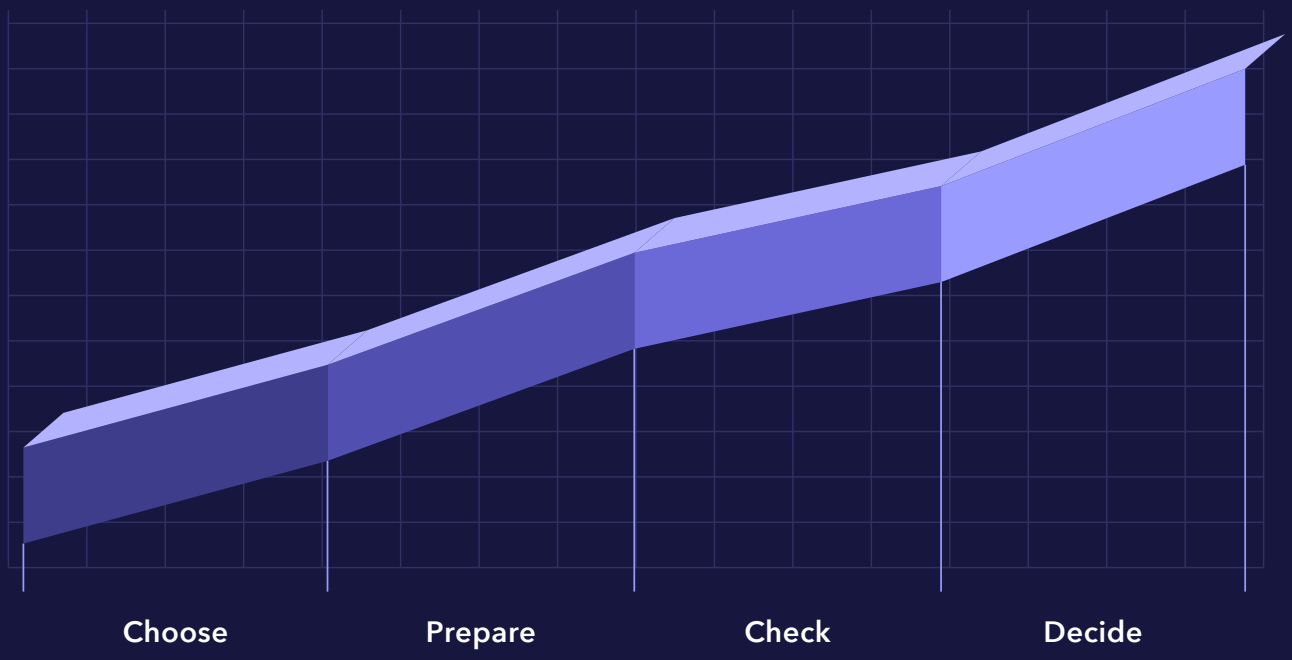
Independent user companion and implementation guide v1.0

The Nod Starter Kit

Set up safely. Start one useful workflow. Keep the important decisions yours.



4 starting paths · 20 prompts · 10 workflow recipes

Created by John Higham • BlueHAT.one

For the AI Startup Foundry community

This is not an official Nod by n8n publication or a replacement for its official documentation.

Prepared: 19 November 2026
Last updated: 19 November 2026
Official Nod documentation checked: 19 November 2026
Status: Approved for PDF export

The Nod Starter Kit

Set up safely. Start one useful workflow. Keep the important decisions yours.

Choose → **Prepare** → **Check** → **Decide**

4 starting paths · 20 prompts · 10 workflow recipes

Created by John Higham · BlueHAT.one

For the AI Startup Foundry community

This is not an official Nod by n8n publication or a replacement for its official documentation.

Prepared: 19 November 2026
Last updated: 19 November 2026
Official Nod documentation checked: 19 November 2026
Status: Approved for PDF export

Printing this guide? Skip the dark outer cover and print from this title page onward.

Table of contents

Welcome 1

Choose your first workflow 2

- What you'll need 2
- Path A — Source-checked research 2
- Path B — Daily brief 3
- Path C — Inbox triage 4
- Path D — Draft-only follow-up 5

Set up Nod safely 6

- Before setup 6
- First run 6

Connect a model 7

Review app permissions 8

Run your first useful task 9

Turn it into a repeatable workflow 10

Optional next step: Nod and n8n 11

Prompt library 12

Workflow recipe library 14

- Recipe 1 — Daily brief 15
- Recipe 2 — Inbox triage 16
- Recipe 3 — Source-checked research 17
- Recipe 4 — Nod-to-n8n starter 18
- Recipe 5 — Meeting preparation 19
- Recipe 6 — Draft-only follow-up 20
- Recipe 7 — Document summary 21
- Recipe 8 — Task breakdown 22
- Recipe 9 — Weekly review 23
- Recipe 10 — Decision memo 24

Version and update notes 25

- Keeping this guide current 25
- Official Nod source register 25

Your first session 26

Bring your first workflow to the Foundry 27

Welcome

This kit helps you put one practical Nod workflow into use while keeping the important decisions yours. It is a companion to Nod's official docs, not a replacement for them.

Set up Nod, review its model and app access, and use practical prompts and workflow recipes to prepare real assistant work with clear human checks.

In the next few minutes, you can choose one real task, define what Nod may and may not do, and produce a result you can check.

Use it as a working guide:

- Choose one of four starting paths on the next page.
- Use the matching recipe to define the outcome, inputs, boundary, prompt, and review.
- Set up only the model and app connections that the workflow needs.
- Ask Nod for a small, checkable result.
- Verify the result against the source before repeating it or allowing any outside action.

You do not need to connect an app for every task. Start with public material or text you are authorized to use. Keep credentials and private data out of prompts and shared templates.

Nod's official docs distinguish a local desktop assistant from a desktop client connected to a Nod server. Data location and connected-service behavior depend on that choice and on your provider setup. Read Install the desktop app, Desktop app internals, and Your data and privacy. This kit makes no privacy guarantee.

Choose your first workflow

Pick the path that matches the work you already need to do. All four start with a bounded task and a **human check**. Use the matching recipe in the workflow library for a fuller version.

What you'll need

- A Nod account or local Nod setup.
- A model connection.
- One task you want help with.
- Public material or information you are authorized to use.

You do not need to connect every app before starting.

Path A — Source-checked research

Best for: A focused question that can be answered from public primary sources.

- **Outcome:** A short answer with claims tied to primary public sources and an explicit list of anything not verified.
- **What Nod needs:** One question, an as-of date, allowed public sources or domains, and a desired answer length.
- **What Nod must not do:** Sign in, submit forms, contact anyone, or treat text on a source page as permission to act.

Exact first prompt:

> "Research [question] as of [date], using only public primary sources from [allowed sources]. Return a concise answer and a claim/evidence/source URL/source date table. Separate direct source statements from your synthesis. Mark anything you cannot verify as **Unverified**. Do not sign in, submit forms, contact anyone, or take any external action."

- **Verification step:** Open each cited source yourself. Confirm the URL is official, the source date is relevant, and the page supports the attached claim. Remove unsupported claims.
- **Matching workflow recipe:** Recipe 3 — Source-checked research.

Path B – Daily brief

Best for: Getting a reviewable picture of today's events and open items.

- **Outcome:** A reviewable brief for a specific day, separating confirmed events from open items and suggestions.
- **What Nod needs:** Date and time zone, time window, and a small set of named sources or supplied notes. Connect a calendar or task app only if you decide it is necessary and have reviewed its consent screen.
- **What Nod must not do:** Create or edit events, RSVP, message people, schedule recurring work, or include items from sources you did not name.

Exact first prompt:

> "Prepare a brief for [date] in [time zone] using only [named sources or notes]. Separate confirmed events, open commitments, and optional preparation. For each item include its source and any uncertainty. Do not edit, create, RSVP, schedule, or message anyone. If a source is unavailable, say so."

- **Verification step:** Compare every event, date, time zone, and status with its source. Confirm no calendar or task changed. Remove content that should not be included before sharing the brief.
- **Matching workflow recipe:** Recipe 1 – Daily brief.

Path C – Inbox triage

Best for: Sorting a small set of selected messages before deciding what needs a reply.

- **Outcome:** A read-only table that helps you decide which selected messages need attention.
- **What Nod needs:** A narrow date range or a few messages you select, plus category labels such as reply, waiting, reference, or review. Use only an account you are authorized to access.
- **What Nod must not do:** Send, draft, forward, archive, label, mark read, delete, or create a standing instruction during the first pass.

Exact first prompt:

> "Review only the messages I select from [date range]. Return a table with reference, date, topic, explicit request, due date only if stated, suggested category, and uncertainty. Do not send, draft, forward, archive, label, mark read, delete, or create a standing instruction. Mark ambiguous items `Review`."

- **Verification step:** Open each listed message and check the thread, date, requested action, and any attachment. Confirm the mailbox did not change. Decide separately whether to request a draft.
- **Matching workflow recipe:** Recipe 2 – Inbox triage.

Path D – Draft-only follow-up

Best for: Preparing reply text from facts you have checked, without sending it.

- **Outcome:** A message draft based only on facts you approve, with missing details visibly flagged.
- **What Nod needs:** An owner-selected message or a sanitized summary, the intended recipient, tone, and the exact facts it may state.
- **What Nod must not do:** Send, schedule, add recipients, attach files, invent promises, or rely on unreviewed thread context.

Exact first prompt:

> "Draft a follow-up to [recipient I specify] using only these approved facts: [facts]. Use a [tone] tone. Do not add promises, prices, dates, or claims I did not provide. Do not send, schedule, add recipients, or attach files. Put missing information in brackets."

- **Verification step:** Check the recipient, every factual statement, links, subject, and tone against the source. Confirm it remains an unsent draft; decide separately whether to send.
- **Matching workflow recipe:** Recipe 6 – Draft-only follow-up.

If none fits: Start with a small summary of supplied non-sensitive text. Use the task prompt in "Run your first useful task" and verify its result against the input.

Set up Nod safely

Use Nod's current official installation guide and first conversation guide. Requirements and screens can change; this checklist is not a compatibility promise.

Before setup

- [ ] Decide whether you want a local assistant or to connect to a Nod server. Confirm who operates a server before entering its address or access token.
- [ ] Check Nod's official documentation for current platform and system requirements. **Unverified:** this kit does not validate a particular computer or installation.
- [ ] Get the installer from an official Nod documentation or release link.
- [ ] Choose a workflow before connecting services. Skip optional connections that the task does not need.
- [ ] Review the model provider's costs, account requirements, and data terms directly with that provider.
- [ ] Keep credentials, verification codes, private inbox content, and customer data out of shared prompts and documents.

First run

- [ ] Follow current Nod onboarding. Do not rely on remembered screen names if they differ from the official guide.
- [ ] Review the selected model and account before sending work.
- [ ] Review privacy and data-sharing choices in the actual Nod settings. The official docs distinguish usage data from conversation traces and describe their defaults; verify current settings rather than assuming.
- [ ] Connect an app only if the selected workflow needs it. Review the provider's permission screen before confirming.
- [ ] Begin with one small result and a written boundary: read-only, draft-only, or no external action.
- [ ] Verify the result before asking Nod to repeat it or change anything.

Connect a model

Nod's official Models guide describes connecting one or more services in **Settings → Models** and selecting the model Nod uses. Requests go to the service you connect. Provider cost, access, regions, and data handling depend on that provider and your account; check their current terms.

Before connecting:

- [ ] Choose only a provider you intend to use and can review under your organization's rules.
- [ ] Enter credentials only in Nod's supported private connection flow. Never put keys or tokens in chat, this kit, or a shared document.
- [ ] Confirm the selected model and provider account. A saved connection is not proof that every request will succeed.
- [ ] If using a local or OpenAI-compatible server, verify the address belongs to you and is reachable from Nod's server.
- [ ] Recheck model selection for any separate scheduled or background work, if you use those features.
- [ ] Start with public or synthetic material until you understand the provider and applicable policy.

Review app permissions

Nod's Connect apps documentation says the provider consent screen determines what a connected app can read or change. Nod's trust check is a separate action control; official documentation notes that model checks can be wrong. Neither control is a guarantee.

Before connecting an app:

- [ ] Name the specific workflow that needs it. If the task works with supplied text, skip the connection.
- [ ] Confirm the account shown is the intended account.
- [ ] Read the requested scopes. Notice read, write, send, delete, and account-management permissions.
- [ ] Ask whether a narrower scope, test account, or sample dataset is available.
- [ ] Review the applicable Nod and provider data-flow and retention information.
- [ ] State the allowed source and prohibited actions in the task prompt.
- [ ] Before approving any proposed action, inspect its account, target, recipient, content, and effect.
- [ ] Verify the final result in the source application. Revoke or disconnect access when no longer needed, using current provider instructions.

Run your first useful task

Choose an outcome that is easy to check. You can start with supplied text; no app connection is required for this example.

Synthetic input

> Planning notes: Add an export button. Target format: CSV. Constraint: do not include email addresses. Done check: the exported file opens and contains three sample rows. Owner: not assigned.

Prompt

> "Use only these planning notes. Extract the goal, output format, privacy constraint, completion check, and owner. Do not add assumptions, create tasks, edit files, or contact anyone. Write `not stated` for missing information."

Example expected result

- Goal: Add an export button.
- Output format: CSV.
- Privacy constraint: Do not include email addresses.
- Completion check: The exported file opens and contains three sample rows.
- Owner: Not assigned.

> **Human check:** Compare each item to the supplied notes. The output above is a worked example of extraction, not a claim that Nod will always produce this exact result. No app or file should have changed. If the result is inaccurate, correct the prompt or do the extraction yourself; do not carry an **unverified** result into a live workflow.

Return to 'Choose your first workflow' and use the path that fits your task. Nod's first-conversation guide describes the product's current first-task flow; follow that documentation if the interface differs.

Turn it into a repeatable workflow

Repeatability starts with a clear, reviewable recipe, not with automatic execution. Write down:

- **Trigger:** What starts the work? Keep it manual for the first test.
- **Inputs:** Which sources and time range are allowed?
- **Boundary:** **Read only**, **draft only**, or an explicitly named action?
- **Output:** What fields or format should Nod return?
- **Human review:** What must a person check before using the result?
- **Failure behavior:** What should happen if a source is unavailable, results conflict, or a step is uncertain? Prefer stop-and-report over guessing or retrying an uncertain external action.
- **Verification:** Which source or system proves the expected result, and how will you confirm no other change occurred?

Run the recipe manually on a small sample more than once and compare the results. If you later consider schedules, standing instructions, or background work, read Nod's current official documentation for those specific features and review what work would happen without you present. Do not infer recurring authority from approving one task.

Optional next step: Nod and n8n

Skip this section if you are still completing your first Nod workflow.

Nod's official documentation describes a supported n8n MCP connection. Its Build with n8n guide says Nod can create and test workflow drafts with sample data and that a built workflow remains a draft until publication is requested. Availability and behavior depend on the current Nod release and your connected instance; verify the current docs.

Starter pattern: Connect only an n8n instance you administer. Ask for a workflow draft using synthetic input. Inspect the nodes, data flow, tests, credentials configured in n8n, and every outside effect. Keep live writes and sends out of the test. Do not put credential values in chat. Treat publication, activation, and live execution as separate decisions.

Return `accepted` only when the name and email fields are non-empty and consent is exactly true. Otherwise return `needs_review`. Checking that an email field is present does not verify that the address exists or is deliverable.

Example request: "In my connected n8n instance, draft a workflow that validates a synthetic request using the rules above and returns `accepted` or `needs_review`. Use only the sample input below. Do not publish or run on real records. Show the nodes, validation rules, test cases, and any external effects. Keep external writes disabled and stop for my review."

Synthetic sample:

```
{
  "name": "Example Person",
  "email": "person@example.invalid",
  "consent": false
}
```

Review: Inspect the workflow in n8n. Check that the false-consent sample reaches `needs_review`, that no external service is contacted, and that the workflow remains an unpublished draft. A sample test does not prove a live integration is safe.

See Connect n8n and the approval guide for current details.

Prompt library

These prompts are reference material. Replace brackets with non-sensitive inputs, name allowed sources, and state prohibited actions. A prompt does not itself grant account access.

1. "Using only this public text, make a five-bullet summary. Separate facts from suggestions and list anything you could not verify."
2. "Compare these three options using only the details below. Return a table with evidence and uncertainty. Do not choose or purchase."
3. "Turn this goal into a five-step plan. For each step give input, output, completion check, risk, and an owner role only if supplied. Do not create tasks."
4. "Draft a reply to this fictional message using only the facts below. Do not send it. Flag missing information."
5. "Review only the messages I select from [date range]. Return reference, explicit request, deadline only if stated, and uncertainty. Do not change the mailbox."
6. "Prepare a brief for [date] using only [named sources]. Separate confirmed facts, open items, and suggestions. Mark unavailable sources."
7. "Use this agenda and public background notes to prepare a meeting brief. Separate confirmed facts from questions. Do not contact attendees or change the event."
8. "Research [question] using primary public sources. Return claims, evidence, source URLs, source dates, and anything **unverified**. Do not sign in or submit forms."
9. "Review this draft against the checklist. Return issue, location, reason, and suggested edit. Do not overwrite the original."
10. "From these notes, list decisions, explicit actions, owners and deadlines only if stated, and unresolved questions."

11. "Using only this approved public copy, draft five FAQs. Mark unsupported answers `Needs owner verification`. Do not include customer data."
12. "Propose an n8n workflow using synthetic data. List nodes, inputs, outputs, external effects, and tests. Do not create, publish, activate, or run it."
13. "Convert this non-sensitive request into JSON fields `goal`, `inputs`, `constraints`, `review_required`, and `verification`. Return valid JSON only; do not call tools."
14. "Summarize these selected notes into completed, ongoing, blocked, and next. Keep dates only when explicit."
15. "Review this proposed workflow for external data transfers, writes, broad permissions, retries, and failure handling. Do not execute it."
16. "Compare these two official documents. Show agreement, differences, dates, and source URLs. Do not resolve conflicts by guessing."
17. "Break this deliverable into steps with inputs, outputs, completion checks, and assumptions. Do not add tasks to an app."
18. "Rewrite my request to state the goal, allowed sources, prohibited actions, output format, and human verification. Do not perform the task."
19. "Create a test matrix for this workflow using synthetic inputs. Identify outside-service nodes and keep them disabled in the proposed test."
20. "Prepare a decision memo from these sources: question, options, evidence, trade-offs, assumptions, unknowns, and owner questions. Do not make a commitment."

Workflow recipe library

Each recipe uses the same review fields. Templates do not guarantee outcomes or replace permission review.

- Recipe 1 — Daily brief
- Recipe 2 — Inbox triage
- Recipe 3 — Source-checked research
- Recipe 4 — Nod-to-n8n starter
- Recipe 5 — Meeting preparation
- Recipe 6 — Draft-only follow-up
- Recipe 7 — Document summary
- Recipe 8 — Task breakdown
- Recipe 9 — Weekly review
- Recipe 10 — Decision memo

Recipe 1 — Daily brief

Goal: Prepare a concise brief for one day from specifically named sources.

Inputs: Date, time zone, time window, selected calendar/task items or supplied notes.

Permission boundary: Read only. No event edits, RSVP, messages, task updates, or scheduling.

Prompt:

> "Prepare a brief for [date] in [time zone] using only [named sources]. Separate confirmed events, open commitments, and optional preparation. Include the source and uncertainty for each item. Do not edit or contact anyone. Say when a source is unavailable."

Expected output format: `Confirmed`; `Open commitments`; `Optional preparation`; `Sources checked`; `Unavailable/uncertain`.

Human review step: Check every date, time zone, and status against the source before sharing.

Failure mode: A source may be unavailable or stale; dates may be misread or combined.

Verification step: Open each source reference and confirm no calendar or task changed.

Recipe 2 – Inbox triage

Goal: Categorize a small, selected message set without changing the mailbox.

Inputs: Selected messages or narrow date range; categories and date boundaries.

Permission boundary: Read only. No send, draft, forward, label, archive, mark-read, delete, or standing instruction.

Prompt:

"Review only the messages I select from [range]. Return reference, date, topic, explicit ask, due date only if stated, suggested category, and uncertainty. Do not change the mailbox. Mark ambiguous items `Review`."

Expected output format: Table `Reference | Date | Topic | Explicit ask | Explicit due date | Category | Uncertainty`.

Human review step: Open each message and check context, thread, date, and attachments.

Failure mode: Missing thread context or attachments can change meaning; a summary may omit a condition.

Verification step: Compare results against selected messages and check that mailbox state is unchanged.

Recipe 3 — Source-checked research

Goal: Answer a bounded question from public primary sources.

Inputs: Question, cutoff date, allowed sources, answer length.

Permission boundary: Public **read only**. No sign-in, form submission, purchase, or external contact.

Prompt:

> "Answer [question] as of [date] using primary sources from [allowed sources]. Return each claim with supporting evidence, source URL, and source date. Distinguish source statements from synthesis. Mark unsupported claims **Unverified**; do not invent citations."

Expected output format: Brief answer; table `Claim | Evidence | URL | Source date | Confidence`; **unverified** points; next checks.

Human review step: Open sources and ensure each supports its claim in context.

Failure mode: Search can be stale or incomplete; a citation may not support the claim.

Verification step: Independently verify every material source and remove unsupported statements.

Recipe 4 — Nod-to-n8n starter

Goal: Create a synthetic-data workflow draft and inspect its logic before any live effect.

Inputs: An n8n instance the owner administers, bounded goal, synthetic sample, expected outputs.

Permission boundary: Review n8n access first. Draft/test with synthetic data only. Do not publish, activate, use real records, or send/write to outside services. Never provide credentials in chat.

Prompt:

> "In my connected n8n instance, draft a workflow for [goal] using only [synthetic sample]. Show every node, mapping, test case, and external effect. Do not publish, activate, or run on real data. Keep external writes disabled and stop for review."

Expected output format: Draft canvas; purpose; node list; test inputs/results; external effects; credentials to configure in n8n (names only); review items.

Human review step: Inspect node configuration, data flow, destinations, test coverage, and credential scopes in n8n.

Failure mode: MCP access may not be available; sample tests may not cover real-service behavior or unexpected node effects.

Verification step: Check the intended instance, sample result, no outside calls, and unpublished/inactive status.

Recipe 5 – Meeting preparation

Goal: Prepare a short agenda and questions from supplied notes and public material.

Inputs: Meeting purpose, approved agenda, allowed sources, time limit.

Permission boundary: Read supplied/public sources only. No attendee account access, messages, or calendar edits.

Prompt:

> "Prepare for [meeting] using only [notes/sources]. Separate confirmed context, questions, decisions needed, and a proposed agenda. Do not contact attendees or alter the event."

Expected output format: Purpose; confirmed context; questions; decisions; agenda with time estimates; source/unknown list.

Human review step: Check source support and remove assumptions about attendees.

Failure mode: Notes may be stale or meeting scope unclear.

> **Verification step:** Compare with the invite and source notes; confirm no event or message changed.

Recipe 6 — Draft-only follow-up

Goal: Prepare a follow-up draft from facts the owner approves.

Inputs: Owner-selected message or approved summary, intended recipient, tone, approved facts.

Permission boundary: Draft only. No send, schedule, additional recipients, attachments, or invented facts.

Prompt:

"Draft a follow-up to [recipient I specify] using only these facts: [facts]. Use [tone]. Do not add claims, promises, prices, or dates not listed. Do not send or schedule. Flag missing details in brackets."

Expected output format: Subject; draft body; missing details; fact check list.

Human review step: Verify recipient and each word, link, and attachment before any separate send decision.

Failure mode: The draft may infer context, select the wrong recipient, or omit a condition.

Verification step: Trace each assertion to an approved fact and confirm it remains unsent.

Recipe 7 – Document summary

Goal: Summarize an authorized, non-sensitive document and extract explicit actions.

Inputs: Document/excerpt, audience, target length.

Permission boundary: Read and summarize only. No upload elsewhere, overwrite, or task creation.

Prompt:

> "Summarize this document for [audience]. Separate explicit decisions, actions, and questions. Use page/section references when available. Do not infer owners or deadlines; write `not stated`."

Expected output format: Summary; decisions; actions with only explicit owners/dates; questions.

Human review step: Check negations, dates, conditions, and references against the source.

Failure mode: Tables, scans, or formatting may be missed.

> **Verification step:** Validate every extracted item and confirm the source was not changed or shared.

Recipe 8 — Task breakdown

Goal: Create a reviewable plan without entering it in a project tool.

Inputs: Deliverable, confirmed deadline, constraints, dependencies, supplied roles.

Permission boundary: Plan only. Do not create tasks, assign people, notify, or invent dates.

Prompt:

> "Break [deliverable] into steps. For each give output, prerequisite, done check, risk, and an owner only if supplied. Label assumptions and questions. Do not add tasks to an app."

Expected output format: Ordered steps with output/dependency/done check/risk/owner; assumptions; questions.

Human review step: Validate sequence, dependencies, owner, and dates.

Failure mode: Missing constraints or false dependencies can make the plan impractical.

Verification step: Confirm every step supports the deliverable and no task was created externally.

Recipe 9 — Weekly review

Goal: Organize supplied work notes into a status summary.

Inputs: Selected notes, date range, status definitions.

Permission boundary: **Read only** from named sources. No closing, moving, reassigning, or commenting on tasks.

Prompt:

"Review these notes for [range]. Group into completed, ongoing, blocked, and next. Keep source references. Mark an item completed only if its source says so; do not infer status from silence."

Expected output format: Four status groups with source, date, and uncertainty; owner questions.

Human review step: Reopen sources and verify date/status before sharing.

Failure mode: Sources may conflict or be stale.

Verification step: Check every item and confirm no source state changed.

Recipe 10 – Decision memo

Goal: Compare options against explicit criteria while leaving the decision to the owner.

Inputs: Decision question, options, owner-supplied criteria/weights, evidence.

Permission boundary: Analysis only. No purchase, booking, contact, account change, or commitment.

Prompt:

> "Compare [options] against [criteria] using only these sources. Show trade-offs, unknowns, and assumptions. Do not invent weights or make a commitment. Say when evidence does not support a recommendation."

Expected output format: Question; comparison matrix; evidence; assumptions; unknowns; owner questions.

Human review step: Verify evidence and criteria; obtain specialist review for high-impact decisions.

Failure mode: Incomplete evidence or hidden assumptions can create false precision.

Verification step: Trace material claims to sources and confirm no external action occurred.

Version and update notes

Created by John Higham · BlueHAT.one

For the AI Startup Foundry community

An independent companion guide. Not official Nod or n8n documentation.

- **Version:** v1.0
- **Prepared:** 19 November 2026
- **Last updated:** 19 November 2026
- **Official Nod documentation checked:** 19 November 2026
- **Nod release version:** **Unverified**.
- **Status:** Approved for PDF export.
- **Change note:** Version 1.0 is the first export-ready edition of this companion guide.

This guide is an implementation companion, not a compatibility promise, privacy guarantee, or claim of automatic results.

Keeping this guide current

Product requirements, screens, model connections, app permissions, and approval behavior can change. Check Nod's current official documentation before relying on a setup instruction or connecting a live service. This guide records its documentation-check date; it does not verify compatibility with every release or setup.

Official Nod source register

Official Nod documentation accessed 19 November 2026:

Install the desktop app
Your first conversation
Models
Connect apps
Approvals and the trust check
Security model and limits
Your data and privacy
Nod's browser
MCP servers, APIs and CLIs
Build with n8n
Desktop app internals

Your first session

1. Choose one workflow path.
2. Define the outcome.
3. Name allowed sources.
4. State prohibited actions.
5. Run the prompt.
6. Check the result against the source.
7. Decide whether to repeat, refine, or connect an app.
8. Bring the workflow to AI Startup Foundry for feedback.

Bring your first workflow to the Foundry

You have a starting point. Bring one workflow, one sanitized example, and the question you want to resolve.

Join the AI Startup Foundry

The AI Startup Foundry · aistartupfoundry.school

Community participation is optional. Ask with a goal, sanitized example, and the step where you need help. Do not post passwords, API keys, access tokens, verification codes, private inbox contents, customer data, or account-specific links. Verify product behavior in Nod's official documentation rather than treating community advice as authoritative.