Automate Digital
For automators

Beginner Guide

automatedigital.ai

Self-Hosted n8n

A clean, practical beginner guide to building reliable workflows with n8n.

YOU'LL LEARN

- How n8n workflows work (nodes, triggers, items)
- Expressions + data mapping that unlock real automations
- Debugging, executions, and error handling for reliability

BEST FOR

- Beginners who want a repeatable build/test method
- Teams who care about data ownership
- Automations that need branching and data shaping

Contents

Hosting first: Deploy n8n on Hostinger (recommended)

This guide is designed to help you go from **zero → running n8n** quickly, with a setup that scales without surprise bills.

From our experience, most beginners get the best results with a managed VPS-style setup because you get:

- **Predictable fixed cost** (no per-execution pricing surprises as you scale)
- **More control** (your environment, data, and the ability to customize)
- **Better privacy** (your data stays on infrastructure you control)

If you want to self-host via Hostinger, you can start here:

- `https://automatedigital.ai/n8n`

Hostinger setup walkthrough (based on our video)

This is the “from landing page → running n8n instance” flow shown in `resources/n8n_video-1_callum.txt`, translated into a checklist you can follow:

1. **Open the Hostinger n8n setup page** and choose a plan.
2. **Review hardware** before you click buy:
   - **vCPU + RAM** drive how many workflows can run at the same time and how heavy they can be.
   - **NVMe disk** holds your n8n data and helps performance.
   - **Bandwidth** matters when you’re calling lots of APIs or moving larger payloads.
3. **Choose your billing cycle** (monthly vs yearly) and confirm the fixed-cost model suits your execution volume.
4. **Enable VPS backups** (recommended) so you have automated restore points.
5. **Pick a server location** aligned with latency + compliance/data sovereignty needs.
6. **Complete purchase**, then set:
   - **root password**
   - optional **SSH key**
7. **Wait for provisioning** (Hostinger sets up the server and installs the n8n app).
8. In the **server dashboard**, you should see options such as:
   - managing the app (often via a “manage app” area)
   - a **Docker manager**
   - VPS management and SSH access
9. **Add a public domain** (if you want external access).
10. **Finish n8n onboarding**:

- create your admin user
- add credentials
- optionally enter an email to receive a license key for additional features

Step 0 — Choose the right VPS size (simple rule of thumb)

Your **vCPU** and **RAM** determine how many workflows you can run at once and how complex they can be.

- **More RAM** helps if you're processing larger payloads (big JSON, files, heavy transforms).
- **More vCPU** helps if you need more concurrency (more simultaneous executions).
- **NVMe disk** stores your n8n data (database/workflows) and affects performance.
- **Bandwidth** matters when you call lots of external APIs or move data around frequently.

> **Pro tip**: Start smaller, measure, then upgrade. Most "I need a huge server" assumptions are wrong until you see actual execution volume.

Step 1 — Pick a plan + billing cycle (why self-hosting stays predictable)

Self-hosting typically works best when you want to avoid usage-based pricing. With a VPS plan you pay a **fixed monthly/annual cost**, independent of how many workflows you run.

Step 2 — Add backups (strongly recommended)

With self-hosting, **you're responsible for data recovery**. Enable automated VPS backups/snapshots if available. This gives you a safety net for:

- accidental misconfiguration
- failed updates
- data loss

Also consider: exporting your important workflows regularly (and storing them off-server).

Step 3 — Choose server location (privacy + compliance)

Pick a server location that aligns with your needs around **latency** and **data sovereignty** (where data is processed/stored).

Step 4 — Provision + secure access

After purchase/provisioning:

- Set a strong **root/admin password**
- Optionally add an **SSH key** (recommended for safer access)

Step 5 — Install n8n (Hostinger path)

On Hostinger-style managed deployments, you'll typically see a dashboard where you can:

- manage the n8n app
- access tooling like a **Docker manager**
- connect via SSH if needed

Once provisioning finishes, you should have a running n8n instance ready for onboarding.

Step 6 — Add a domain + HTTPS

For real use (webhooks, team access), add a public domain and ensure HTTPS is enabled. At a high level:

- point DNS to your server
- enable SSL
- confirm your n8n base URL/webhook URL settings are correct for production triggers

Step 7 — First login + onboarding

When you first open n8n:

- create your **owner/admin user**
- add **credentials** for the apps you'll connect (Google, Slack, Airtable, etc.)
- complete any onboarding steps (templates/survey)

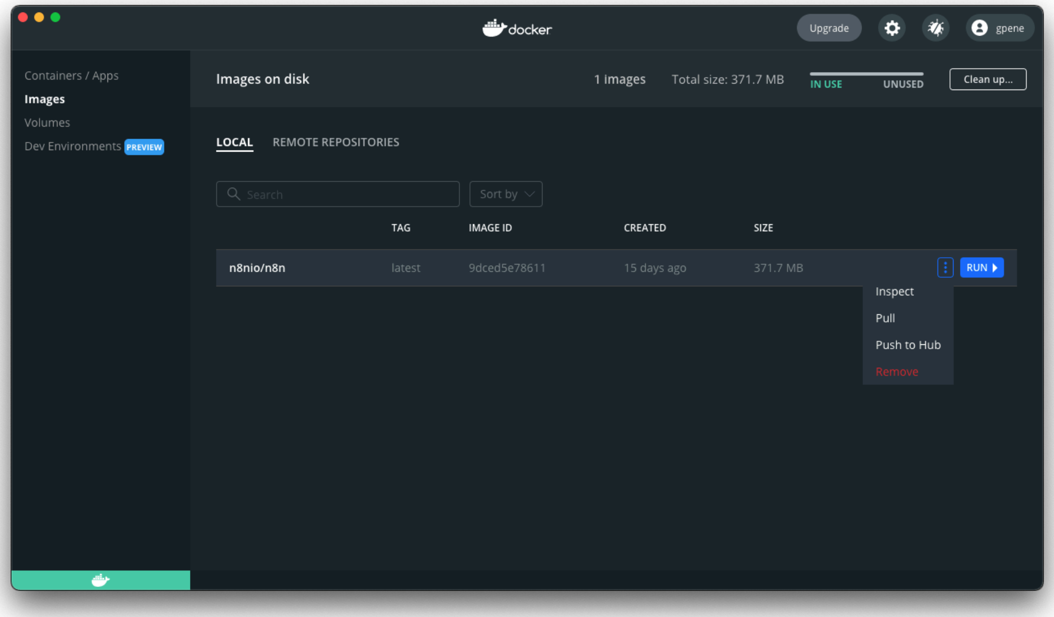
Some setups let you enter an email to receive a **license key** to unlock additional features.

Step 8 — "Production-ready" checklist (do this early)

- **Timezone** set correctly (instance and/or workflow) so schedules run when you expect
- **Backups** enabled and tested (know how you'd restore)
- **Error workflow** configured so failures notify you (Slack/email)
- **Updates**: have a plan for upgrading safely (and rolling back)

Alternative: Run n8n with Docker (official docs)

n8n recommends Docker for most self-hosting needs. This is a good option if you're comfortable managing containers.



	TAG	IMAGE ID	CREATED	SIZE
n8nio/n8n	latest	9dced5e78611	15 days ago	371.7 MB

Screenshot from n8n Docs: https://docs.n8n.io/hosting/installation/docker/

Reference:

- Docker install docs: `https://docs.n8n.io/hosting/installation/docker/`
- Docker Compose option: `https://docs.n8n.io/hosting/installation/server-setups/docker-compose/`

What you'll learn

- **How n8n works**: nodes, triggers, actions, logic, and how data moves.
- **How to build workflows** without getting stuck (a repeatable build/test method).
- **Expressions & data mapping** so your workflows become dynamic.
- **Credentials & security basics** so you don't leak secrets.
- **Debugging, executions, and error handling** so workflows run reliably in production.

Who this guide is for

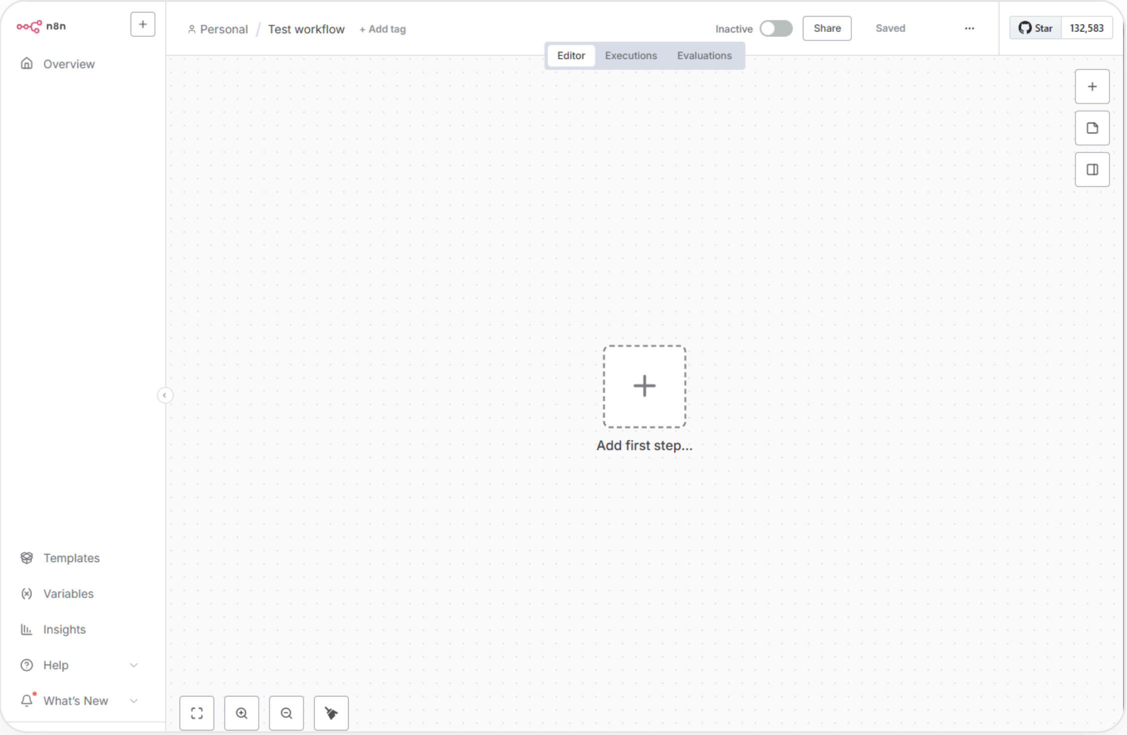
- People who want to **save time** by automating repetitive operations.
- Anyone who's tried Zapier/Make and wants **more control** (logic, branching, loops, data shaping).
- Teams who care about **data ownership** and want to understand self-hosting options.

What you need

- A running n8n instance (Cloud, local, or self-hosted).
- 60–120 minutes to complete the lessons and build your first workflows.

Quickstart: the n8n mental model (read this first)

n8n is a **visual workflow automation tool**. You build automations by connecting steps (“nodes”) on a canvas.



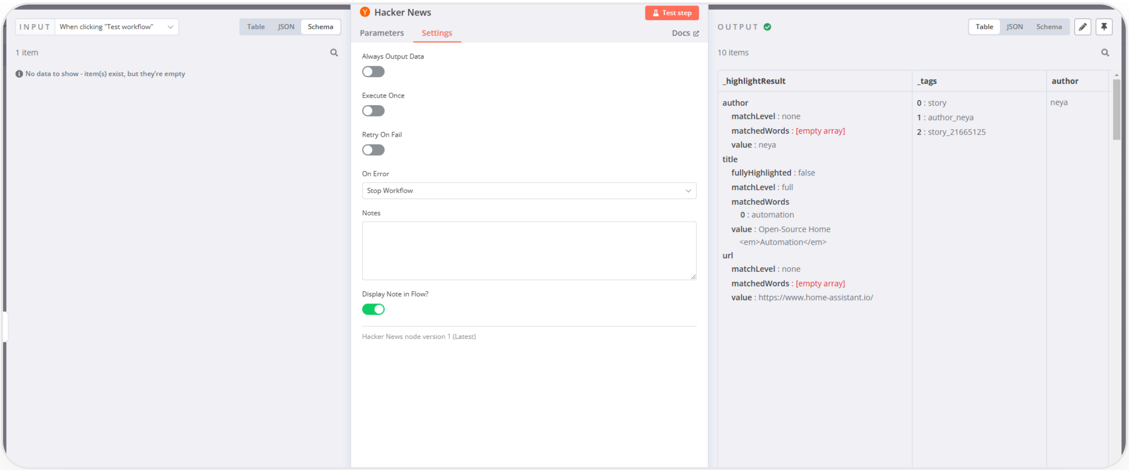
Screenshot from n8n Docs: https://docs.n8n.io/courses/level-one/chapter-1/

Most workflows follow this pattern:

1. **Trigger**: how the workflow starts (schedule, webhook, new email, etc.).
2. **Actions**: nodes that read/write data in apps (Google Sheets, Slack, Notion, etc.).
3. **Logic & transforms**: nodes that shape data, branch, loop, filter, and handle errors.
4. **Operations**: activate, monitor, debug, and improve the workflow over time.

The single biggest idea in n8n: “items”

In n8n, data moving between nodes is typically an **array of items**, where each item is an object.
If a node receives 10 items and is configured to “create something”, it usually creates **10 things** (one per item).



1 item

No data to show - item(s) exist, but they're empty

Always Output Data ○

Execute Once ○

Retry On Fail ○

On Error: Stop Workflow

Notes

Display Note in Flow? ●

Hacker News node version 1 (Latest)

10 items

_highlightResult	_tags	author
author matchLevel : none matchedWords : [empty array] value : neya title fullyHighlighted : false matchLevel : full matchedWords 0 : automation value : Open-Source Home <em>Automation</em> url matchLevel : none matchedWords : [empty array] value : https://www.home-assistant.io/	0 : story 1 : author_neya 2 : story_21665125	neya

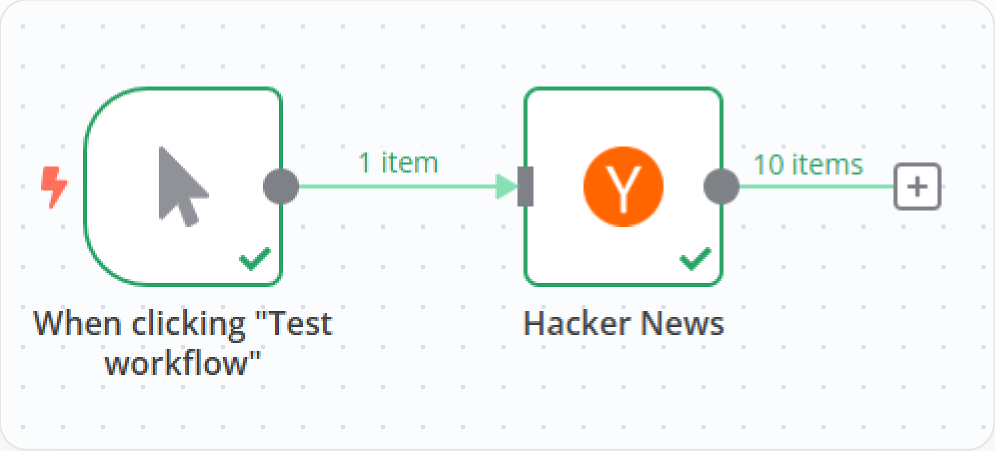
Screenshot from n8n Docs: https://docs.n8n.io/data/data-tables/

> **Pro tip**: When something "runs 10 times" unexpectedly, it's almost always because the node received 10 items.

The build method that keeps you sane

Use this loop for nearly every workflow:

- Add **one node**
- Configure it
- **Test it**
- Confirm the output is what you expect
- Then add the next node

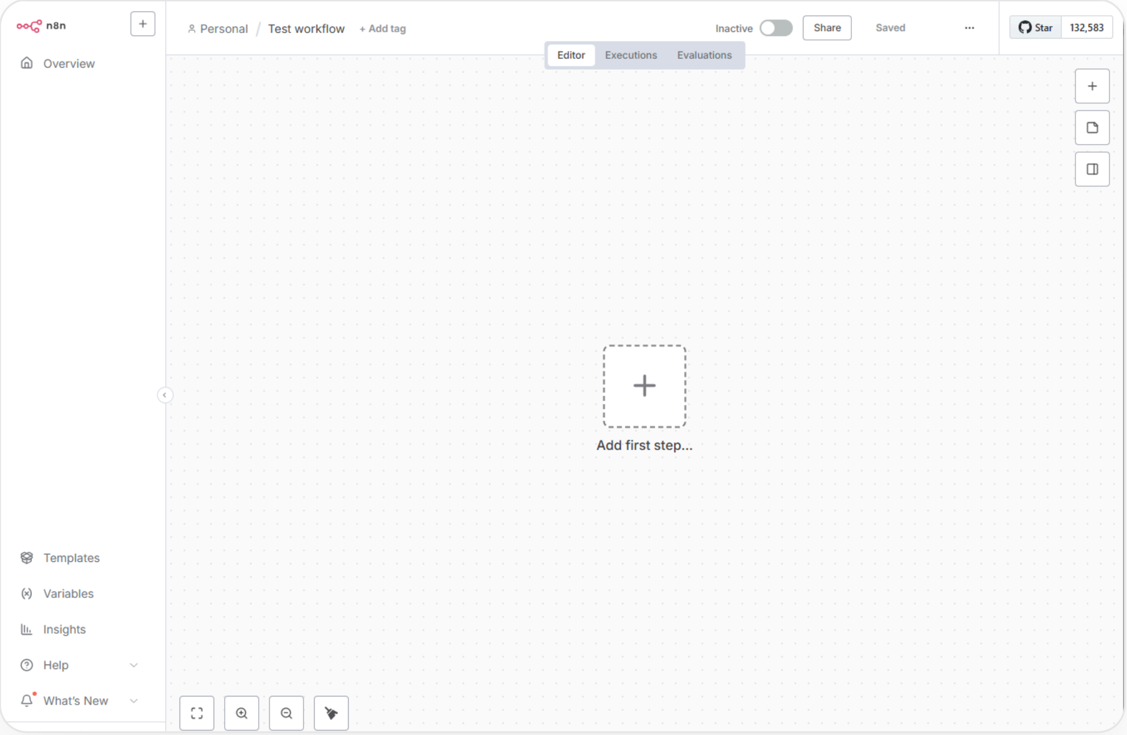


Screenshot from n8n Docs: https://docs.n8n.io/executions/

> **Common pitfall**: Building 10 nodes at once, then debugging a tangled mess.

Lesson 1 — Getting oriented (Editor UI, nodes, and navigation)

Goal: Know where everything is and how to move fast.
Outcome: You can create a workflow, find nodes, run tests, and read outputs.



Screenshot from n8n Docs: https://docs.n8n.io/courses/level-one/chapter-1/

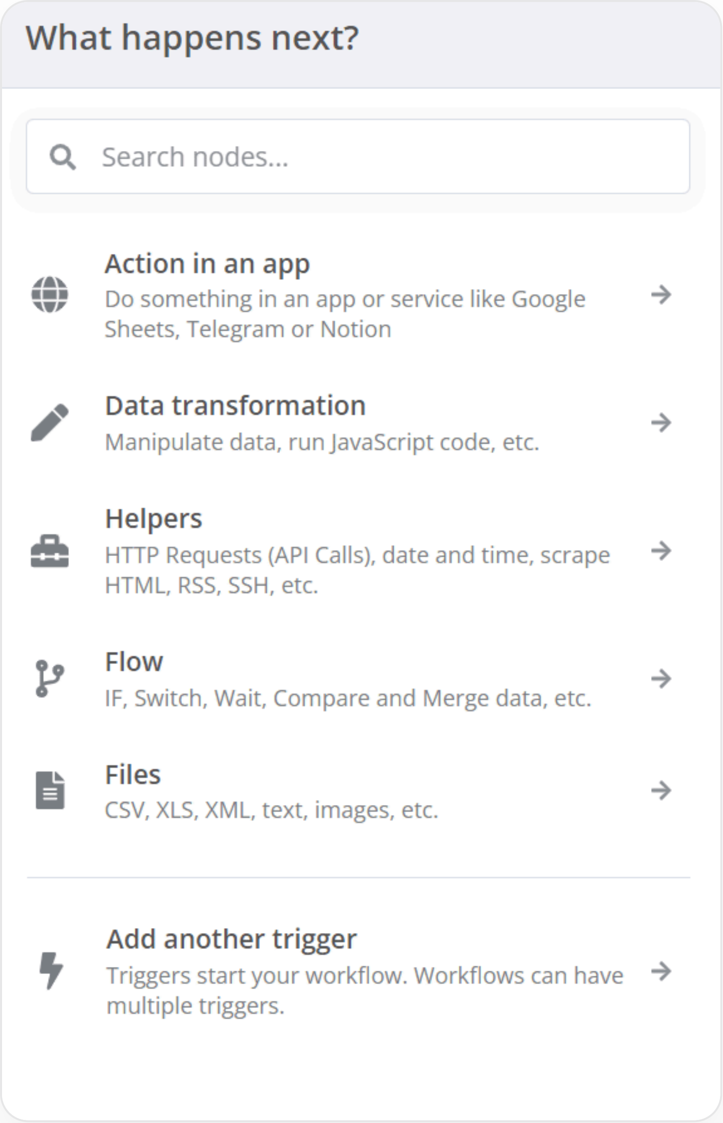
Key areas in the editor

- **Overview**: your workflows, credentials, and executions.
- **Canvas**: where you build workflows.
- **Nodes panel**: where you search/add nodes.
- **Executions**: your run history (for debugging).

Reference: n8n's "Navigating the Editor UI" lesson `https://docs.n8n.io/courses/level-one/chapter-1/`

Node types (what they mean)

- **Trigger nodes**: start the workflow (only one trigger per workflow).
- **Action/App nodes**: interact with external services (send email, create row, post message).
- **Core nodes**: generic tools & logic (IF, Merge, Set/Edit Fields, HTTP Request, Code, etc.).
- **Cluster nodes**: grouped capabilities (often AI-related).



What happens next?

Search nodes...

Action in an app
Do something in an app or service like Google Sheets, Telegram or Notion

Data transformation
Manipulate data, run JavaScript code, etc.

Helpers
HTTP Requests (API Calls), date and time, scrape HTML, RSS, SSH, etc.

Flow
IF, Switch, Wait, Compare and Merge data, etc.

Files
CSV, XLS, XML, text, images, etc.

Add another trigger
Triggers start your workflow. Workflows can have multiple triggers.

Screenshot from n8n Docs: https://docs.n8n.io/courses/level-one/chapter-1/

Keyboard shortcuts you'll actually use

Reference: `https://docs.n8n.io/keyboard-shortcuts/`

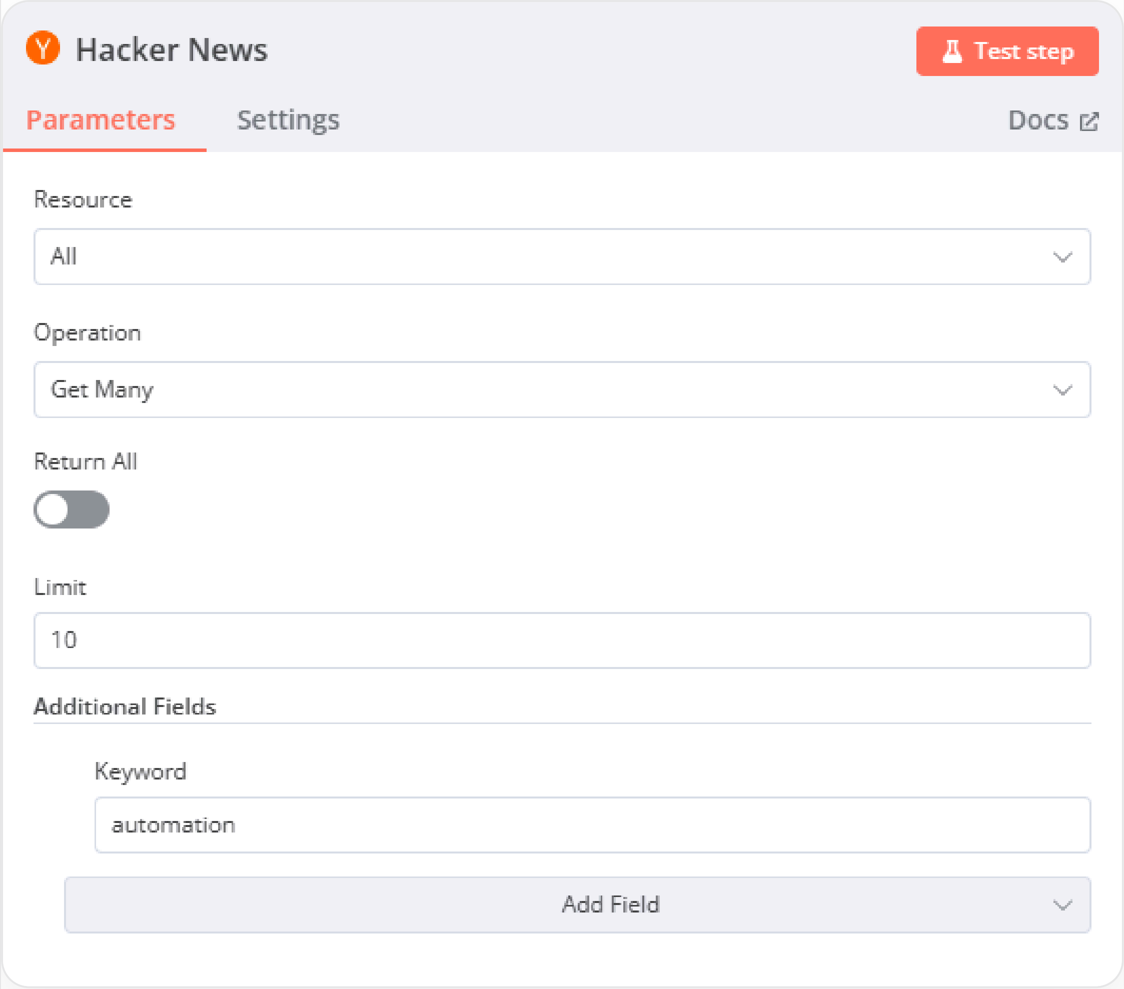
- **Save**: Ctrl/Cmd + S
- **Execute workflow**: Ctrl + Enter
- **Open command bar**: Ctrl/Cmd + K
- **Open nodes panel**: Tab
- **Pin data on a node**: P
- **Add sticky note**: Shift + S

> **Pro tip**: Use **Sticky Notes** to document "why" a workflow exists, not just "what it does."

Lesson 2 — Build your first mini-workflow (a

safe sandbox)

Goal: Learn the build/test loop on a workflow that can't break anything.
Outcome: A working workflow that creates structured output and "sends" a mock notification.



Resource: All

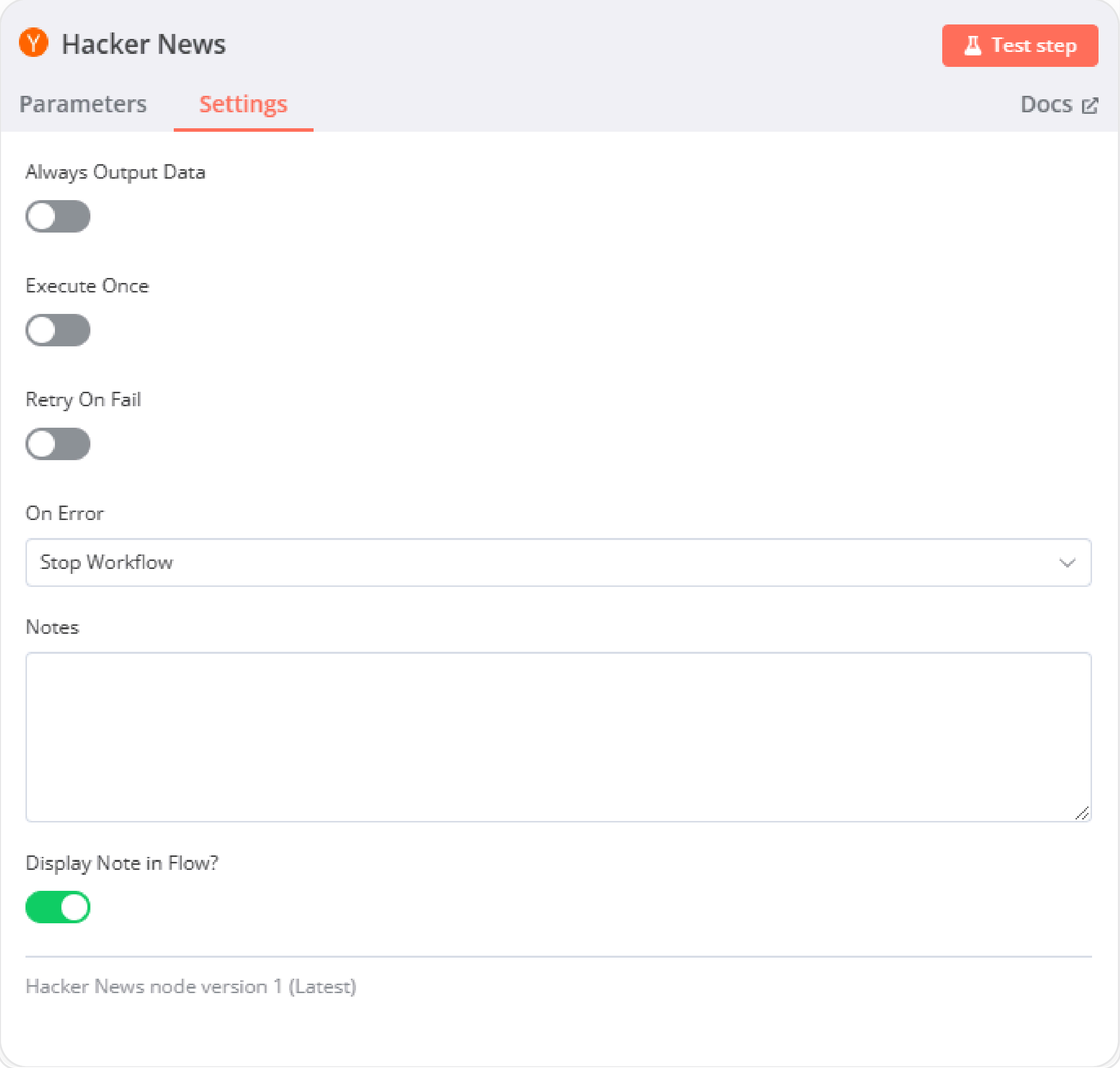
Operation: Get Many

Return All: off

Limit: 10

Additional Fields

Keyword: automation



Always Output Data ○

Execute Once ○

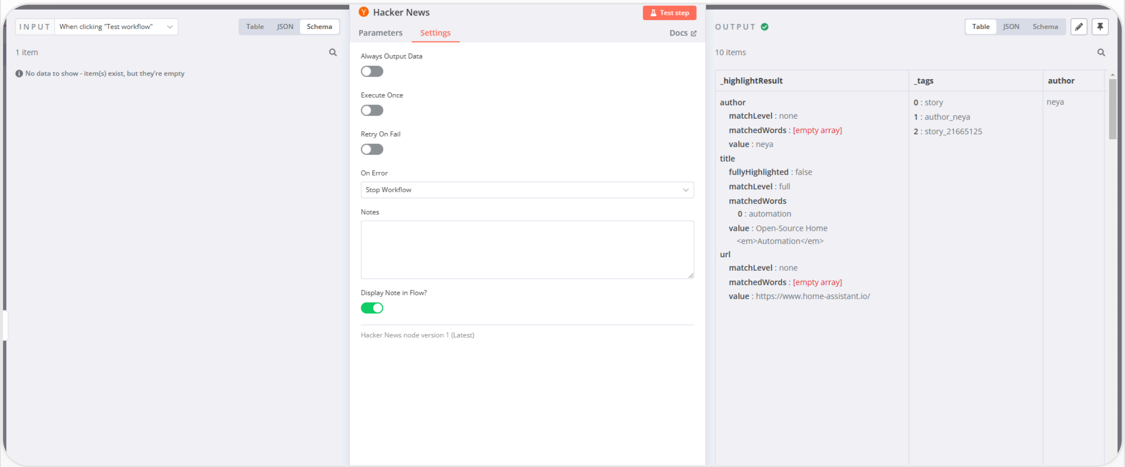
Retry On Fail ○

On Error: Stop Workflow

Notes

Display Note in Flow? ●

Hacker News node version 1 (Latest)



1 item

No data to show - item(s) exist, but they're empty

Always Output Data ○

Execute Once ○

Retry On Fail ○

On Error: Stop Workflow

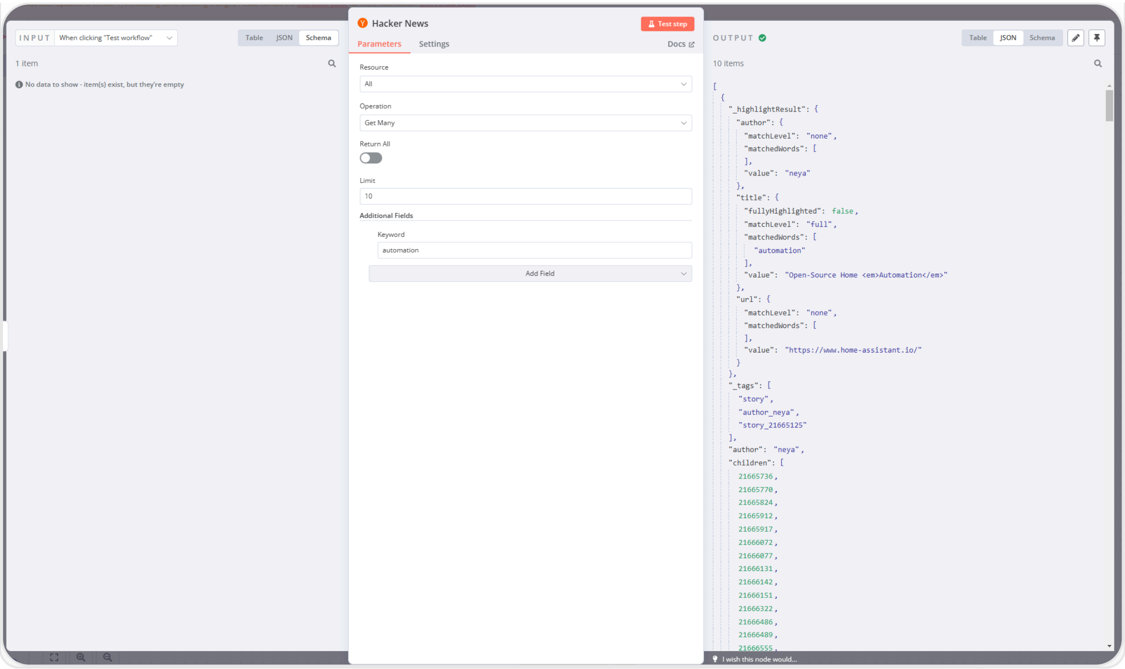
Notes

Display Note in Flow? ●

Hacker News node version 1 (Latest)

10 items

_highlightResult	_tags	author
author matchLevel : none matchedWords : [empty array] value : neya title fullyHighlighted : false matchLevel : full matchedWords 0 : automation value : Open-Source Home <em>Automation</em> url matchLevel : none matchedWords : [empty array] value : https://www.home-assistant.io/	0 : story 1 : author_neya 2 : story_21665125	neya



1 item

No data to show - item(s) exist, but they're empty

Resource: All

Operation: Get Many

Return All: ○

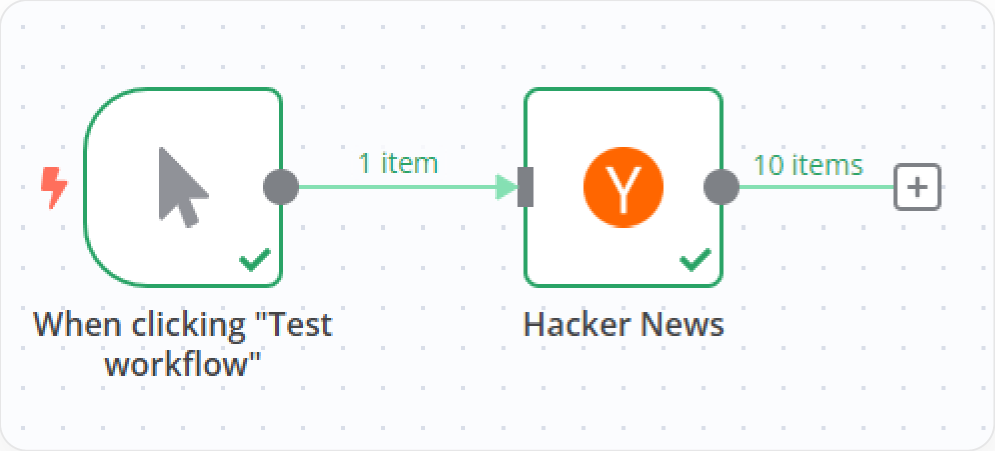
Limit: 10

Additional Fields

Keyword: automation

10 items

```
[
  {
    "_highlightResult": {
      "author": {
        "matchLevel":  "none",
        "matchedWords": [
        ],
        "value":  "neya"
      },
      "title": {
        "fullyHighlighted":  false,
        "matchLevel":  "full",
        "matchedWords": [
          "automation"
        ],
        "value":  "Open-Source Home <em>Automation</em>"
      },
      "url": {
        "matchLevel":  "none",
        "matchedWords": [
        ],
        "value":  "https://www.home-assistant.io/"
      }
    },
    "_tags": [
      "story",
      "author_neya",
      "story_21665125"
    ],
    "author":  "neya",
    "children": [
      21665736,
      21665770,
      21665824,
      21665912,
      21665917,
      21666072,
      21666077,
      21666131,
      21666142,
      21666151,
      21666322,
      21666486,
      21666489,
      21666555,
```



Screenshots from n8n Docs: https://docs.n8n.io/courses/level-one/chapter-2/

What we'll build

A basic workflow that:

1. Starts manually
2. Creates a few sample "items" (fake leads)
3. Filters them
4. Formats a message
5. (Optionally) sends to Slack/email later

Steps (high level)

1. **Manual Trigger** (or any manual start trigger)
2. **Edit Fields (Set)**: create sample data like:
    - `name`
    - `email`
    - `source`
    - `createdAt`

3. **IF**: route VIP vs non-VIP (for learning branching)
4. **Edit Fields (Set)**: build a message string
5. **No Operation** (or a placeholder action) while learning

What to pay attention to

- The output panel: **What items are produced?**
- Each node's input and output: **Are you shaping data as expected?**
- Branches: **What happens to items on the "false" path?**

> **Common pitfall**: Forgetting to connect a node after an IF branch and thinking "n8n didn't run it."

Lesson 3 — Data fundamentals (items, mapping, and why things run multiple times)

Goal: Understand the data structure so you can predict workflow behavior.
Outcome: You can confidently map data and avoid "why did this run 50 times?" confusion.

Reference: "Data structure"
`https://docs.n8n.io/key-concepts/`

The core rule

- n8n data passed between nodes is an **array of objects**.
- Many nodes process **each item**.

Item-by-item thinking

If you have input items like:

- item 1: `{ email: "a@x.com" }`
- item 2: `{ email: "b@x.com" }`

An "email send" node will usually send **two emails** unless you aggregate to a single item.

Common fixes

- **Limit**: when you only want the first N items
- **Aggregate**: when you want to summarize multiple items into one
- **Merge**: when combining data streams
- **Split Out / Loop Over Items**: when you want to process a list deliberately

> **Pro tip**: If you're integrating with rate-limited APIs, batching + aggregation is your friend.

Lesson 4 — Expressions: the feature that unlocks “real” automations

Goal: Make workflows dynamic by pulling values from previous steps.
Outcome: You can write and debug simple expressions confidently.

Reference: `https://docs.n8n.io/code/expressions/`

What expressions are

Expressions let you fill node fields using:

- output from previous nodes
- workflow metadata
- environment values
- small JavaScript snippets

Expressions use the format:

- `{{ ... }}` (double curly braces)

The most-used variable (memorize this)

- `{{$json}}` refers to the **current item**’s JSON data.

Example:

- `{{$json.body.city}}` (common when a webhook sends `{ body: { city: ... } }`)

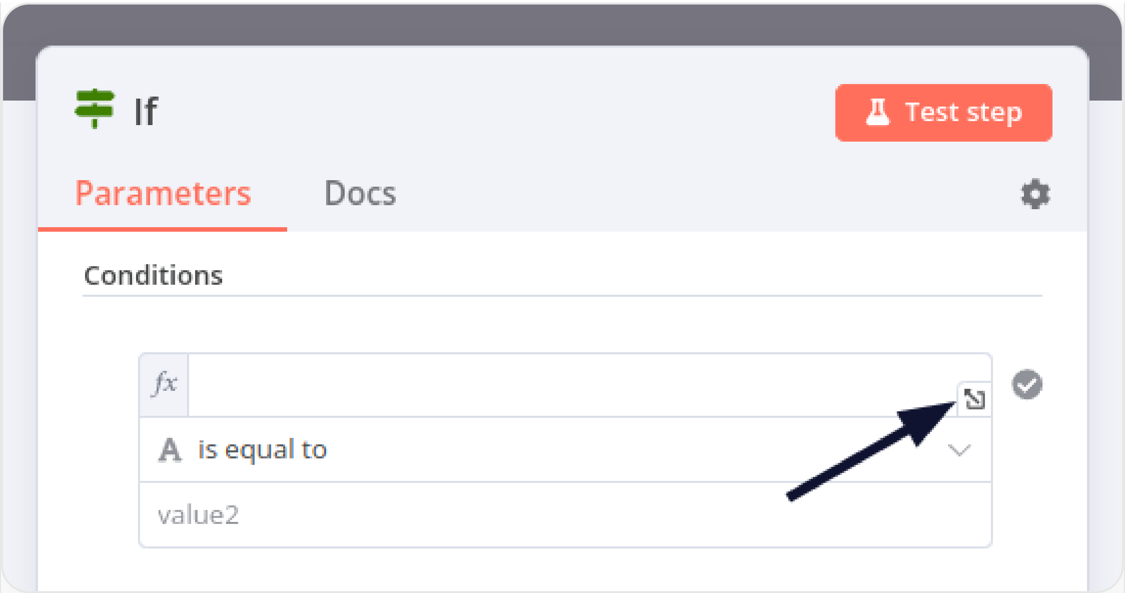
Expression patterns you’ll use often

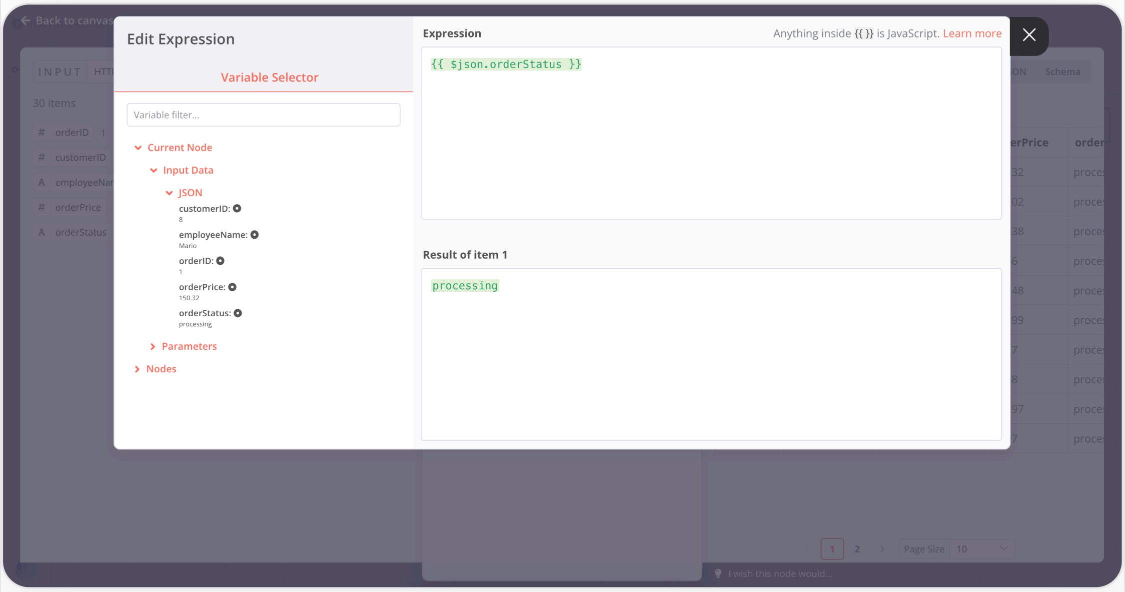
- **Build a string**: `{{ $json.firstName + " " + $json.lastName }}`
- **Fallback value**: `{{ $json.email || "missing-email" }}`
- **Choose based on condition**: `{{ $json.isVip ? "VIP" : "Standard" }}`

When to use Code node vs expressions

- **Use expressions** when you’re setting **one field** and logic is small.
- **Use Code** when you need:
  - multi-step transforms
  - loops
  - building arrays/objects
  - complex parsing/normalization

> **Common pitfall**: Writing a huge expression that becomes impossible to debug. Move it into a Code node.





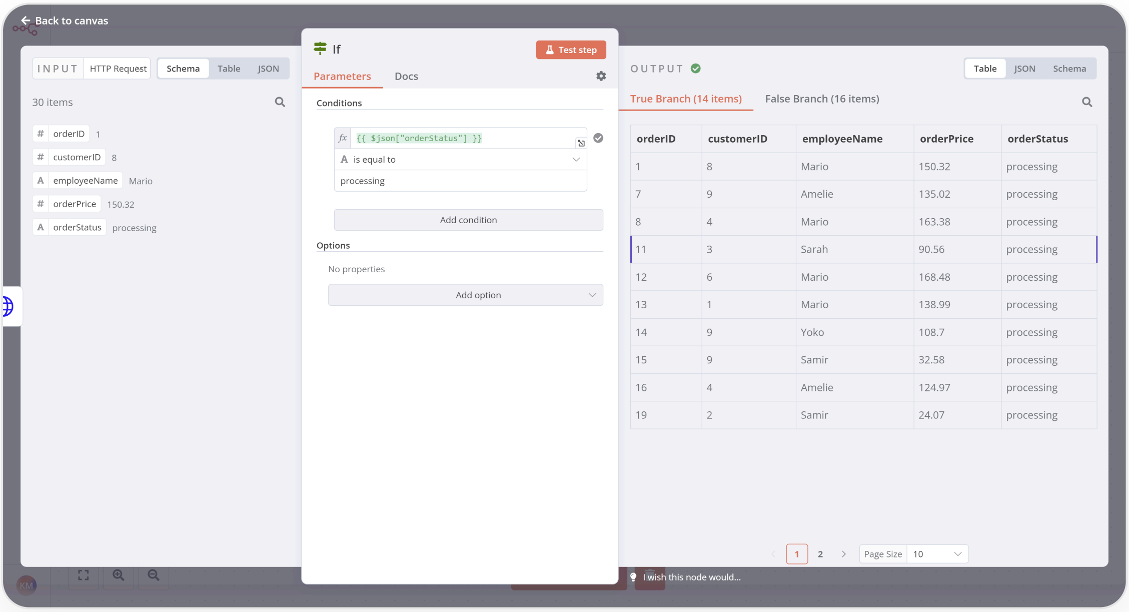
customerID: 8
employeeName: Mario
orderID: 1
orderPrice: 150.32
orderStatus: processing

Expression

```
{{ $json.orderStatus }}
```

Result of item 1

```
processing
```



#	orderID	1
#	customerID	8
A	employeeName	Mario
#	orderPrice	150.32
A	orderStatus	processing

```
{{ $json["orderStatus"] }}
```

A is equal to

processing

orderID	customerID	employeeName	orderPrice	orderStatus
1	8	Mario	150.32	processing
7	9	Amelie	135.02	processing
8	4	Mario	163.38	processing
11	3	Sarah	90.56	processing
12	6	Mario	168.48	processing
13	1	Mario	138.99	processing
14	9	Yoko	108.7	processing
15	9	Samir	32.58	processing
16	4	Amelie	124.97	processing
19	2	Samir	24.07	processing

1 2 Page Size 10

Screenshots from n8n Docs: https://docs.n8n.io/courses/level-one/chapter-5/chapter-5.3/

Lesson 5 — Credentials & security basics (don't leak secrets)

Goal: Connect services safely and make workflows shareable.
Outcome: You can create credentials, name them well, and avoid common security mistakes.

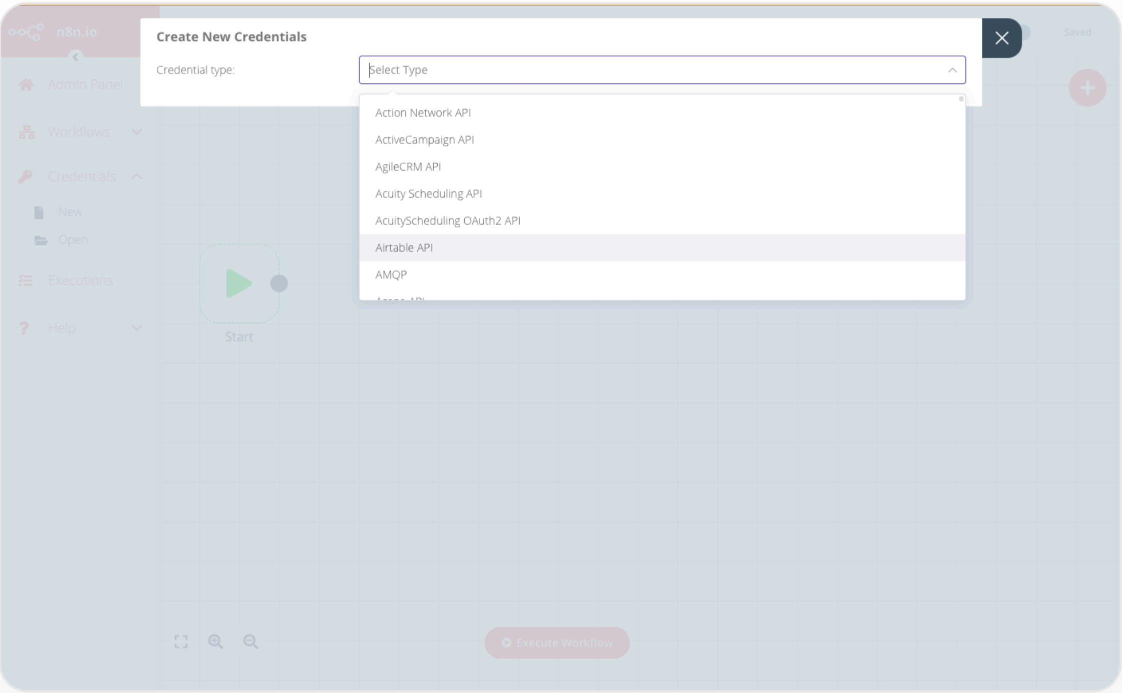
Reference: `https://docs.n8n.io/credentials/add-edit-credentials/`

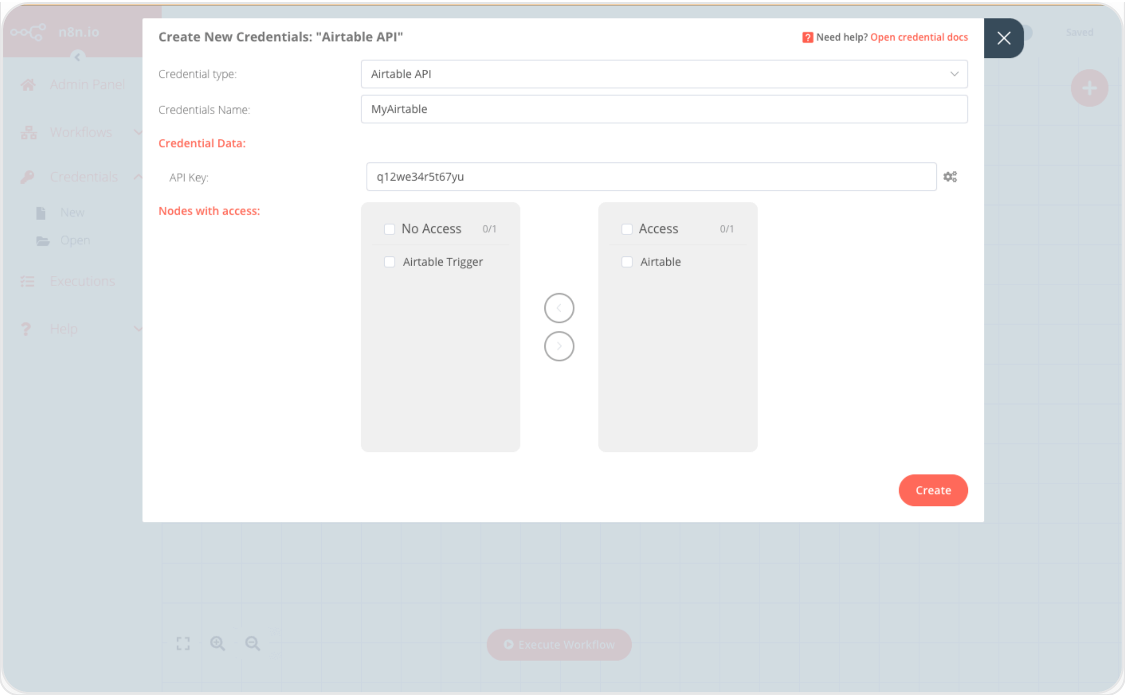
How credentials work

Credentials store authentication info (API keys, OAuth tokens, etc.) so you don't paste secrets into nodes repeatedly.

Good credential hygiene

- **Least privilege**: only grant scopes/permissions you need.
- **Naming convention** (example): `slack_prod_marketing_notifications`
  Include: service + env + purpose.
- **Rotate keys** periodically (especially if shared among team members).





Create New Credentials: "Airtable API"

Need help? Open credential docs

Credential type: Airtable API

Credentials Name: MyAirtable

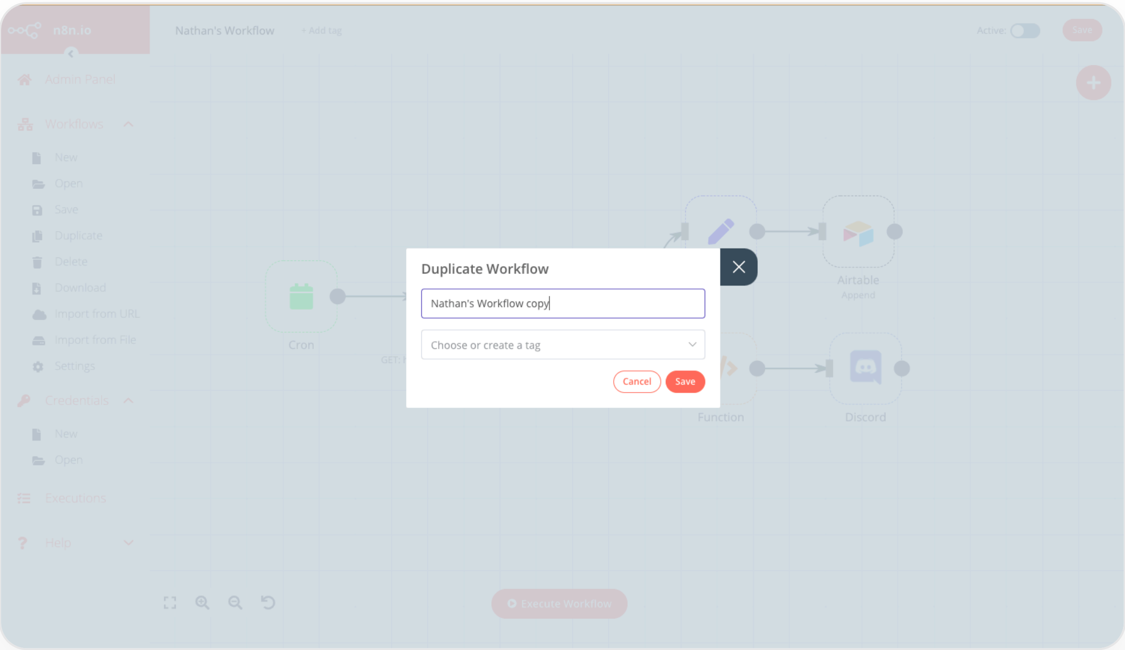
Credential Data:

API Key: q12we34r5t67yu

Nodes with access:

No Access 0/1	Access 0/1
□ Airtable Trigger	□ Airtable

Create



Duplicate Workflow

Nathan's Workflow copy

Choose or create a tag

Cancel Save

Screenshots from n8n Docs (Embed docs): https://docs.n8n.io/embed/managing-workflows/

Sharing workflows safely

n8n exports workflows as JSON. Those exports can include:

- credential **names** and IDs
- in some cases, imported cURL may contain auth headers

Reference: export/import cautions `https://docs.n8n.io/workflows/export-import`

Checklist before sharing a workflow JSON:

- Remove tokens, API keys, headers, and any personal data
- Replace with placeholders like `REDACTED`

- Verify nothing sensitive is stored in node parameters

Lesson 6 — Webhooks + HTTP Request (the universal connector)

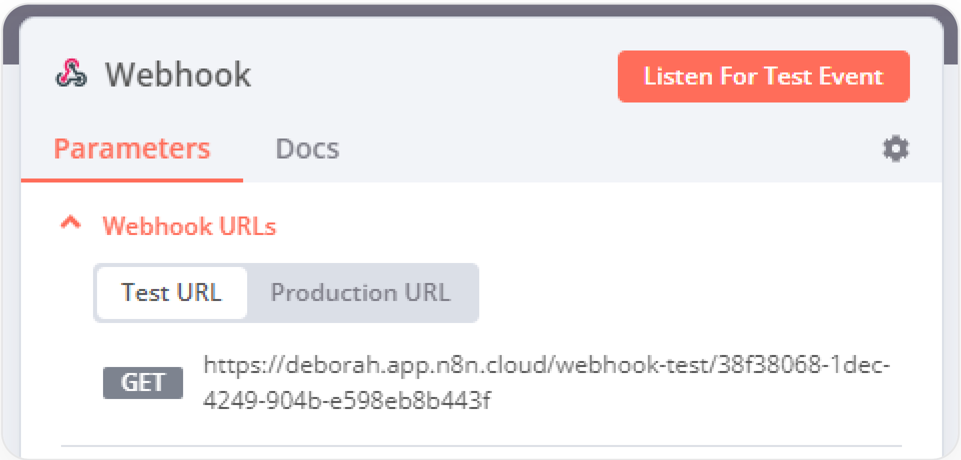
Goal: Understand the two nodes that let n8n connect to almost anything.
Outcome: You know when to use webhooks, when to use HTTP Request, and how to troubleshoot.

Webhooks (in plain English)

A webhook is a URL you give to another system so it can call your workflow when something happens.

Use webhooks when:

- you need real-time triggers (form submit, payment, event)
- the tool you're using can "send webhooks"



GET https://deborah.app.n8n.cloud/webhook-test/38f38068-1dec-4249-904b-e598eb8b443f

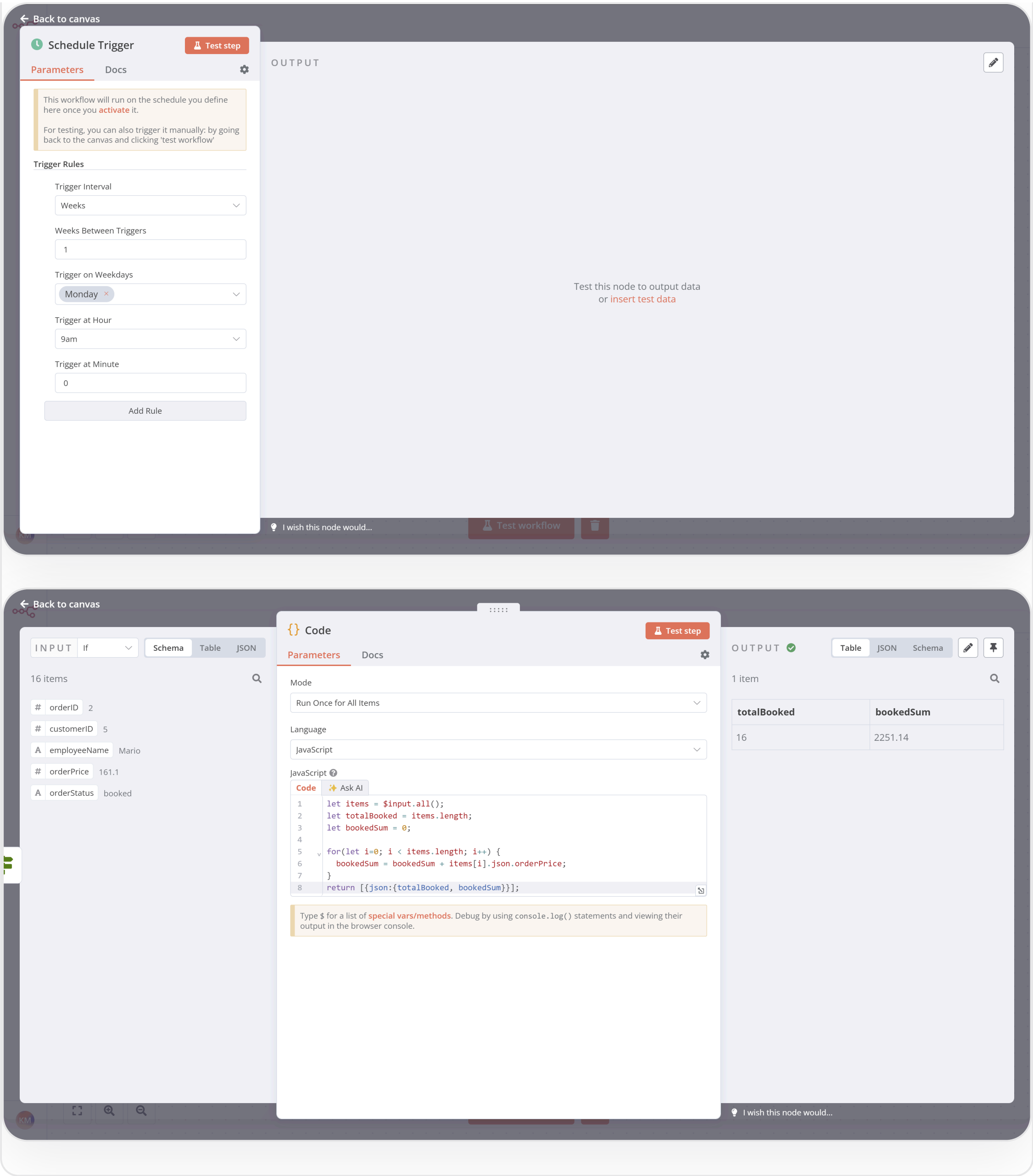
Screenshot from n8n Docs: https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.webhook/

HTTP Request node (in plain English)

The HTTP Request node lets n8n call any API endpoint (REST/JSON) — even if there isn't a dedicated integration.

Use HTTP Request when:

- an app has an API, but no built-in node exists
- you need an endpoint the built-in node doesn't expose
- you want total control (headers, pagination, retries, etc.)



Schedule Trigger

This workflow will run on the schedule you define here once you activate it.

For testing, you can also trigger it manually: by going back to the canvas and clicking 'test workflow'

Trigger Rules

Trigger Interval: Weeks

Weeks Between Triggers: 1

Trigger on Weekdays: Monday ×

Trigger at Hour: 9am

Trigger at Minute: 0

Add Rule

Test this node to output data or insert test data

16 items

# orderID	2
# customerID	5
A employeeName	Mario
# orderPrice	161.1
A orderStatus	booked

Code

Mode: Run Once for All Items

Language: JavaScript

JavaScript

Code | ✨ Ask AI

```javascript
let items = $input.all();
let totalBooked = items.length;
let bookedSum = 0;

for(let i=0; i < items.length; i++) {
  bookedSum = bookedSum + items[i].json.orderPrice;
}
return [{json:{totalBooked, bookedSum}}];
```

Type $ for a list of special vars/methods. Debug by using `console.log()` statements and viewing their output in the browser console.

1 item

totalBooked	bookedSum
16	2251.14

Screenshots from n8n Docs: https://docs.n8n.io/courses/level-one/chapter-5/chapter-5.7/ and https://docs.n8n.io/courses/level-one/chapter-5/chapter-5.5

A beginner-friendly HTTP checklist

- **Method**: GET (read), POST (create), PUT/PATCH (update), DELETE (remove)
- **Auth**: use credentials where possible (avoid pasting tokens)
- **Payload**: JSON body for POST/PATCH
- **Response**: confirm the shape and the number of returned items

> **Pro tip**: Build an API call *outside* n8n first (curl/Postman), then replicate it in HTTP Request. It's faster than guessing.

Troubleshooting: the 3 fastest checks

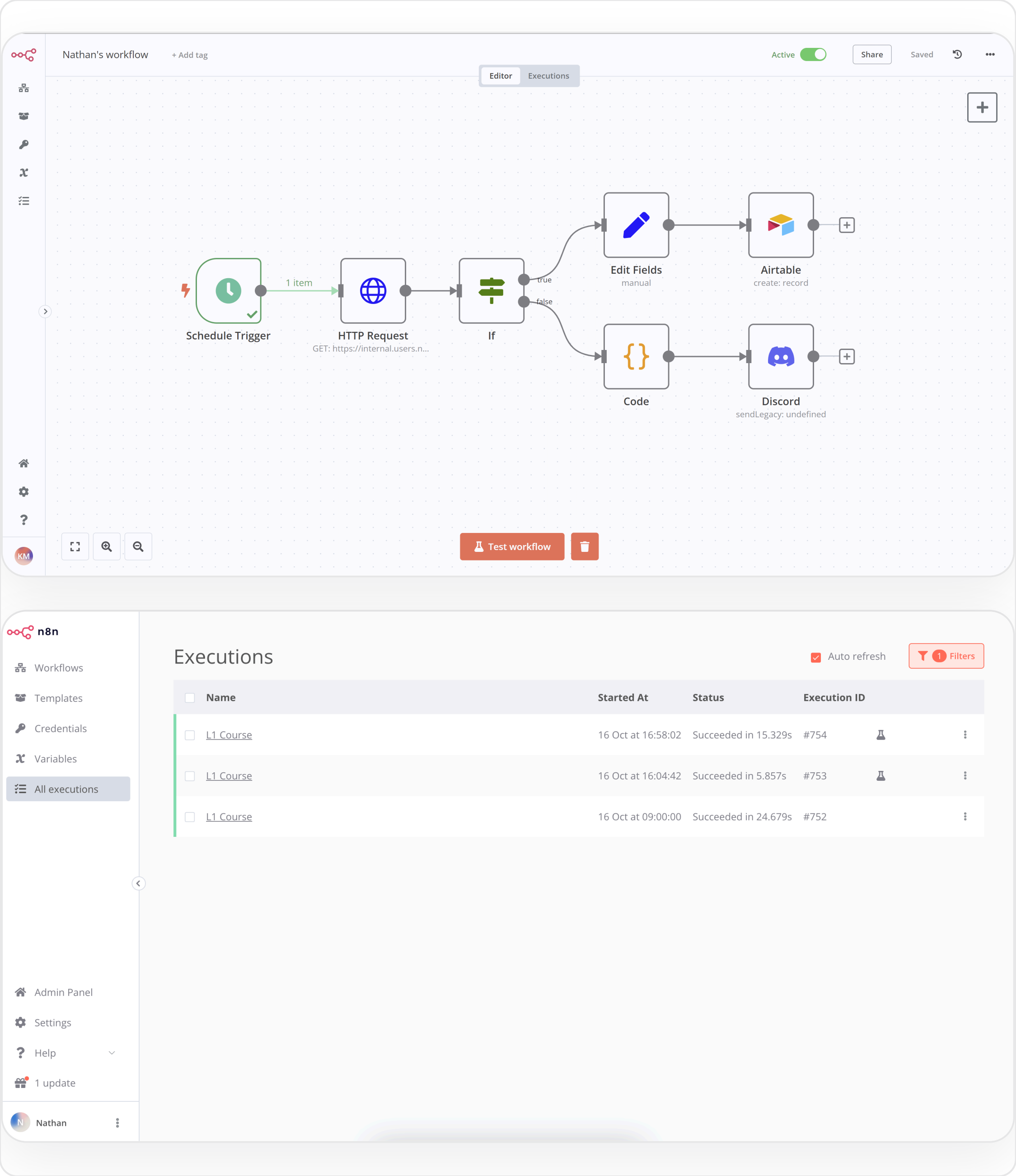
- **Status code**: 401/403 = auth, 404 = wrong URL, 429 = rate limit, 5xx = server issue

- **Request details**: headers + body match what the API expects
- **Items**: confirm you're calling the API once per item *only when you intend to*

Lesson 7 — Executions, testing, and debugging like a pro

Goal: Learn how n8n runs workflows and how to debug failures efficiently.
Outcome: You can use executions, data pinning, and "debug in editor" to fix issues quickly.

Reference: executions overview `https://docs.n8n.io/workflows/executions/`

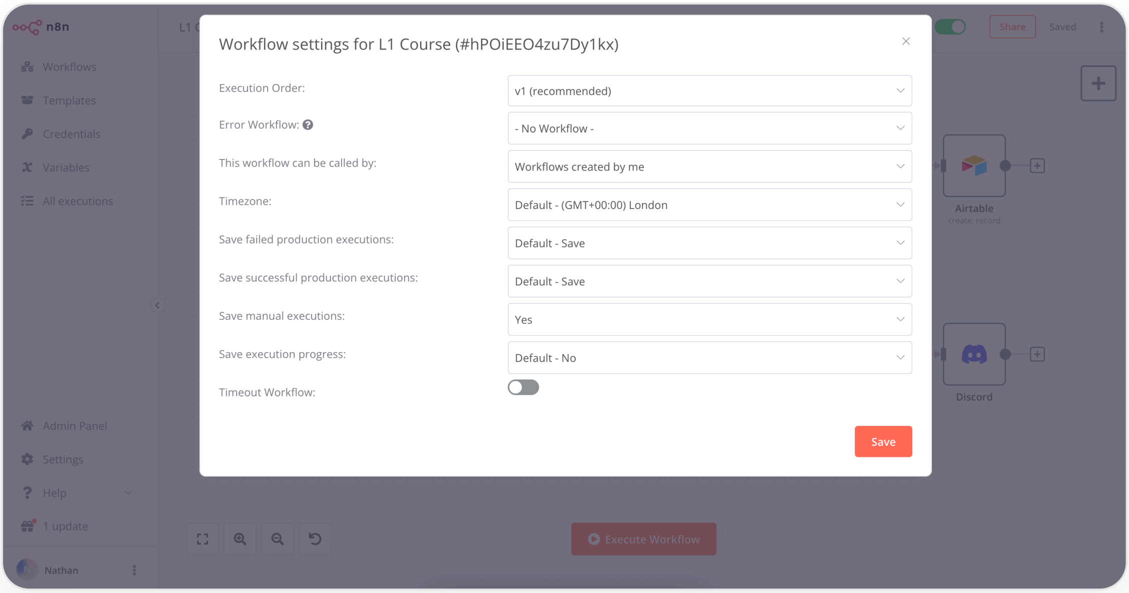


Executions

☑ Auto refresh

1 Filters

	Name	Started At	Status	Execution ID
□	L1 Course	16 Oct at 16:58:02	Succeeded in 15.329s	#754
□	L1 Course	16 Oct at 16:04:42	Succeeded in 5.857s	#753
□	L1 Course	16 Oct at 09:00:00	Succeeded in 24.679s	#752



Workflow settings for L1 Course (#hPOiEEO4zu7Dy1kx)

Execution Order: v1 (recommended)

Error Workflow: - No Workflow -

This workflow can be called by: Workflows created by me

Timezone: Default - (GMT+00:00) London

Save failed production executions: Default - Save

Save successful production executions: Default - Save

Save manual executions: Yes

Save execution progress: Default - No

Timeout Workflow:

Save

Screenshots from n8n Docs: https://docs.n8n.io/courses/level-one/chapter-5/chapter-5.8/

Manual vs production executions

- **Manual**: you click "Execute Workflow" while building/testing.
- **Production**: workflow runs automatically when **Active**.

> **Pro tip**: Keep workflows **Inactive** while building. Activate only when you're confident and have basic error handling.

Data pinning (your testing superpower)

Pinned data lets you freeze a node's output so you can iterate downstream without re-triggering upstream steps.

Useful when:

- the trigger is slow (e.g., external webhook)
- the source data changes frequently
- the upstream service has rate limits

Screenshot from n8n Docs: https://docs.n8n.io/data/data-pinning/

Debug past executions (especially production failures)

Reference: `https://docs.n8n.io/workflows/executions/debug/` When a run fails, you can load past execution data back into the editor and re-run with fixes.

Workflow to fix a failure:

1. Open **Executions**
2. Choose a failed execution
3. Select **Debug in editor**
4. Make a small change

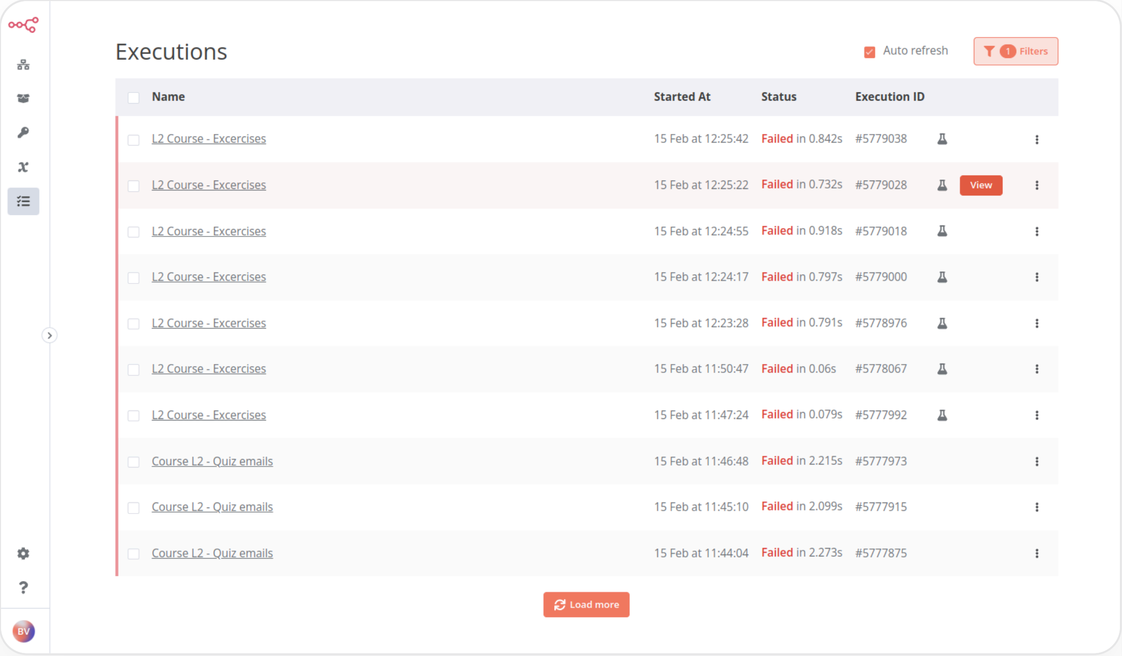
5. Re-run and confirm the fix

> **Common pitfall**: “Fixing” a workflow without reproducing the failing data. Always debug with the same inputs.

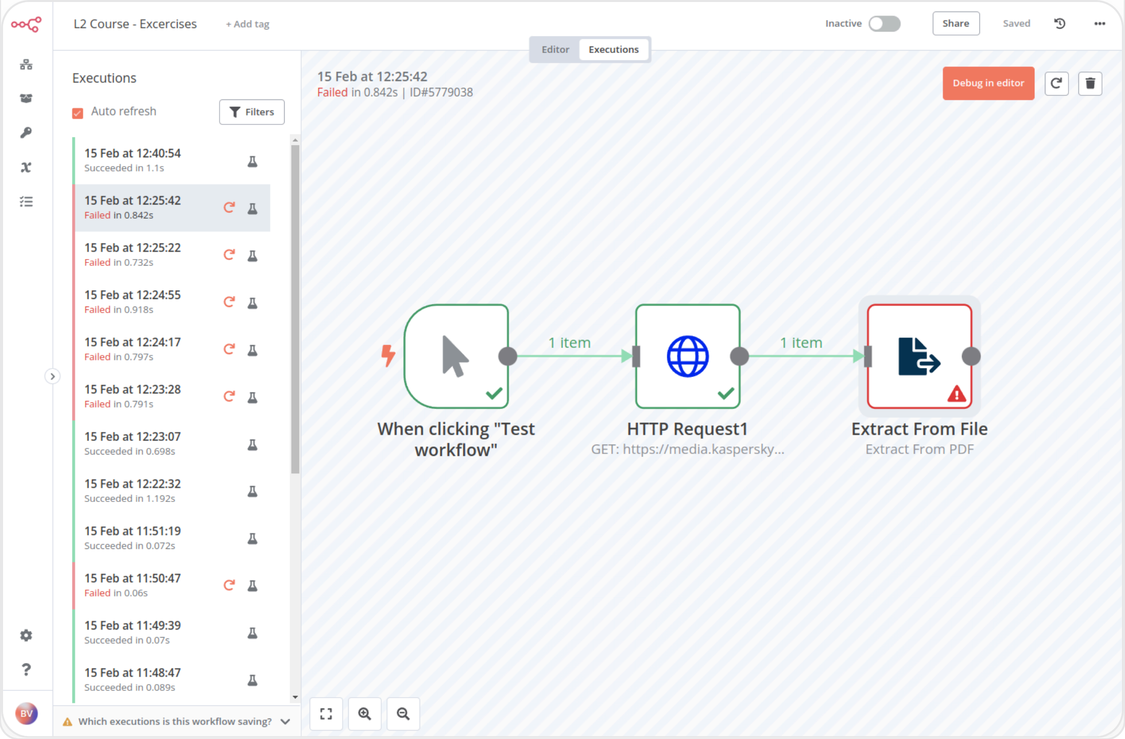
Lesson 8 — Error handling that keeps workflows reliable

Goal: Build workflows that fail loudly (and usefully) instead of silently breaking.
Outcome: You can create an error workflow and get notified when things go wrong.

Reference: `https://docs.n8n.io/flow-logic/error-handling`



□	Name	Started At	Status	Execution ID		
□	L2 Course - Excercises	15 Feb at 12:25:42	Failed in 0.842s	#5779038		⋮
□	L2 Course - Excercises	15 Feb at 12:25:22	Failed in 0.732s	#5779028	View	⋮
□	L2 Course - Excercises	15 Feb at 12:24:55	Failed in 0.918s	#5779018		⋮
□	L2 Course - Excercises	15 Feb at 12:24:17	Failed in 0.797s	#5779000		⋮
□	L2 Course - Excercises	15 Feb at 12:23:28	Failed in 0.791s	#5778976		⋮
□	L2 Course - Excercises	15 Feb at 11:50:47	Failed in 0.06s	#5778067		⋮
□	L2 Course - Excercises	15 Feb at 11:47:24	Failed in 0.079s	#5777992		⋮
□	Course L2 - Quiz emails	15 Feb at 11:46:48	Failed in 2.215s	#5777973		⋮
□	Course L2 - Quiz emails	15 Feb at 11:45:10	Failed in 2.099s	#5777915		⋮
□	Course L2 - Quiz emails	15 Feb at 11:44:04	Failed in 2.273s	#5777875		⋮



Executions

☑ Auto refresh

Filters

15 Feb at 12:40:54
Succeeded in 1.1s

15 Feb at 12:25:42
Failed in 0.842s

15 Feb at 12:25:22
Failed in 0.732s

15 Feb at 12:24:55
Failed in 0.918s

15 Feb at 12:24:17
Failed in 0.797s

15 Feb at 12:23:28
Failed in 0.791s

15 Feb at 12:23:07
Succeeded in 0.698s

15 Feb at 12:22:32
Succeeded in 1.192s

15 Feb at 11:51:19
Succeeded in 0.072s

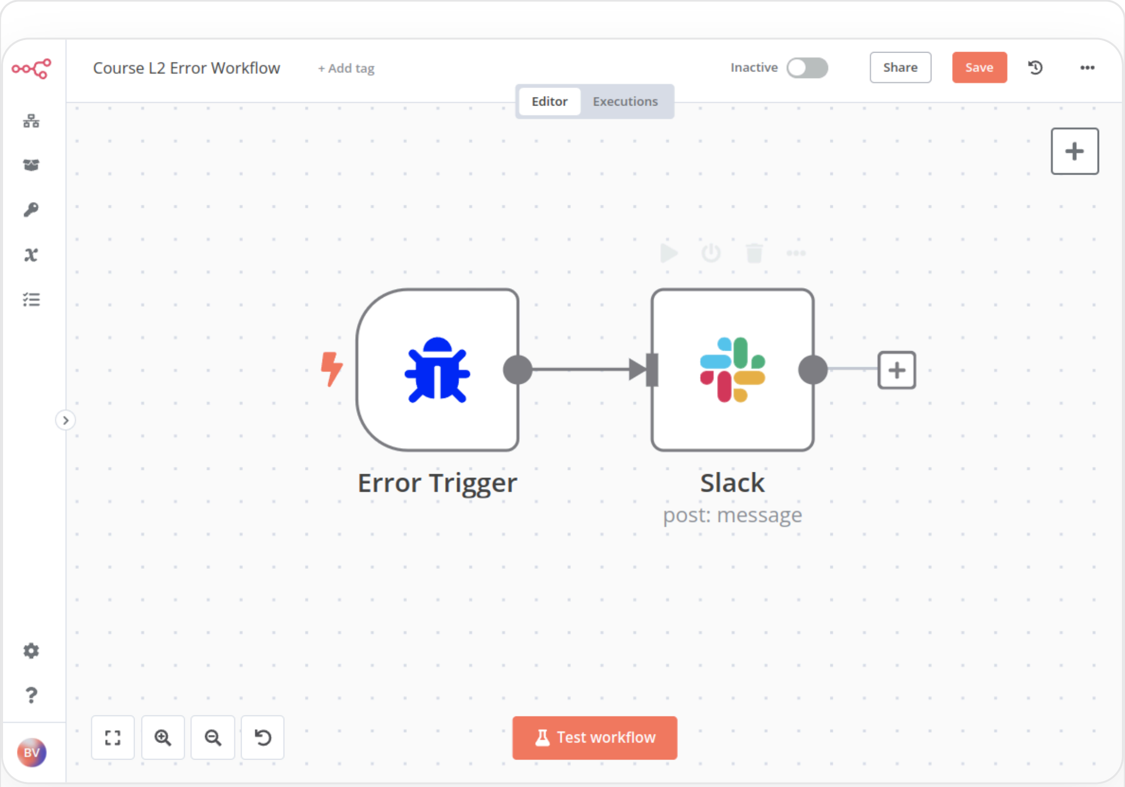
15 Feb at 11:50:47
Failed in 0.06s

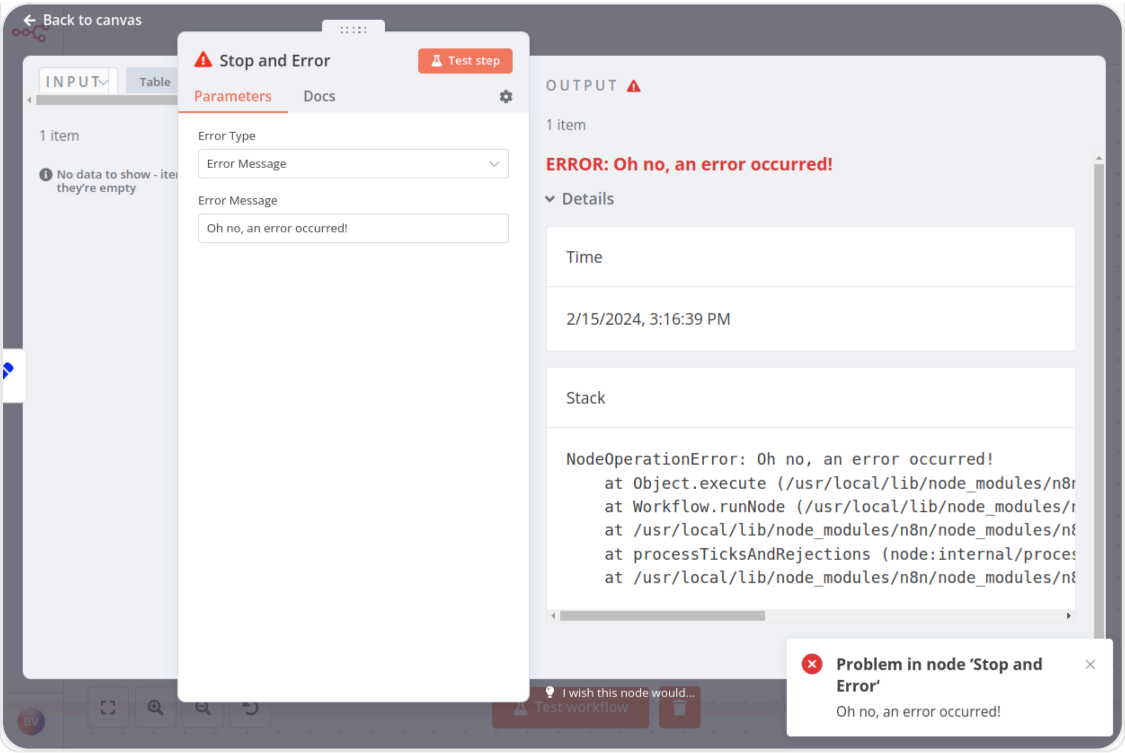
15 Feb at 11:49:39
Succeeded in 0.07s

15 Feb at 11:48:47
Succeeded in 0.089s

Which executions is this workflow saving?

15 Feb at 12:25:42
Failed in 0.842s | ID#5779038





Error Type: Error Message

Error Message: Oh no, an error occurred!

ERROR: Oh no, an error occurred!

Details

Time

2/15/2024, 3:16:39 PM

Stack

```
NodeOperationError: Oh no, an error occurred!
    at Object.execute (/usr/local/lib/node_modules/n8r
    at Workflow.runNode (/usr/local/lib/node_modules/r
    at /usr/local/lib/node_modules/n8n/node_modules/n8
    at processTicksAndRejections (node:internal/proces
    at /usr/local/lib/node_modules/n8n/node_modules/n8
```

Problem in node 'Stop and Error'

Oh no, an error occurred!

Screenshots from n8n Docs: https://docs.n8n.io/courses/level-two/chapter-4/

The error workflow pattern (recommended)

Create a separate workflow that starts with **Error Trigger** and then sends alerts (Slack/email) with execution details.

Why this matters

- You learn about failures quickly
- You can log failures centrally
- You can route issues to the right person/team

Using "Stop And Error" intentionally

Sometimes you want to fail on purpose:

- required field is missing
- validation fails
- upstream API returned unexpected data

Add **Stop And Error** to force a controlled failure, which then triggers your error workflow.

> **Pro tip**: "Fail fast" with a clear message. It turns mystery failures into obvious fixes.

Lesson 9 — Shipping, maintaining, and sharing workflows

Goal: Get from “it works on my laptop” to “it runs for weeks without drama.”
Outcome: You can activate, monitor, and safely share workflows.

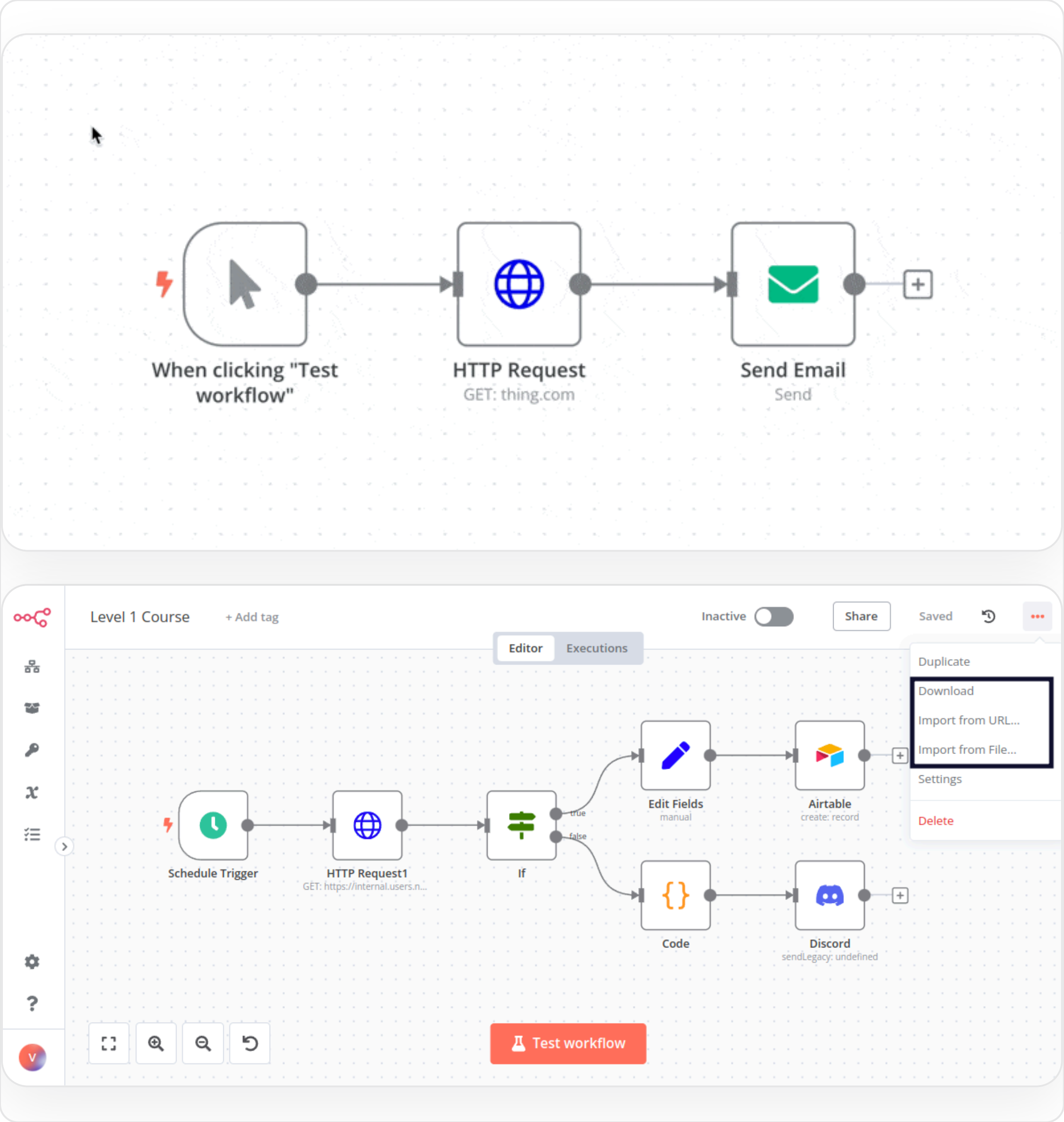
Before you activate (quick checklist)

- Inputs are validated (required fields checked)
- You’ve tested with real-ish data (not only perfect samples)
- You know how many items you expect at each step
- You’ve added basic error handling / notifications
- You’ve documented the workflow purpose + owner

Export and import workflows (for backups and reuse)

Reference: `https://docs.n8n.io/workflows/export-import`

- Export is JSON
- Great for templates and collaboration
- Review exports for sensitive info before sharing



Screenshots from n8n Docs: https://docs.n8n.io/workflows/export-import/

Workflow history and rollback (practical habit)

Even when you're solo:

- Save often
- Add notes about *why* a change was made
- Keep a "known good" version you can revert to quickly

Cheat sheets & quick references

"Before you build" workflow design canvas

Write these down before dragging nodes:

- **Trigger**: what starts it?
- **Inputs**: what fields do I receive?
- **Output**: what should happen (end state)?
- **Edge cases**: what if fields are missing? duplicates? rate limits?
- **Notifications**: how do I learn about failures?

Common node settings (what they usually mean)

(These labels vary by node, but the intent is consistent.)

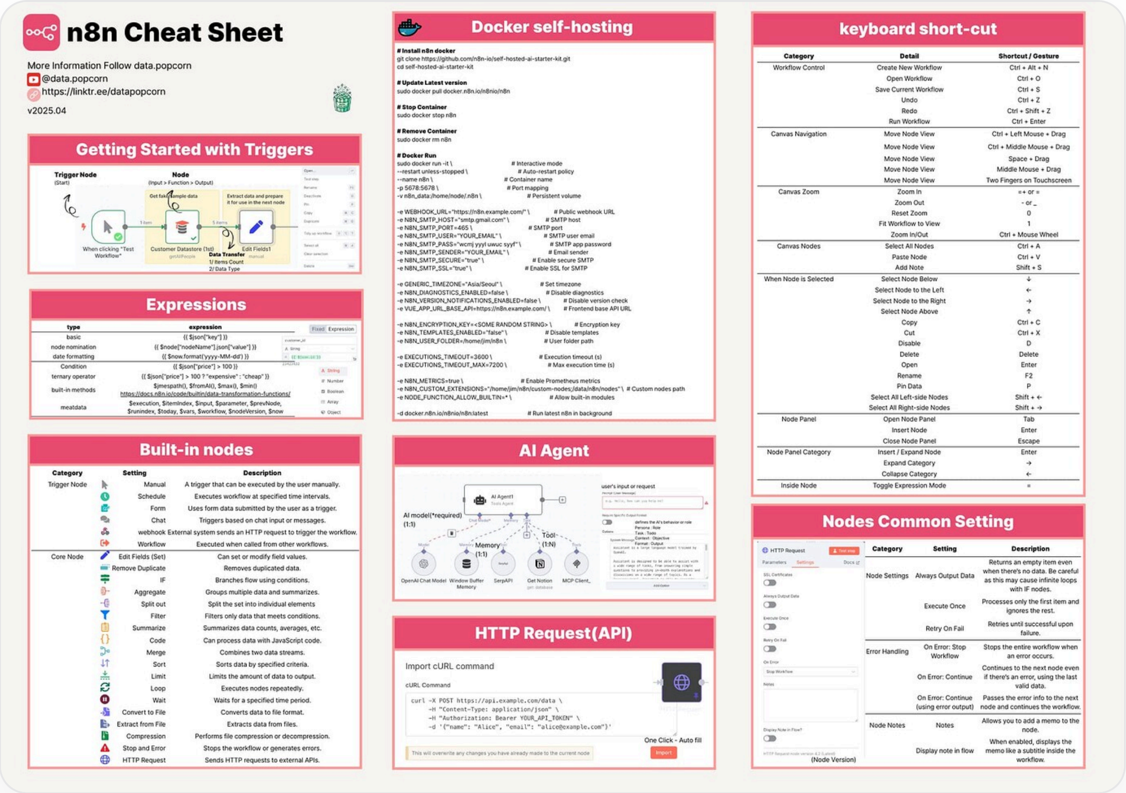
- **Continue On Fail**: keeps the workflow running even if this node errors (use carefully)
- **Retry On Fail**: retries transient errors (network, rate limits)
- **Execute Once**: reduce repeated calls when input has multiple items (use when appropriate)

Keyboard shortcuts (bookmark)

Reference: `https://docs.n8n.io/keyboard-shortcuts/`

- Ctrl/Cmd + S: save
- Ctrl + Enter: execute workflow
- Ctrl/Cmd + K: command bar
- Tab: open node panel
- P: pin node data
- Shift + S: sticky note

n8n cheat sheet (visual)



n8n Cheat Sheet

More Information Follow data.popcorn
@data.popcorn
https://linktr.ee/datapopcorn
v2025.04

Getting Started with Triggers

Trigger Node
(Start)
Node
(Input > Function > Output)
Get fake sample data
Extract data and prepare it for use in the next node
1 item
5 items
When clicking "Test Workflow"
Customer Datastore (1st)
getAllPeople
Edit Fields1
manual
Data Transfer
1/ Items Count
2/ Data Type

Expressions

type	expression
basic	{{ $json["key"] }}
node nomination	{{ $node["nodeName"].json["value"] }}
date formatting	{{ $now.format('yyyy-MM-dd') }}
Condition	{{ $json["price"] > 100 }}
ternary operator	{{ $json["price"] > 100 ? "expensive" : "cheap" }}
built-in methods	$jmespath(), $fromAI(), $max(), $min() https://docs.n8n.io/code/builtin/data-transformation-functions/
meatdata	$execution, $itemIndex, $input, $parameter, $prevNode, $runIndex, $today, $vars, $workflow, $nodeVersion, $now

Built-in nodes

Category	Setting	Description
Trigger Node	Manual	A trigger that can be executed by the user manually.
	Schedule	Executes workflow at specified time intervals.
	Form	Uses form data submitted by the user as a trigger.
	Chat	Triggers based on chat input or messages.
	webhook	External system sends an HTTP request to trigger the workflow.
	Workflow	Executed when called from other workflows.
Core Node	Edit Fields (Set)	Can set or modify field values.
	Remove Duplicate	Removes duplicated data.
	IF	Branches flow using conditions.
	Aggregate	Groups multiple data and summarizes.
	Split out	Split the set into individual elements
	Filter	Filters only data that meets conditions.
	Summarize	Summarizes data counts, averages, etc.
	Code	Can process data with JavaScript code.
	Merge	Combines two data streams.
	Sort	Sorts data by specified criteria.
	Limit	Limits the amount of data to output.
	Loop	Executes nodes repeatedly.
	Wait	Waits for a specified time period.
	Convert to File	Converts data to file format.
	Extract from File	Extracts data from files.
	Compression	Performs file compression or decompression.
	Stop and Error	Stops the workflow or generates errors.
	HTTP Request	Sends HTTP requests to external APIs.

Docker self-hosting

```
# Install n8n docker
git clone https://github.com/n8n-io/self-hosted-ai-starter-kit.git
cd self-hosted-ai-starter-kit

# Update Latest version
sudo docker pull docker.n8n.io/n8nio/n8n

# Stop Container
sudo docker stop n8n

# Remove Container
sudo docker rm n8n

# Docker Run
sudo docker run -it \                               # Interactive mode
--restart unless-stopped \                          # Auto-restart policy
--name n8n \                                        # Container name
-p 5678:5678 \                                      # Port mapping
-v n8n_data:/home/node/.n8n \                       # Persistent volume

-e WEBHOOK_URL="https://n8n.example.com/" \         # Public webhook URL
-e N8N_SMTP_HOST="smtp.gmail.com" \                 # SMTP host
-e N8N_SMTP_PORT=465 \                              # SMTP port
-e N8N_SMTP_USER="YOUR_EMAIL" \                     # SMTP user email
-e N8N_SMTP_PASS="wcmj yyyl uwuc syyf" \            # SMTP app password
-e N8N_SMTP_SENDER="YOUR_EMAIL" \                   # Email sender
-e N8N_SMTP_SECURE="true" \                         # Enable secure SMTP
-e N8N_SMTP_SSL="true" \                            # Enable SSL for SMTP

-e GENERIC_TIMEZONE="Asia/Seoul" \                  # Set timezone
-e N8N_DIAGNOSTICS_ENABLED=false \                  # Disable diagnostics
-e N8N_VERSION_NOTIFICATIONS_ENABLED=false \        # Disable version check
-e VUE_APP_URL_BASE_API=https://n8n.example.com/ \  # Frontend base API URL

-e N8N_ENCRYPTION_KEY=<SOME RANDOM STRING> \        # Encryption key
-e N8N_TEMPLATES_ENABLED="false" \                  # Disable templates
-e N8N_USER_FOLDER=/home/jim/n8n \                  # User folder path

-e EXECUTIONS_TIMEOUT=3600 \                        # Execution timeout (s)
-e EXECUTIONS_TIMEOUT_MAX=7200 \                    # Max execution time (s)

-e N8N_METRICS=true \                               # Enable Prometheus metrics
-e N8N_CUSTOM_EXTENSIONS="/home/jim/n8n/custom-nodes;/data/n8n/nodes" \ # Custom nodes path
-e NODE_FUNCTION_ALLOW_BUILTIN=* \                  # Allow built-in modules

-d docker.n8n.io/n8nio/n8n:latest                   # Run latest n8n in background
```

AI Agent

defines the AI's behavior or role
Persona : Role
Task : Todo
Context : Objective
Format : Output

HTTP Request(API)

Import cURL command

cURL Command

```
curl -X POST https://api.example.com/data \
     -H "Content-Type: application/json" \
     -H "Authorization: Bearer YOUR_API_TOKEN" \
     -d '{"name": "Alice", "email": "alice@example.com"}'
```

One Click - Auto fill

This will overwrite any changes you have already made to the current node
Import

keyboard short-cut

Category	Detail	Shortcut / Gesture
Workflow Control	Create New Workflow	Ctrl + Alt + N
	Open Workflow	Ctrl + O
	Save Current Workflow	Ctrl + S
	Undo	Ctrl + Z
	Redo	Ctrl + Shift + Z
	Run Workflow	Ctrl + Enter
Canvas Navigation	Move Node View	Ctrl + Left Mouse + Drag
	Move Node View	Ctrl + Middle Mouse + Drag
	Move Node View	Space + Drag
	Move Node View	Middle Mouse + Drag
	Move Node View	Two Fingers on Touchscreen
Canvas Zoom	Zoom In	=+ or =
	Zoom Out	- or _
	Reset Zoom	0
	Fit Workflow to View	1
	Zoom In/Out	Ctrl + Mouse Wheel
Canvas Nodes	Select All Nodes	Ctrl + A
	Paste Node	Ctrl + V
	Add Note	Shift + S
When Node is Selected	Select Node Below	↓
	Select Node to the Left	←
	Select Node to the Right	→
	Select Node Above	↑
	Copy	Ctrl + C
	Cut	Ctrl + X
	Disable	D
	Delete	Delete
	Open	Enter
	Rename	F2
	Pin Data	P
	Select All Left-side Nodes	Shift + ←
	Select All Right-side Nodes	Shift + →
Node Panel	Open Node Panel	Tab
	Insert Node	Enter
	Close Node Panel	Escape
Node Panel Category	Insert / Expand Node	Enter
	Expand Category	→
	Collapse Category	←
Inside Node	Toggle Expression Mode	=

Nodes Common Setting

(Node Version)

Category	Setting	Description
Node Settings	Always Output Data	Returns an empty item even when there's no data. Be careful as this may cause infinite loops with IF nodes.
	Execute Once	Processes only the first item and ignores the rest.
	Retry On Fail	Retries until successful upon failure.
Error Handling	On Error: Stop Workflow	Stops the entire workflow when an error occurs.
	On Error: Continue	Continues to the next node even if there's an error, using the last valid data.
	On Error: Continue (using error output)	Passes the error info to the next node and continues the workflow.
Node Notes	Notes	Allows you to add a memo to the node.
	Display note in flow	When enabled, displays the memo like a subtitle inside the workflow.

Common mistakes (and the fix)

"It ran 50 times"

- **Cause**: you had 50 input items
- **Fix**: add a Limit/Aggregate/merge-to-single-item step before the action

"Expressions aren't working"

- **Cause**: wrong data path, or you're referencing a field that doesn't exist
- **Fix**: inspect the incoming item JSON and copy the path from the variable selector

"Webhook works in test but not in production"

- **Cause**: different URLs or environments; or workflow inactive
- **Fix**: confirm active workflow, correct webhook URL, and any auth headers expected by the sender

"Credentials keep failing"

- **Cause**: wrong scopes/permissions or expired token
- **Fix**: re-authorize with least-privilege scopes and test connection on save

Next steps (where to go from here)

- Learn from n8n's tutorials and courses:

- Try it out: `https://docs.n8n.io/try-it-out/`
- Level one (UI + fundamentals): `https://docs.n8n.io/courses/level-one/chapter-1/`

- Pick a real workflow to automate:
  - Lead capture → CRM/Sheet → Slack alert
  - Daily report → Slack/email digest
  - Customer support → classify + route + notify
- Start a template library:
  - “Webhook to Slack”
  - “HTTP Request + pagination”
  - “Error handler workflow”

Sources & further reading

Automate Digital

- `https://automatedigital.ai/n8n`

n8n Docs

- Editor UI: `https://docs.n8n.io/courses/level-one/chapter-1/`
- Expressions: `https://docs.n8n.io/code/expressions/`
- Data structure: `https://docs.n8n.io/key-concepts/`
- Credentials: `https://docs.n8n.io/credentials/add-edit-credentials/`
- Executions: `https://docs.n8n.io/workflows/executions/`
- Debug past executions: `https://docs.n8n.io/workflows/executions/debug/`
- Error handling: `https://docs.n8n.io/flow-logic/error-handling`
- Export/import: `https://docs.n8n.io/workflows/export-import`
- Keyboard shortcuts: `https://docs.n8n.io/keyboard-shortcuts/`

Want help building workflows that actually save you time?

Once your n8n instance is running, Automate Digital can help you:

- design high-ROI workflows (based on your real bottlenecks)
- build reliable production automations with monitoring and error handling
- integrate your tools, data, and AI capabilities safely

Free Automation Opportunity Audit (live demo + roadmap):

Book Free Automation Opportunity Audit