

1. The exported JSON of **M10 Triage with approval gate** (workflow menu, Export JSON).
2. A screenshot of one approval e-mail showing the full draft and both buttons.
3. Your **Approval log** with at least one *approved*, one *rejected* and one *timed out* row.

Finished early? Make the reply *edit-then-approve*: set Response Type to **Custom Form** with a textarea field *Final reply* (the draft is already in the message above it), and send what the approver typed instead of the draft. Untested in class: check the node's output in the run before you wire it. Or open your Session 8 Ops Desk agent and set **Options, Max Iterations** to 8.

Before you leave: if you publish M10, set the time limit back to 24 hours and check the trigger search first. Export every workflow you want to keep: the trial ends soon.